THE MAKING OF COMPUTER SCIENTISTS:

RENDERING TECHNICAL KNOWLEDGE, GENDER, AND ENTREPRENEURIALISM IN SINGAPORE

by

Samantha Breslin

A dissertation submitted to the

School of Graduate Studies

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Department of Anthropology

Memorial University of Newfoundland

August 2018

St. John’s, Newfoundland and Labrador

Abstract

This dissertation explores the making of computer scientists in Singapore. I explore how transnational computer science epistemologies and Singaporean state policies work to render the world into technical problems that computer scientists can manipulate and solve. Computer science knowledge and practice is thereby presented as mobile, while masking the colonization of places like Singapore by specifically rendered and gendered American computer science. I also map out the diffractive effects of these transnationally mobile renderings. This research is based on participant observation and interviews centring on an undergraduate computer science program in Singapore. Singaporean and technology media, Singaporean government policies, and university and computer science curricula are also analyzed.

I first show how students learn to model and render "reality" into technical frames, creating naturalized computing "worlds," but ones wherein magic is real and computer scientists are the magicians. Heteronormative binary renderings of gender are (re)produced within these worlds through narratives about algorithms and computing "problems" that constitute a transnational, but US-centric, tradition and that govern the possible ways for students and professors to think about and do computer science. I also show how students themselves are "rendered technical" and their lives and identities "torqued" as they are summoned to inhabit gender norms and hegemonic values of neoliberal competition, passion, and entrepreneurialism. In particular, the performance of passion by certain students works to create a gendered benchmark against which all students come to measure themselves, but which often turns to promoting over-work and exploitation in the name of career development and innovation.

Moreover, while some students perform situationally dependent and fluid gender identities, I argue that the predominance of research reducing gender to the question of "women in" computing limits the possibilities both for research on and enactments of gender in computer science and works both to mask and reproduce gender inequalities. Yet, I also show how – in the space produced through conflicting intra-actions of different norms and values – students' performances of self complicate binary renderings of gender and disrupt the hegemonic status of neoliberal passion and entrepreneurialism, suggesting new possibilities for becoming/being a "good" computer scientist.

Acknowledgements

Thank you to all those in Singapore who contributed to this research. Thanks especially to the students who shared their stories about their lives and their experiences learning computer science, and to the professors who welcomed me into their classrooms. I also thank the many people, including administrators, professors, and professionals, who helped me during my time in Singapore, made the time to speak with me, and shared their insights about computing, education, and work in Singapore. Thank you also to Bimlesh Wadhwa for enthusiastically engaging and working with me on questions of gender and computing.

My colleagues and professors have provided much intellectual and personal guidance. To my supervisor, Robin Whitaker, who has seen me through two degrees and has always had insightful suggestions, helpful advice, and given such dedicated and unwavering support, thank you so much for being a great supervisor. Thank you to my committee, Reade Davis, Wayne Fife, and Lucy Suchman, for your thoughtful suggestions throughout the course of this project, and to my examiners, Flis Henwood, Nicole Power, and Mark Tate for your helpful insights and comments on this dissertation. Thank you also to all members of the Department of Anthropology for all that I've learned over these many years. My fellow graduate students and friends, particularly Dianne West, Dave Cooney, Consuelo Griggio, Josh Lalor, Karen Samuels, Tracy Winters, and Joonas Plaan have also provided much commiseration and encouragement. The Department of Sociology at Lancaster University provided an intellectually stimulating and inspiring visit over the three months I was there preparing for fieldwork.

The members of SafetyNet have also provided a productive and welcoming place to work, as well as always having tea and food.

I am also grateful for the financial support provided by the Social Sciences and Humanities Research Council in the form of a Joseph-Armand Bombardier Canada Graduate Scholarship and a Michael Smith Foreign Study Supplement; by Memorial University of Newfoundland, including a School of Graduate Studies Fellowship, the Scotia Bank Bursary for International Study, the Professor Peter Hart Memorial Scholarship, and the Scholarship in the Arts Doctoral Completion Award; and through research assistant work with Lachlan Barber and Barb Neis with the On the Move Partnership.

My friends and family have also been of great help. Special thanks to Susan Walling, Michelle Brophy, and Rob Brown for all the tunes and good times. Thank you to my parents who have always supported me in all my goals and endeavours and for always being there for me. My cat Amber, who passed shortly before the final submission of this dissertation, provided constant (if sometimes insistent) companionship. Finally, thank you to Ben Staple who has provided unending encouragement, support, and insight, along with reading and editing multiple drafts.

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

I am sitting in a large common area full of beige walls, typing on my laptop. The cool air around me belies the hot dampness outside, kept out only by two sets of glass doors. Out there it feels like being hit in the face with a soggy wool blanket, as a Mormon missionary that I met on the bus one evening put it. The room is mostly quiet, save for the hum of fans and aircon (the local term for air-conditioning), chairs shifting, and faint music coming from a monitor embedded in a wall nearby scrolling headlines such as "Google chops $100 off Nexus prices" and "High-tech views of massive Yosemite Rim fires" from international tech news, along with school events. Along one wall there are several framed posters entitled "FYP [Final Year Project] Innovation Award" and "Dean's List: Academic Year 2012/2013," each with multiple wallet sized photos of the proud recipients.

Throughout the room, small clusters of people sit around tables, near pillars that provide crucial plug-in points for laptops, tablets, and mobile phones. Most stare intently at the laptops opened in front of them, headphones in their ears to block out surrounding distractions, and fingers resting on keys, ready to start typing. They keep their mobile phones nearby, frequently checking them for new WhatsApp notifications or texts. Almost everyone, myself included, is dressed casually and for the hot weather outside: t-shirts plastered with large logos – "Palantir," "Adidas," "Microsoft" – shorts, and slippers (the local term for flip-flops or thongs). A woman looking to be in her fifties, dressed in a white shirt trimmed in blue and wearing a head-covering, walks past us pushing a cleaning cart towards the nearby toilets. From another direction an older man wearing dark coloured slacks, a polo shirt, and dress shoes, and carrying a book and stack of papers in one hand, walks through a large wooden door along a far wall marked "Seminar Room."

This area is part of the computing building at the university where I conducted my fieldwork, which I am referring to as Temasek University.[1] Temasek University is a large publicly funded university in the Republic of Singapore. The building dedicated to computer science, and the focal point of my research, was near the centre of a campus so large and sprawling that it had its own bus system, which enabled students to avoid walking or running in the heat and humidity with the added luxury of air-conditioning. Computing related activities, however, were also spread across the campus. The student hacker group, for example, offered workshops and regularly met elsewhere, student project groups often met in student residences, and individual and groups of students studied and worked in any of the university's numerous libraries and common areas. Through participant observation in these various contexts, as well as semi-structured interviews, and curriculum, policy, and media analysis, my research explores the "making of computer scientists" – how students are shaped as socio-technical persons through computer science education.

Computer science is the study of both the theory and practice of computing. This involves designing software, solving computing problems, and developing new ways of using computers. The definition and boundaries of what constitutes "computer science" are highly contested and I do not seek to resolve them here. Rather, this dissertation

[1] I discuss in Chapter 2 my decision to use a pseudonym for the university.

explores what it means to become a computer scientist in Singapore as manifest in the systematic and informal kinds of training entailed in undergraduate computer science education. That is, I consider the processes of subject-formation that are part of learning and doing computer science: What are students' experiences in learning computer science? How are gender, culture, and citizenship implicated in learning and doing computer science? How does computer science in Singapore relate to computing in other times and places?

Computer science as a discipline (however defined) is growing in both scope and influence. Code and programs are embedded in and control the functioning of a vast array of activities:

> Whilst we are dead to the world at night, networks of machines silently and repetitively… monitor, control and assess the world using electronic sensors, updating lists and databases, calculating and recalculating their models to produce reports, predictions and warnings…During our waking hours, a multitude of machines open and close gates and doors, move traffic-lights… and generally keep the world moving. To do this requires millions, if not, billions of lines of computer code, many thousands of man-hours [*sic*] of work, and constant maintenance and technical support to keep it all running (Berry 2011, 1).

There are also increasing calls within and beyond academia to see computer science and programming as a new and basic form of literacy (Guzdial 2016, 1–10); programming is foretold as "the language of the future." Alongside the spread of computing, the ubiquity of computer science and programming are thus also being promoted. It is becoming a moral imperative for everyone – including children as young as two years old, in Singapore and many other countries – to "learn to code" (Ames 2017; Devlin 2017; Missio 2015; Tan 2018; Tham 2017). This dissertation explores what it is like to learn to code, and particularly to learn to do computer science, for undergraduate students in

Singapore. At the same time, while this dissertation is a story about students' experiences at one particular university in Singapore, it is simultaneously a story about computer science as a transnational discipline and practice, particularly in relation to the discipline's history in the United States, as well as about the worlds that are learned and created through its study.

The main purpose of this chapter is to situate the analytical and theoretical approaches that guide this dissertation. First, I discuss my attention to practices of "rendering technical" that drive my analysis, and then discuss the role of and literatures relating to gender and technology, and the social study of education and subject-formation. Finally, I offer a brief roadmap to the rest of the dissertation.

1.1 Computer Science Renderings

Undergraduate computer science in Singapore in many ways bore a striking resemblance to my own undergraduate experiences studying computer science 15,000 km away at the University of Waterloo. In both places, first-term students are introduced to programming and to algebraic and discrete mathematics, alongside several elective courses. Yet, this initial breadth of choice quickly shrinks as students delve deeper into the key data structures, algorithms, technologies, and specialties that make up the discipline of computer science. Students become integrated into worlds of mathematics and computation where they learn how to represent, model, and manipulate information – and "reality" – through modelling languages, usage diagrams, abstract data structures, language objects, and algorithmic processes.

Throughout this thesis, I explore how the work of constructing and translating reality through code, programs, algorithms, and computing worlds operates as a process of "rendering technical." This is a term I borrow from Tania Murray Li (2007) in relation to her work on development in Indonesia, while also building on Natasha Myers's (2014; 2015) work on machine models and metaphors for protein molecules and Karen Barad's (1998; 2003; 2007) discussions of "intra-action" and posthumanist performativity. "Renderings… recursively transform the ways we see and intervene in the world" (Myers 2014, 156). In this regard, reality, along with code, languages, programs, models, and computer scientists themselves, are mutually (re)produced through their intra-actions. "Intra-action" denotes these processes of mutual production, where subjects and objects with stable boundaries do not exist, but rather intra-act to produce current historically-situated ways of being in the world (Barad 2007, 33, 139-140).[2]

Rendering technical, in particular, is a kind of biopolitics whereby particular practices and processes are delimited and made visible, explicit, and then subject to management, (re)producing technical governance as part of computer students' practices of self-making and as part of the construction of reality more broadly (Li 2007; Ferguson 1990; Foucault 1982,1991,2008; Rose 2004). Li adapted the concept from Nikolas Rose (2004) and from James Ferguson (1990). As Rose explains about practices of governance:

> This is a matter of defining boundaries, rendering that within them visible, assembling information about that which is included and devising techniques to mobilize the forces and entities thus revealed. For example, before one can seek to manage a domain such as an economy it is first necessary to conceptualize a set

[2] I discuss Barad's approach to posthumanist performativity further below.

> of bounded processes and relations as an economy which is amenable to management (Rose 2004, 33).

Ferguson additionally uses the concept of development discourse to explore how political and economic issues are continually reconstructed and presented as problems with technical solutions (Ferguson 1990, 27–29). Li draws on these insights to explore how development operates through a "will to improve," a form of Foucauldian governmentality that enacts calculated interventions through two practices: problematization and rendering technical (Li 2007, 6–7; Foucault 1991). Problematization, rendering technical, and development discourse are practices of translation whereby the complexities and politics of social relations, cultural lives, and economic inequalities are reinterpreted and captured in numerical quantifications, abstract symbols, data structures, and analytical problems such that computer science students – like development experts – have the power to solve them. In Chapter 3, I consider some of the ways this delimitation, specification, and problematization is accomplished in computer science, as students are presented with precise problems that they must solve through code, and taught various techniques for modelling and translating reality with a view to defining the problem and creating a programmatic solution.

This facet of rendering technical depends on the process of "inscription" discussed by Bruno Latour (1987) whereby aspects of the world or reality are made mobile, combinable, and stable, properties of what Latour terms "immutable mobiles" (Latour 1987, 223). They work to increase the mobility and combinability of encounters by creating "traces," by translating reality into different forms. Latour (1987) argues that scientific and mathematical equations are forms of translation, integral to the process of network building. Throughout this dissertation, I similarly explore the ways that code, programs, computing curricula, technologies, and persons are rendered as immutable mobiles that circulate through transnational disciplinary (and other) networks, constituting those networks in the process. Yet, while mobility, stability, and combinability can be gained through technical renderings or inscriptions, such translations are rarely accomplished without some accompanying loss and transformation.

Moreover, I consider how the mobility of these immutable mobiles entails directional frictions that often privilege values and practices from the US and other "Western" countries. Following Anna Tsing's idea of "friction," movements are made easier or harder based on particular symbolic and material encounters – previous paths that have been traced, shared symbolic or institutional connections – and sometimes produces surprising results: "friction inflects historical trajectories, enabling, excluding, and particularizing" (Tsing 2005, 6). I consider the particularities of the movement of code, algorithms, curricula, and expertise across trans-national spaces and boundaries, where trans "denotes both moving through space or across lines, as well as changing the nature of something" (Ong 1999, 4). I show how "flows" of persons, knowledges, technologies, and curricula are not unfettered and ungrounded, but entail the colonization of specifically rendered and gendered American computer science in Singapore, while often restricting the mobility of actors from Singapore to other places.

In this regard, David Harvey (2005) has shown how institutions, social relations, and ways of life have been transformed around the world through transnational projects of neoliberalization, where neoliberalism entails both a system of governmentality and a specific political economic system (Ortner 2016, 52). As Harvey explains,

> Neoliberalism is in the first instance a theory of political economic practices that proposes that human well-being can best be advanced by liberating individual entrepreneurial freedoms and skills within an institutional framework characterized by strong private property rights, free markets, and free trade. The role of the state is to create and preserve an institutional framework appropriate to such practices. (Harvey 2005, 2).

The Singaporean government has had an active role in fostering neoliberalization through openness to multinational corporations and inviting foreign finance, foreign companies, and foreign expertise to operate in and through Singapore (Gopinathan 2001, 23). Within that context, I explore in Chapters 5 to 8 the cultivation of individualism, entrepreneurialism, and competition among students at Temasek University, with one another and with (sometimes imagined) others around the world. At the same time, the transnational relationships produced and enacted through these processes of neoliberalization continually position Singapore – and most people and places outside of Silicon Valley – as deficient, reproducing qualitative labour shortages and the recurring need for new and new kinds of workers, and for new efforts towards technological and economic development.

Taken together, from an anthropological and feminist perspective, rendering technical is a socially and culturally constructed practice, related to what Donna Haraway has termed an "informatics of domination" (Haraway 1991a, 161–64).[3] More specifically, practices broadly relating to rendering technical have been explored and subjected to critique by feminist scholars and others for reproducing universalist world-views that contribute to forms of colonialism, imperialism, and domination in terms of gender, race, and other inequalities (Bowker 1993; Foucault 1997b; Haraway 1991a, 1997; Hayles 1999; Suchman 2007b). Related practices include the disembodiment of information (Hayles 1999); C^3I – "command-control-communication-intelligence, the military's symbol for its operations theory" (Haraway 1991a, 164, see also Edwards 1997); audit culture (Shore and Wright 2015a,b; Strathern 2000); statistical and other forms of modelling (Hacking 1990); high-modernist imperialism (Scott 1998), among others. I explore different ways that students learn to enact practices of rendering technical, enabling and producing for them, and for computer scientists more generally, the power and possibility to "change the world."

The relevance of theoretical insights from the anthropology of development to the study of computer science is revelatory here; in many ways international development and technological development have similar goals, and in some cases they are overlapping practices. The "will to improve" that Li identifies as intrinsic to development practices also pervades computer science education. During an interview with a second year student, for example, I asked him what he wanted to do after he finished his degree. His answer was simple. He wanted to change the world: "That's the vision, change the

[3] I explore further below what precisely I mean by a socially and culturally constructed, drawing on the work of Karen Barad (2003, 2007) to treat this as a material-semiotic performative practice.

world." This student was not alone in his vision. Some professors also suggested that this was an explicit goal of the program. As a department chair stated "students currently think that they are coming to learn computing for an IT career. I would like them to come to learn to take on the world." Although not all students were so certain of what they wanted to achieve with their degree, most commented on the scope and power of computing, and of the software and technologies they were learning to create, to disrupt inefficient industries and social relations and improve life.

Students are thus learning to translate and construct reality through models, code, algorithms, data structures, and programs. At the same time, I also explore how students' and others' practices, desires, aspirations, and beliefs in relation to computer science are suffused with affective, cultural, political, and gendered practices. Here, I add to the growing body of research in science and technology studies that has shown the ways that cultural norms and values are embedded in scientific and technological practices (Cohn 1987; Forsythe 2001; Haraway 1988, 1991a, 1997; Sandra Harding 1998; Helmreich 1998; Keller 1985; Traweek 1988; Wajcman 1991). More distinctively, however, this dissertation explores the implications of these technical renderings as they are experienced by students as part of computer science disciplinary norms, university and industry practices, and Singapore government discourses. I consider what various practices of rendering technical entail and show how students' lives and selves are "torqued" – "twisted, even torn" – by technical renderings (Bowker and Star 1999, 28). For example, I discuss in Chapters 5 and 6 how students are evaluated and judged according to such measures as grades, marked according to a bell curve, and their performance of qualities, such as passion. I also consider in Chapters 4, 7, and 8 how computer science students, and the knowledges and practices that they are learning, are rendered in terms of binary gender categories that structure and shape the ways students and others think about computer science, their identities and selves, and their possibilities and aspirations for the future, even as some students contest and reconfigure these renderings.

1.2 Technology and Gender

My approach to gender and technology examines their mutual constitution and production. I see technologies as socially, culturally, materially, and discursively shaping how people understand and experience gender and, correspondingly, gender as shaping the way technologies are designed, developed, and used. I focus on the construction of gender in intersection with other dimensions of identity, particularly (trans)national citizenship and technological expertise. My approach treats gender as performative; gender is constituted through its ongoing performance by both humans and nonhumans. Particular algorithms, teaching practices, textbook explanations, assignment problems, and so on, are thus also performative of gender. In this regard, the performance of gender norms is historically contingent and also continually negotiated.

This approach allows me to consider how the knowledges and technologies that students learn as part of computer science education (re)produce particular gender and disciplinary norms. For example, particular algorithms have gendered stories that frame their use. I explore what constructions of gender are reproduced through these framings; how these norms relate to national policy and transnational discourses; how students adopt these algorithmic stories; and the norms that students enact through their own re-tellings of these stories and performances of gender. I also consider how students' performances of gender – in intersection with professional identities, family and citizenship obligations, and personal aspirations – sometimes work to contest and reconfigure these norms.

Anthropologists have long attended to the significance of particular objects or nonhuman artefacts, as objects of exchange, for example (e.g. Mauss 1966). As Jenna Burrell points out, however, these were often considered in a generic sense, ignoring the specific materialities and agencies of the objects (Burrell 2012, 12–13).[4] Bruno Latour (1993) argues that this tendency to treat non-humans, such as technologies, as the passive objects of *human* symbolic and material creation and manipulation is tied to an assumed divide between "nature" and "culture," along with a host of other binary distinctions including subject-object and self-other. It is also tied to conceptions of humans as autonomous subjects – a notion that has been extensively critiqued by feminist and postcolonial scholars, among others, for the universalization of one imagined ideal subject, usually represented by white European maleness (Hayles 1999; Haraway 1997).

In contrast, science and technology studies (STS) and feminist technology studies scholars (FTS) have shown the complex imbrications of humans, machines, and animals found within the sciences and technologies and produced by them (Barad 1998, 2003; Bray 2007; Currier 2003; Downey 1998; Escobar 1994; Fischer 1999; Franklin 1995;

[4] For Marcel Mauss's analysis of gifts, for example, agency was situated in relationships between persons and objects, in terms of affects and effects produced for humans through exchange.

Haraway 1991b, 1997, 2003; Latour 1993; Suchman 2007b). Looking specifically at gender, FTS scholars have argued that gender and technology are socially and materially "co-produced" (Berg 1999; Cockburn and Ormrod 1993; Wajcman 1991). That is, gender performances and identities are intimately related to using and creating technologies, and the meanings, designs, and usages of technologies are similarly related to gender norms and values. FTS scholars have explored this co-production in action in relation to a variety of technologies and technological contexts (Berg 1996; Cockburn and Ormrod 1993; Faulkner 2000, 2001; van Oost 2003; Wajcman 2000).

Research has also shown how gendering becomes implicit in designs, as well as how the meanings and usages associated with technologies can take on a life of their own when used by new persons and in new contexts (Oudshoorn and Pinch 2003; Oudshoorn, Rommes, and Stienstra 2004). Implicit genderings and the ways they can subsequently influence or shape users' experiences and identities, however, has been a source of critique of much technology design, which has been shown to be based on masculine norms and values thereby hindering women's and others' usages of those technologies and reproducing gender inequalities (Berg 1999; Oudshoorn, Rommes, and Stienstra 2004; van Oost 2003). Early voice recognition software, for example, failed to recognize and respond to women's voices (Churchill 2010). More recently, the Apple Health App would allow users to track almost everything about their bodies, except for women's menstrual cycles (Quinn 2014).[5]

[5] These and several other examples are also briefly discussed in (Breslin and Wadhwa 2015, 2017).

To attend to these interconnections between subjects and objects and against a "modern" notion of a divide between nature and culture, Latour (1993) argues we need to recognize nonhumans – such as code, software, computers, and policies – as actors in their own right. He points out that such nonhumans are not just intermediaries that reliably and consistently transmit information. Instead, they "mediate": "they transform, translate, distort, and modify the meaning or the elements they are supposed to carry" (Latour 2005, 39). In short, they make a difference. As such, they are mobilized within multiple and extended networks of humans and nonhumans that act to constitute particular identities and realities. This attention to nonhumans as actors differs extensively from a common approach to agency, which relies on the idea of intentionality as emerging from the individual actor (Latour 2005, 71). Making intentionality a defining feature of agency is problematic for ignoring or denying the ability of nonhuman animals to have intention and for precluding a consideration of the ways that artefacts are themselves significant in contributing to the ideas and practices of others. Latour pointedly suggests we consider how "hitting a nail with and without a hammer… slowing down a car with or without a speed-bump… running a company with or without bookkeeping" are certainly not the same activities (Latour 2005, 71). Following Latour (1987, 1993; 1996; 2005), I attend to the specificities of computing technologies, including particular programming languages, modelling techniques, and other technologies that students create and use. I discuss my approach to agency in greater detail below.

My research weaves ethnographies of computing with critical literature on gender and computing, looking both at practice and subject-making in computer science in general and at how these relate to gender performance. Anthropological research on computer programming has explored processes of identity formation through learning and doing computing and coding. Much of this research has focused, in particular, on practices of resistance against commercial, corporate, and governmental regulation of software, code, and media, looking at free and open source software (FOSS) and hacking (Coleman 2013; Coleman and Golub 2008; Irani 2015; Kelty 2008; Takhteyev 2012; Turkle 1988, 2005). Christopher Kelty, for example, argues that FOSS development operates as a "recursive public" wherein programmers "build, control, modify, and maintain the infrastructure [the Internet, code, programs] that allows them to come into being in the first place and which, in turn, constitutes their everyday practical commitments and the identities of the participants as creative and autonomous individuals" (Kelty 2008, 7). Gary Lee Downey (1998) has also written an insightful ethnography exploring how engineering students learning to program and use CAD/CAM software adjust themselves to the requirements of this technology, while also negotiating the boundary that separates them from the machines they use.

However, while FOSS development and hacking are practices of growing significance, they are also only one small facet of computer science practice in terms of form and context.[6] By focusing on computer science education, my research explores the

[6] Over the last twenty years, FOSS practices have been adopted and promoted by many large multinational tech companies despite originally operating in part in resistance to logics of private property and secrecy entailed in corporate computing (Kelty 2008, 98–112),

more mundane facets of programming and computer science, those judged to be foundational by educators and curriculum designers, in addition to the explicitly political and exceptional. A significant component here is then exploring the way technological knowledge itself is constructed and acts. This approach also connects to STS research that has attended to the construction and enaction of representations or renderings as part of scientific knowledge production and practice (Coopmans et al. 2014; Keller 1985; Lynch and Woolgar 1990; Martin 1991, 1994; Myers 2015, 2014). Computer science renderings in these contexts are indeed political in that they govern and (re)produce particular realities that often unequally affect different groups, as suggested above, even if they are not part of the deliberate political economic projects of FOSS and other social and cultural movements (cf. Kelty 2008). I explore how various practices and nonhuman actors such as mathematical theorems, algorithms, data structures, coding problems, and textbooks work to frame and (re)produce particular natures and realities in ways that shape how students learn and do computer science, as well as their interpretations of reality and selfhood.

While ethnographic studies of FOSS and other forms of computing practice provide in-depth analyses of processes of identity formation and the interactions of humans and nonhumans, detailed discussion of the role of gender in computing is often limited (e.g. Coleman 2009; Kelty 2008; Takhteyev 2012). At the same time, when gender is explored as part of the analysis of computing cultures and practices, it often becomes the sole focus of discussion (e.g. Nafus 2011).[7] As discussed above, gender is an integral facet in designing, developing, producing, and using technologies, including those related to computing. Focusing on gender and technology co-production, however, often comes at the expense of exploring other often intersecting forms of difference or inequality, such as race or class (Bray 2007, 47–48). Gendered and other distinctions and inequalities do not operate in parallel, but as a multiple intersection whereby "race, class, and national hierarchies are themselves everywhere constructed in gendered ways, and gender divisions are established with 'communal' materials" (Tsing 1993, 18). As such, feminist scholars have argued for and shown the necessity of exploring gender in intersection with other forms of identification such as race, class, age, and sexuality (Crenshaw 1989, 1991; Tsing 1993).

Catharina Landström (2007) has additionally argued that much research looking at gender and technology co-production is slanted towards addressing the social construction of technology, leaving gender as an implicitly heteronormative and stable "black-box." Some scholars have explored different forms of masculinities that are expressed in interaction with technologies (Mellström 2003, 2002; Wajcman 1991), based on "toughness and practical skills" versus "intellectual acuity," for example (Bray 2007, 41). Yet, the focus largely remains on showing the masculine or feminine associations of technological design and practice, rather than questioning the construction of gender itself and the role technologies play in that construction and in the making of

[7] Sherry Turkle (2005) is an exception in this regard, providing a detailed discussion of multiple facets of identity formation, including gender. Yet, her analysis of gender is framed in terms of essential differences between women and men in relation to forms of sociality and intimacy with machines.

subjectivities more generally. This is clearly seen, for example, in Vivian A. Lagesen's (2008) work on Malaysia, which seeks to explain the presence rather than absence of women in computer science. She points to ideas that women have equal skill and ability in computing, as well as religious and gendered valuations of indoor spaces, that promote the presence of women in the discipline. Her discussion presents alternate performances of femininities, but assumes a gender binary defined by the presence or absence of women. My research extends and complicates these questions, exploring the situated *performance and creation* of gender associations, relations, and boundaries that constitute students' subjectivities in intra-action with computing technologies such as code and algorithms, as well as computer science curricula, government policies, and transnational discourses relating to the tech industry.

1.2.1 Posthumanist Performativity

In order to explore the intersectional processes of technology-identity co-production, I follow Karen Barad's approach to looking at material-discursive production of difference in/of the world, an approach she terms posthumanist performativity (Barad 2003, 2007). While Latour argues for considering the actions of nonhumans, his attention to such subjects/objects treats them as stable entities to be dispatched, assembled, or reassembled in the mobilization of heterogeneous networks (Latour 2005). In other words, as entities, the boundaries of humans and nonhumans are left unquestioned. Combining insights from physicist Neils Bohr, along with Judith Butler and Michel Foucault, Barad extends and reconfigures Latour's insights by arguing that humans and nonhumans are mutually part of and produce the material-discursive "becoming" of the world (Barad 2003, 822).

Barad uses the concept of "apparatuses" as more than a particular tool or piece of equipment for scientific measurement, akin to the concept of an "apparatus" of government. An apparatus, following Barad, entails the dynamic (re)arrangements of humans and nonhumans that enact discursive and ontological realities, which include particular boundaries (Barad 2003, 816). "Apparatuses are particular physical arrangements that give meaning to certain concepts to the exclusion of others; they are the local physical conditions that enable and constrain knowledge practices such as conceptualizing and measuring" (Barad 2007, 147). The concept of apparatuses relates to Foucault's notion of discourse, but insists on the materiality of such practices that include both humans and nonhumans and their mutual constitution/division (Barad 2007, 147–48). Barad's focus on the contingent production of boundaries allows for a consideration of the ways that particular genderings, along with multiple other forms of embodied performance and boundary creation, are produced and enacted.

I follow Judith Butler (1988, 1993, 1999) and Barad (2003, 2007) in seeing gender as constituted performatively, where gender is an iterative process of doing and becoming. Extending Butler's insights, however, Barad argues that performativity is not just about the constitution of *human* bodies or enacted through *human* agency (Barad 2007, 145–46). Rather, gender and performativity are enacted and constituted in dynamic intra-action with material-semiotic apparatuses. In this regard, Raewyn Connell argues that "reproductive biology is socially dealt with in the historical process we call gender"

(Connell 1987, 79). Here, “dealt with” entails an intra-active process of socio-materially constituting both gender and reproductive biology.

As such, biological and gendered intra-actions entail agential-cuts, following Barad (2003, 815), which draw attention to the performance of boundaries as they are drawn and redrawn by and among particular assemblages of humans and nonhumans. Using Barad’s notion of “cuts” I attend to the specific boundaries that are drawn among students, computers, code and software, as well as genders and nationalities/citizenships. For example, as mentioned above, I explore the intra-actions that constitute the division between masculine and feminine while simultaneously enacting exclusions (such as other, multiple, or intersectional genders), as part of the intra-actions among professors, teaching practices, algorithms, text-books, and students.

My analysis centres on the concept of intra-action, but understood as operating within the context of specific apparatuses and as enacting agential-cuts. Barad’s approach to intra-action is significant and useful for understanding and exploring the “making” of computer scientists in how it challenges us to keep in view the infinite multiplication of realities that *could* exist in any given instance of students’ encounters with technologies, curricula, and policies, and reminding us that things could always be different (Barad 2010, 250–52). This challenge is necessarily impossible to satisfy in full, thereby pointing to our own partiality and situatedness as we are embedded in particular apparatuses (Haraway 1988). More specifically, considering how human and nonhuman intra-actions create particular realities and exclude others points to how code, programs, computer science knowledge, and computer science education taken as a “whole” contribute to creating particular selective traditions and therefore to shaping the possibilities for computer science perspectives and subjectivities (Williams 1977, 115–20). My attention to rendering technical as a social and cultural practice is then not about particular representations of an independent reality, but rather a performative and diffractive practice that produces (but does not determine) particular social and material realities (Barad 2003, 2007; Haraway 1994).

1.3 Education and Subject-Making

My approach to education sees it as interconnected with the workings of gender, race, citizenship and state power, class, and so on, while taking these categories as unstable and at the same time irreducible to one another. As such, there are multiple processes through which student subjectivities are cultivated. For example, I discuss how passion is promoted among students simultaneously as part of both the explicit and implicit curricula of the computer science discipline, the university, and the Singaporean education system, as well as through national policy promoting entrepreneurship to develop Singapore's knowledge-based economy, and as part of transnational and neoliberal discourses and practices of the tech industry. Moreover, I consider how students may cultivate themselves in particular ways. For example, students often display passion through personal projects and intimate knowledge pursued outside of class time.

The workings and interactions of these varying actors, discourses, and practices, however, often have conflicting or paradoxical implications in relation to gender roles and norms, (trans)national citizenship, and the production of performance of expertise. In turn, such conflicting intra-actions can open up space and possibility for students and others to reconfigure the meanings and boundaries of particular norms, values, and behaviours as they enact different relationships among gender, citizenship, and entrepreneurial passion as part of their education towards becoming computer science students. For example, discourses of passion promoted by the Singaporean government as part of a national project are tied to ideas about transnational movement and personal mobility and a valorization of Silicon Valley as *the* model for entrepreneurship and innovation, while also necessitating freedom from social obligations, other interests, and other claims to personal time. Some students work to make themselves into passionate persons by becoming mobile entrepreneurial subjects and working elsewhere, particularly Silicon Valley, but in doing so also challenge state policies that seek to promote and develop the local tech industry.

Anthropology has a long history of studying education, looking both at forms of cultural education and at socialization in schooling institutions. Early social theorists, for example, looked at processes of cultural transmission and how children, as well as anthropologists, learn to "think, act, and feel appropriately" (Spindler and Spindler 2000, 142). Ethnographies of cultural education also showed what and how norms, beliefs, and practices are learned and incorporated by people in different cultural groups (Briggs 1998; Fortes 1938; Mead 1928). Similarly, early ethnographies of schools show their role in implicitly socializing students into norms, values, and skills to be functional members of society (Dreeben 1968; Jackson 1990[1968]). However, many of these studies rely on an idea of "culture" as bounded, coherent, and singular wherein people are harmoniously and unproblematically socialized and enculturated – an approach that has been heavily criticised in anthropology since at least the 1980s (Abu-Lughod 1991; Clifford and Marcus 1986; Gupta and Ferguson 1992; Lave 2011; Rosaldo 1993; Wolf 1997, 1999).

As a result, scholars have also explored how class, gender, and racial inequalities are enacted and reproduced in education (e.g. Holland and Eisenhart 1990; Khan 2011; MacLeod 2009; Weis 1985; Willis 1981). There is also a significant body of literature that explores the expression of biases in relation to gender and race in science, technology, engineering, and mathematics (STEM) fields; this research shows how girls and women and ethnic minorities are judged differently, primarily relative to white men, and how qualities associated with femininity are devalued in relation to masculine performances of technical prowess (Ashcraft and Blithe 2010; Ashcraft, Eger, and Friend 2012; Blickenstaff 2005; Carrigan 2017; Ensmenger 2015; Margolis and Fisher 2002). This research is discussed in greater detail in Chapter 7. The reproduction of ideological norms and values are often inextricable from broader systems of inequality and the maintenance of dominant privileges, such as those of patriarchal, state, and class interests, albeit in often complex ways (Apple 2004; Bourdieu 1967; Bourdieu and Passeron 1977; Bowles and Gintis 1976; Fife 1991; Holland and Eisenhart 1990; McRobbie 1978; Willis 1981). Pierre Bourdieu's well-known exposition of the significance of "cultural capital," for example, explores how certain behaviours and sensibilities are validated in school systems, implicitly valuing dominant-class tastes, speech, dress, and consumption patterns and showing the ideological and value-laden facets of schooling and other social and cultural practices (Bourdieu 1984; Bourdieu and Passeron 1977).

The concept of “hidden curriculum” has been employed, particularly in relation to state-run primary and secondary schooling systems, to explore how education works to shape student subjectivities in particular and unequal ways (Apple 1980, 2004; Fife 1991, 1994; Giroux and Purpel 1983). As suggested by Wayne Fife, “hidden curriculum refers to all the ways that the verbal and physical organization of education affects the production of an ethos or form of cultural consciousness among students” and it is performed concomitantly with the overt curriculum (Fife 2005, 46). Michael Apple points to three areas, in particular, where hidden curriculum is enacted: (1) the daily regularities of schooling – its organization of time and space, material objects in classrooms and hallways, students’ interactions with one another, and so on; (2) the curricular knowledges that are promoted, such as the divisions of disciplines and forms of assessment; and (3) the actions and meanings performed by educators, including physical appearance and interactions with students (Apple 2004, 12–13).

G. Bergenhenegouwen (1987) suggests that hidden curriculum might better be referred to as “implicit education,” since these facets of education are not necessarily hidden and their effects are often unintentional. Regardless, studies of the reproduction of class hierarchy do point to the more nefarious effects of some hidden curricula (e.g Willis 1981). “Implicit education” points to all aspects of education that are not explicitly a part of the official or overt curriculum. At the same time, in relation to computer science education, many of the values and behaviours that I saw emphasized by professors, such as the promotion of passion and entrepreneurialism – which were only in a few cases included in the technical skills and knowledge that made up the overt curriculum – were very explicit. As Marina Gair and Guy Mullins point out, the hidden curriculum is often "hidden in plain sight," but taken for granted and unmarked (Gair and Mullins 2001, 23). I use the concept of hidden curriculum for inquiring into the multiple norms, values, and behaviours that are part of computer science education in Singapore. I discuss here the relationship of hidden curricula to state power and to performative agency.

1.3.1 State Power and Hidden Curricula

Scholars have explored implications of the Singaporean government's policies on education – as well as reproduction, housing, immigration, the Central Provident Fund, multiracialism, and gender, among others – for governing the city-state's population, some of which I discuss in subsequent chapters (Barr 2006; Goh 2008; Gopinathan 2001; Heng and Devan 1995; Hill and Lian 1995; Ho 2012; Huat 1997; Kho 2013; Koh 2014; Li 1989, 1998a, 1998b; Liew 2014; Olds 2007; Ong 2005; PuruShotam 1998; Sun 2012; Tan 2007; Yang 2014b). In Singapore, policies and practices of governance are particularly significant given that scholars have suggested the state be construed as "developmental productivist," "aimed primarily at developing citizens' human capital to strengthen their economic productivity" (Sun 2012, 2,9). That is, the Singaporean state strives to manage and produce its citizens and subjects in particular, and sometimes unequal ways, with the goal of economic production and the reproduction of power relations (Ong 2006). As similarly shown in research in the United States and elsewhere, national education in Singapore works to produce Singaporeans as national citizens and workers through aspects of the overt curriculum such as mandatory citizenship courses, as well as via hidden curricula that foster Singaporeans as resources for the state, but differentiated in relation to particular gender, class, and racial roles and differences (Comunian and Ooi 2015; Gopinathan 2001; Ho 2010, 2012; Kho 2013).

In this regard, Cris Shore and Susan Wright argue that policy works via "complex processes by which policies not only impose conditions, as if from 'outside' or 'above', but influence people's indigenous norms of conduct so that they themselves contribute, not necessarily consciously, to a government's model of social order" (Shore and Wright 1997, 6). Kenneth Paul Tan, for example, explores how the Singaporean government explains the city-state's (and the government's) social and economic success through discourses of "good governance" (referring primarily to the government's approach of pragmatic administration) and meritocracy (promoting the idea that "the best" people are working in the government) (Tan 2007, 3). These claims are supported by Singapore's historical developmental achievements, which have justified the continuous rule by the People's Action Party since 1959 and the extensive powers of governance enacted by the state, and produced citizens accustomed to and supportive of government intervention and administration (Tan 2007, 3). The government thus shapes the values and possibilities of its citizens, producing what are constituted as "normal" modes of living through educational curricula, laws, policies, media, and administration.

At the same time, while policies relating to gender, education, and technology, among others, undoubtedly influence and govern Singaporean citizens and students in Singapore, these should not be seen as fully determining the values and behaviours of citizens and subjects in the city-state. In particular, Mary Margaret Steedly warns against treatments of the state as a coherent and all-powerful force, which she suggests has led to factoring upward ideas of an "organic culture" to the state (Steedly 1999, 443–44, see also Gupta 1995; Mitchell 1991; Ong 2006; Scott 2009; Trouillot 2003). A variety of social-science research on Singapore focuses on government policy and economic progress and, in doing so, often depicts the Singaporean state as a largely coherent and omnipresent, if not infallible, force (Arun and Yap 2000; Heng and Devan 1995; Ho 2012).[8]

Yet, Shirley Sun (2012) has shown how attempts by the Singapore state to promote population growth according to its vision of the family are frustrated by individual and family decisions in Singapore to raise fewer children due to the competitive climate, their economic insecurity as labourers, and the limited and non-egalitarian support offered by government policies. Similarly, I discuss in Chapter 8 how the state's attempts to cultivate entrepreneurial and risk-taking citizens and subjects have paradoxically been frustrated by those citizens' entrepreneurial desires to pursue risks elsewhere, particularly in Silicon Valley. Some laws are also not actively enforced, and the government has contradictory or paradoxical goals and policies in a variety of cases.[9] Scholars have pointed, for example, to the government's contradictory and ambivalent

---

[8] This treatment is seen in Yao's (2006) discussion of Singapore culture as produced by a state of anxiety and "excess" reproduced in its legal code, forms of punishment, international relations, and even in Singaporean's artful lying known as "talking cock."

[9] In a more general sense, I was always surprised during my fieldwork by the variety of ways that minor laws, regulations, and policies were casually disregarded by many in Singapore, although there were many others that were strictly followed. Despite fines for jaywalking, guidelines against keeping cats in Housing Development Board (HDB) flats, rules against renting out flats for less than 6 months, and laws against downloading copyrighted material such as movies or TV shows, to name a few examples, I saw or heard of all of the above being practiced at one point or another with little to no concern or recognition of policies or laws. At the same time, other rules, such as not eating, drinking, or bringing durians on the Massive Rapid Transit (MRT) were strictly followed. On the few occasions where I did see a person taking a sip of water on the MRT, for example, they did so quickly and discretely and usually because of a cough or other discomfort.

approach to cultural policy that promotes "creativity, non-conformity, and risk-taking as drivers of the knowledge economy" even as "the state's pragmatic endorsement of the conservative 'moral majority' has led to the silent vilification of sexual minorities, in part through the maintenance of anti-gay laws and policies" (Liew 2014; Tan 2003).

Singapore's focus on developing its knowledge economy has additionally fostered multiple connections with international institutions and scientific and technological communities of practice, leading to a "reimagining of the nation as a platform in a chain of knowledge production" (Ong 2006, 179).[10] International connections are fostered as part of the education system, especially in university, where students are encouraged to go on exchanges to other countries, universities hosts numerous exchange students from other countries, and many co-curricular programs involve international travel, at least within the region. Additionally, practices in the computer science discipline are intimately and explicitly tied to the interests and practices of multinational corporations and professional organizations such as the Association for Computing Machinery (ACM). Computer science values and behaviours originating from outside of Singapore, largely from the US, are fostered as part of Singaporean national policy and enacted through computer science education at Temasek University. As discussed above, these transnational connections are

[10] Multiple other university and institutional connections are being created such as: The Singapore MIT Alliance, a joint program between the National University of Singapore and the Massachusetts Institute of Technology with faculty from both universities (Ong 2006, 181); the Duke-NUS medical school focusing on biomedical education, research, and development; Yale-NUS College, a liberal arts college in Singapore; the Lee Kong Chian School of Medicine, a medical training partnership between Nanyang Technological University and the Imperial College of London; and the Global Alliance of Technological Universities a network initiated by Nanyang Technological Institute and comprising nine other institutions focusing on science and technology including Caltech, ETH Zurich, and Carnegie Mellon University, among others.

part of a broader movement towards neoliberalization in many places around the world, including Singapore, (re)producing the value of entrepreneurialism and reconfiguring education and labor based on the needs of transnational industries and markets (Harvey 2005, 2006).

In exploring the "elusive culture" – the fluid, ambiguous, and multiple constitution of identity – among high-school students in Toronto, Daniel A. Yon (2000a, 2000b) effectively points to the importance of exploring possible multiple and contradictory hidden curricula in schools, although he does not use the term. He argues in relation to considering schools as discursive spaces:

> Such ideas push further the nonsynchronous workings of race, gender, and class as they pay attention to the instability of these categories… They open educational ethnographies to the surprises, contradictions, conflicting desires, ironies, and ambivalence of everyday life of students and teachers, inside and outside the classroom, and they do so without trying to domesticate the incongruities for the sake of theoretical coherence (Yon 2003, 424).

Levinson et al. similarly suggest looking at the "cultural production" of "educated persons" (Levinson, Foley, and Holland 1996). This approach expands on Willis's conception of cultural production, discussed further below, which has been critiqued for overdetermining and overemphasizing the structural influence of class. Levinson et al. seek to broaden the consideration of "educated person" to encompass various social and cultural practices within and outside of schools:

> We see schools as sites of learning which hegemonic groups, in alliance with consolidating states and/or expanding bureaucracies, often utilize to form certain kinds of subjectivities… Teachers play a crucial role in enforcing such models of the educated person, though they may in practice challenge or ignore models bequeathed them by policymakers and politicians… Finally, students and their families exercise agency in responding to the practices and discourses of the school. They, too, engage in the cultural production of practices and discourses

> which accommodate, resist, or otherwise adapt to the dominant school definition of the educated person (Levinson, Foley, and Holland 1996, 24).

Thus the making of student subjectivities happens at various levels and in relation to many different actors.

In his research on schooling in Papua New Guinea, Wayne Fife discusses the potential for "secondary forms of hidden curriculum in schools," correlating to waning "traditional" community interests that sometimes disrupt the dominant messages of "order, authority and hierarchy" in local schools, if only temporarily, pointing to the influence of shifting dominant actors, values, and social organizations (Fife 1992, 219). Fife argues for the importance of exploring these secondary messages in understanding educational experiences, although he argues that such secondary forms of hidden curriculum are "weak and less patterned" than dominant messages (Fife 1992, 215, 220). Recent research on higher education additionally suggests that, in these contexts, the "less densely codified curricula provide a landscape for potential multiple hidden curricula to exist" (Cotton, Winter, and Bailey 2013, 193). I discuss in Chapter 7 contradictory approaches by the government to technical education and labour participation for women, where policies encourage women to join the labour force and technical fields while simultaneously limiting their enrollment in educational programs and treating as primary their roles as mothers and home-makers. Chapters 4, 6, and 7 also consider how computer science education and government policies in Singapore intra-act with transnational discourses and practices relating to the tech industry and computer science discipline.

Michael Apple suggests that the hidden curriculum works to present a "selective tradition" – a partial reinterpretation of how things are or have been – as reality amidst infinite other possibilities (Apple 2004, 5). Here, Apple draws on Raymond Williams' (1977) and Antonio Gramsci's (1988) elaborations of "hegemony," which "refers to an organized assemblage of meanings and practices, the central, effective and dominant system of meanings, values and actions which are lived" to explore how consent for unequal power relations is reproduced through schooling (Apple 2004, 4). Reconfiguring the focus on dominance to Barad's consideration of material-discursive apparatuses, I use the concept of hidden curriculum as potentially multiple and contradictory performances of selective realities, enacted through particular dominant discourses and practices as well as through the changing and embodied practices of students, faculty, technologies, administrators, and policies, among others. As Jean Lave suggests, "learning, knowledgeability, skillfulness, whatever else they might be, are always only part of ongoing social arrangements and relations" (Lave 2011, 3). In other words, hidden curricula are part of ongoing intra-active social-material performances and practices involving humans and nonhumans that enact and produce particular boundaries and realities in relation to technical renderings, expertise, student subjectivities, gender, and citizenship and (trans)nationalism. I thus seek to both recognize the ways students and professors contest and reconfigure gender norms, state policies, and computing world renderings, while also recognizing the influences, structures, and discourses that powerfully shape how they act and enact.

1.3.2 Performative Agency

Paul Willis's (1981) well-known study of "the lads," a group of young working-class white men in the United Kingdom, highlighted the importance of attending to students' own interpretations and practices in relation to education. Willis explores forms of cultural production by the lads as "the creative, varied, potentially transformative working out… of some of the fundamental social/structural relationships of society" (Willis 1981, 137). In doing so, he shows how they make rational decisions based on "penetrations" of the ideas and material reality underlying the schooling system and what it offers. In particular, they perceive and reject the value of qualifications (the jobs available are mostly working class regardless of the schooling they achieve); the idea that careers/jobs offer a source of individual meaning (manual labour is largely interchangeable); and the realistic possibility for upward mobility (individuals may be upwardly mobile, however, the working class as a class inevitably remains in a situation of structural subordination) (Willis 1981, 128–30).

The clinch for the lads is the "limitations" imposed on these realizations in relation to divisions between manual and mental labour and gender. These redirect their insights towards a valorization of manual labour as an expression of masculinity, leading them to willingly choose working class jobs (Willis 1981, 148). Willis's study, among others, points to the importance of considering multiple lines of difference and dominance in the operation of a hidden curriculum, and educational practices more broadly (MacLeod 2009; Willis 1981), dovetailing with feminist scholarship, discussed above, which argues that class must be explored in intersection with gender and race, along with other forms of difference and inequality (Crenshaw 1991, 1989; Gilroy 1991).

Willis's approach to the relationship between individual agency and structural determination is quite open-ended. As Willis argues of the creativity and agency of working class lads and their development of a counter-school culture:

> We must accept a certain autonomy of the processes… which both defeats any simple notion of mechanistic causation and gives the social agents involved some meaningful scope for viewing, inhabiting and constructing their own world in a way which is recognisably human and not theoretically reductive (Willis 1981: 172).

He also further insists "that cultural forms cannot be reduced or regarded as the mere epiphenomenal expression of basic structural factors… they are part of a necessary circle in which neither term [culture/structure] is thinkable alone… The cultural is part of the necessary dialectic of reproduction" (Willis 1981: 174). Indeed, many Marxist approaches have insisted that structure and super-structure are intimately intertwined (Donham 1999; Gramsci 1988). However, Willis gives ultimate explanatory power to *class*-based culture, even if this culture is expressed in interaction with structures of gender in terms of patriarchy. As Dorinne Kondo argues of Willis's approach: "the society he constructs possesses the predictability of a well-oiled – if somewhat complicated – reproduction machine. No matter what people do, they manage to reproduce the capitalist system" (Kondo 1990, 222). More generally, many approaches drawing on the concept of hidden curriculum rely on a correspondence between the hidden curriculum and dominant structures of power, whether based on class, gender, or race.

Rather, as Julie Bettie clearly shows in her ethnography of girls in their last year of high-school in central California, "it is not the case that race and gender are mere ideologies that mask the reproduction of class inequality; they are organizing principles in their own right, processes that are co-created with class" (Bettie 2000, 29). Moreover, as indicated by Yon above, ethnographies of education should work against "trying to domesticate the incongruities for the sake of theoretical coherence" (Yon 2003, 424). My approach thus works to unravel and explore the intra-actions among students, curricula, citizenship, gender, disciplinary norms, and state policies, not to produce a whole coherent "machine," but as a multiplicitous cyborg where "permanently partial identities and contradictory standpoints" persist (Haraway 1991b: 154). In doing so, I seek to recognize the "creative, varied, potentially transformative" contributions of students in these intra-actions, following Willis, while also exploring the ways that students are "twisted, even torn" as they are "summoned" to become technological experts, gendered persons, and (trans)national citizens through their intra-actions in and with computer science education (Bowker and Star 1999, 28; Mahmood 2005, 28; Willis 1981, 137).

Following Barad, as well as Michel Foucault and Judith Butler, I see agency as enacted not through resistance or opposition to – or even "penetrations" or understandings of – some monolithic force such as patriarchy or capitalism, but through performative iterations and intra-actions. As Michel Foucault has argued:

> In effect, what defines a relationship of power is that it is a mode of action which does not act directly and immediately on others. Instead, it acts upon their actions: an action upon an action, on existing actions or those which may arise in the present or the future… it forces, it bends, it breaks on the wheel, it destroys, or it closes the door on all possibilities (Foucault 1982, 789).

As Amy Stambach points out in her introduction to the edited collection on *Anthropological Perspectives on Student Futures* (Stambach and Hall 2017), education "structures possibilities for imagining a future within a set of possibilities full of hope and opportunity, and it discursively empowers people with authoritative responsibility but it cannot place them into motion within any stable or predictable horizon" (Stambach 2017, 13). That is, education produces agency, intra-action, and the structuring of possibilities, but these agencies and structurings do not produce determinate futures. In this regard, "agency is not an attribute but the ongoing reconfigurings of the world" (Barad 2007, 141, 178).

Students' practices of self-discipline and self-formation are also significant to consider in this regard, to explore how their choices and aspirations "bend" with the power enacted by disciplinary, state, and transnational discourses and norms (Bryant 2001, 2005; Kelty 2008; Kondo 1990; Mahmood 2005). Rebecca Bryant argues that education in Cyprus is part of a self-conscious endeavour towards becoming a moral person and a good citizen, a process that cannot be explained in full by theories of class and cultural reproduction in and through education (Bryant 2001, 587, 590–91). Saba Mahmood (2005) also explores the ways that women involved in the Islamic revival movement in Egypt practice multiple forms of self-discipline and self-formation in order to make themselves into pious subjects. Mahmood additionally discusses how such practices of self-making are simultaneously processes of "subjectivation," following Foucault (1990, 1997a).

For Foucault (1990, 1997a) the operation of power and governance "summons" a subject to constitute him/herself in particular ways and according to particular moral codes, which also limits the field of possibilities for action and thought (Mahmood 2005, 28). Moreover, this governance enacts a "conduct of conduct" such that certain questions (or ways of framing them) are made possible, while others excluded (Foucault 1982,1991; Wolf 1990). I explore this process in greater detail in Chapter 7 in considering the "problem" of women in computing. This summoning entails embodied work on oneself that is never completed, to meet desires and expectations of particular moral and ethical codes (Butler 1999; Freeman 2014; Mahmood 2005).

While Foucault suggests that the opposite of agency is passivity, Mahmood explores how passivity, restraint, and subordination too can enable these women to effect change in the world – to intra-actively reconfigure and reconstitute the boundaries of women's political and religious agency (Foucault 1982, 789; Mahmood 2005). Lauren Berlant (2007) additionally suggests there is a kind of "lateral agency" in mundane episodic activities, such as eating, that provide a kind of intermission from (even as and because these activities are necessary to) the experience and ongoing project of living. Based on research in a shelter in downtown Boston, Robert Desjarlais similarly points out that "the elements of everyday life in the shelter are distinct from the elements of 'experience' as the process is typically understood… as an inwardly reflexive process that proceeds, coheres, and transforms through temporally integrative forms" (Desjarlais 1996, 70). Rather, experience in the shelter is lived through a series of episodes entailed in "struggling along" (Desjarlais 1996). While projects of passion, which I discuss in Chapter 6, are intimately intertwined with the kind of cohesive and transformative processes contrary to and distinct from "lateral agency" and "struggling along," attending to students' and others' daily routines and mundane acts of life and living allows for recognition of multiple kinds of agencies.

In sum, this dissertation explores the ways that "norms are lived and inhabited, aspired to, reached for, and consummated" (Mahmood 2005, 23). I show how students work to make themselves into passionate persons, submitting and consenting to – but also dynamically embodying, inhabiting, and fostering – norms, behaviours, and values constituted in intra-action with, but not fully captured or determined by, computer science disciplinary knowledges, practices and curricula, gender norms, Singaporean state policy and governance, and capitalist and tech-industry interests and discourses. In this regard, capacity for action (agency) – even passion – is enabled through subjectivation and subordination (i.e., through intra-action) (Barad 2007, 218–20; Mahmood 2005, 29).

Like Donna Haraway's cyborg world, which is simultaneously about "the final imposition of a grid of control on the planet" and potentially "lived social and bodily realities in which people are… not afraid of permanently partial identities and contradictory standpoints," I work to hold these two perspectives together (Haraway 1991a, 154). Haraway further points out that "the political struggle is to see from both perspectives at once because each reveals both dominations and possibilities unimaginable from the other vantage point" (Haraway 1991a, 154). Moreover, conflicting intra-actions among varying and competing material-discursive phenomena open up space and possibility for reconfiguring boundaries and processes of subjectivation (Barad 2007, 213–14). As such, I also explore how some students contest and reconfigure norms relating to passion and gender as part of the intra-actions of affect, gender norms and roles, and personal aspirations.

1.4 Outline of Chapters

The next chapter (Chapter 2) outlines the methodologies that I used during my fieldwork. I also explore particular challenges I faced while conducting this research in relation to the conceptual and practical role that the idea of "community" plays as an organizing principle in anthropological fieldwork, and the relationship between gender norms and personal subjectivity in conducting research on gender and subject-formation.

Chapters 3 – 8 constitute the substantive chapters of this dissertation. Chapter 3 explores the ways computer science knowledge is taught and learned, focusing on early-year courses when students first learn the basics of programming. I explore how students learn practices of rendering technical as they learn to use and create computing "worlds." I consider how students are thereby initiated into the *sacra* of computer science thought and practice. Chapter 4 extends this consideration of computer science knowledge and practice, showing how heteronormative gender binaries are embedded in the ways code, algorithms, data structures and other aspects of computer science are conceptualized, learned, and taught. I also explore some of the transnational connections that constitute various aspects of computer science education – text books and algorithmic stories, for example – showing their selective origins, as well as how some performances of computer science knowledge work as transnational traditions that reproduce gender norms.

Chapters 5 and 6 focus on further aspects of the overt and implicit curricula in computer science. Chapter 5 considers how students are rendered into comparable and competitive persons. Students are cultivated as independent and autonomous but "networking" individuals who actively and instrumentally seek out knowledges, technologies, and resources. In this regard, students also learn to assess and judge code, algorithms, and programs critically even as students themselves are measured, evaluated, and compared. Students are thereby summoned to compete for grades and, ultimately, for future employment. Chapter 6, additionally explores the cultivation of "passion" among students in computer science, showing how the intersection of practices by the student hacker group at Temasek University, Singaporean government policy, and transnational tech industry discourses work to foster entrepreneurialism among students. Students in the student hacker group, who both promote and work to meet the calls of government and industry to become the "techiest of the techies," are accorded status and privileges reflecting the hegemonic way of being/becoming a "good" computer scientist. The promotion of passion, however, works to (re)produce and privilege neoliberal forms of selfhood and work and manufacture a continual sense of deficiency in terms of labour, and of the quality of people and places embodying that labour.

Chapter 7 returns to considering the role of gender, building discussions of the normalization of gender binaries, practices of rendering technical, and the influence of transnational discourses to explore how these frame the possibilities for thought for both research on and enactments of gender in computer science. Chapters 7 and 8 complicate assertions by many that "gender actually doesn't matter" in computer science and in

Singapore, and Chapter 8 explores the conflicts and paradoxes produced through the intra-action of gender and computer science norms. While students' identities and selves are torqued by these norms, and the values and behaviours they entail, I also consider the ways some students complicate and reconfigure heteronormative gender binaries, and, in the process, what it means to be/come a "good" computer scientist.

Chapter 2 : Methodology

I started writing this chapter during a bout of late night insomnia as I prepared to travel to Singapore for a brief return visit. My mind was unfortunately active for the late (or early) hour, as I worried about reconnecting with participants, discussing my research "findings" with them, and simply whether I would be able to function with the jetlag, heat, and humidity. My concern bubbled up from my memories of a difficult fieldwork experience: when every time I felt like I was gaining some traction or momentum it became midterm or exam time and students would disappear to focus on their studies; when doing research in an academic department of a couple thousand people was not how I envisioned fieldwork in a "community" of the same size; when my personal perspectives and politics were often at odds with those of my participants; and when my body's ability to deal with the heat and humidity would frequently fail leading to repeated visits to the doctor and even minor surgery.

Methods chapters and ethnographic introductions are replete with tales of anthropologists' intrepid adventures and struggles to "enter" the field and gain the trust and acceptance of the "natives," thus establishing their ethnographic authority (Crapanzano 1986; Gupta and Ferguson 1997, 12–15). Typically, periods of self-discovery and deep inspiration are meant to follow and the struggles become repaid with meaning and triumph (e.g. Geertz 1973, 412–54). This chapter aims to give a glimpse of my perspectives and my experiences while doing fieldwork, while also seeking to side-step some of these common tropes. I discuss the how and why I chose this topic, my experiences during fieldwork, and post-fieldwork reflections and writing decisions.

As feminists have argued, the personal is political (Hanisch 1970, 2006). This personal account is thus meant to provide some understanding of the inevitably political decisions (both implicit and explicit) that I have made in writing this dissertation – the decisions about the stories that I have chosen to tell. This story is inevitably partial. I have pages of field-notes, interview transcripts, collected documents and other items, and many photos that are not included in my discussions. Similarly, the narratives that I collected are themselves inevitably partial and contextually situated. As one professor requested that I point out, interviews and statements are also the perspectives and opinions of particular individuals and not necessarily reflective of the institutions that they are part of, which enact their own (multiple) agendas, goals, and values. Some of these perspectives – their meanings, their implications, and the ways they compare and sometimes compete – are the very subject of this dissertation.

In focusing on personal experiences here, I also hope to make explicit the mundane challenges and struggles, along with the successes, that were part of my fieldwork. While methodology texts and chapters often provide a general overview of fieldwork practices, details are less often provided about how initial contacts were made and administrative details like visas and housing were arranged. As these are key facets of starting new projects and doing fieldwork, I offer some of the specifics of my experiences to lessen the frictions for those who want to follow similar paths. This information also provides further insight on my positionality during fieldwork, which, in turn, conditioned what I was able to learn.

2.1 Studying Computer Science

My decision to study computer science education was deeply personal, given that I majored in computer science as an undergraduate student. My turn to anthropology reflects my sense that I did not comfortably fit the goals, interests, and future possibilities offered by the Computer Science discipline. I was a successful student and employee, participating in an extensive co-op program where I worked at several tech-related jobs. Yet, I struggled with the ways doing computer science seemed to demand becoming intimately and thoroughly intertwined with and dedicated to computers, both in terms of time and engagement (Downey 1998; Turkle 2005). In addition, at the time the career opportunities seemed to me to be technical, boring, and constrained within limited frameworks. I return to these issues throughout this dissertation.

But, what initially sparked my desire to study computer science ethnographically was my participation in the "Task Force for Gender Equality in Computer Science" as an upper-year undergraduate student. The taskforce was created to help promote and resolve issues of gender balance in the computer science department at the University of Waterloo. The report we produced was based on literature reviews, task force members' experiences, and a survey administered to undergraduate students (Task Force on Gender Equality at UW-CS 2007). The survey, while meant to provide an understanding of students' perspectives on studying computer science in relation to gender, was primarily based on multiple choice questions. The results could thus be statistically analyzed, but the possibilities for students to convey their experiences were limited to the tiny boxes and numerical measurements of their satisfaction with various facets of the discipline and department. Faculty and student members of the taskforce also had many other commitments, limiting their ability to pursue questions of gender in depth.

From my discussions with fellow students there seemed to be much more to explore about students' struggles, frustrations, joys, and challenges. I also had my own experiences in relation to gender: a classmate saying I only got hired for a co-op position because I was a girl, employers ogling women co-op students, and the sheer disparity between the number of women and men in my program, to name a few. Around the same time, I had also started taking undergraduate courses in anthropology and knew that a holistic and qualitative perspective would yield better, or at least different, insights.

I set this idea aside to pursue an MA in anthropology studying the meanings and practices around traditional Irish and Newfoundland music, another topic of interest to me. There are numerous reasons that I chose to pursue postgraduate studies in anthropology rather than a career (academic or industry-based) in computer science. These include some of the issues mentioned above in terms of the scope and framing of the discipline and disparity and discrimination relating to gender, as well as having multiple interests that seemed better accommodated by anthropology. The reasons I give change based on the context, but as I discuss in Chapter 7, I inevitably am counted as one of (and become representative of) the many other women who choose to leave computer science and related disciplines – part of the leaky or "shrinking" pipeline (Camp 1997). For my PhD I thus decided to return to questioning this relationship between gender and computer science, but from an anthropological perspective.

2.1.1 Proposed Research

I titled my proposed research topic for my dissertation "Gendering citizens and subjects: The making of computer scientists in Singapore." The decision to study computer science in Singapore emerged out of my literature review and a series of decisions based on feasibility of access, language usage, and the presence of an established and specialized department focused on computer science. I wanted to take an international perspective since the large majority of social science research on computer science, and particularly gender and computer science, centres on so-called Western countries (Abbate 2012; Acker and Oatley 1993; Coleman 2013; Downey 1998; Durndell 1991; Kirkup et al. 2010; Margolis and Fisher 2002; Misa 2010; Robertson et al. 2001; Turkle 1988, 2005). In trying to answer the question of "why so few women?," explanations have centred on the masculine associations with the field, women's lack of interests and abilities, and gendered divisions of labour by sub-discipline (e.g. Hill, Corbett, and St. Rose 2010). However, these projects focusing on "women *in*" computer science and related STEM fields have done little to correct disparities in academic enrollment, graduation, and pursuit of computing professions (Faulkner 2000, 87–88). In some cases, disparities have even increased (Ashcraft and Blithe 2010).

The few historical articles on gender and computer science in Singapore that I found suggested that in the 1980s the number of men and women in computer science in the city-state was approximately equal, although the relative number of women has since declined (Kheng 1989, 1990). I was determined to take a more performative approach to gender than used in many studies on women in computer science. However, the historical numerical equality of men and women and then growing disparity pointed towards changing valuations of gender in Singapore. Additionally, since the 1980s, the Singaporean government has made a concerted effort to promote and use IT, particularly software development, to develop the national economy, which I discuss in Chapter 6. The government has also instituted many programs that promote marriage and procreation among (some of) its citizens. The combination of technological development and government policies on gender, marriage, and sexuality, I thought, would provide easy insight into the relationship between gendering and computing in Singapore, although I ultimately found that eliciting discussion on these topics was not so straightforward.

I had never been to Singapore and so in developing my project I relied heavily on what I read in the literature, which was extensive but often focused on government policy and practice.[11] There were few options for arranging long-term stay in Singapore in terms of visas and immigration; I could stay only 30 days with a possible limited extension as a tourist. Thanks to the help of a professor from Singapore who I met at the American Anthropological Association meetings, however, I was able to complete the paperwork for a student visa and become a "non-graduating research student" at Temasek University, an amorphous category in practice, it turned out, as much as in name. I also emailed the head of the computer science department ahead of time and received the response that participation in my research was at the discretion of professors and students. With my ethics application approved, permission from the computer science

[11] Singapore's particular form of paternalist authoritarian governance (cf. Chong 2005; Deyo 1997; Tan 2003, 2012) has become a kind of theoretical metonym for the city-state (Appadurai 1986, 357).

department to begin research, and temporary accommodation at a tourist hostel until I arranged something more permanent, I flew to Singapore.

My proposed research centred on three broad questions, entailing three overlapping levels of inquiry:

(1) What are the technical and social learning experiences of students, particularly in relation to gender?
(2) In what ways do university and computing curricula (hidden and overt) seek to shape student subjectivities?
(3) How and to what extent are various nationalistic and globalizing projects of the Singapore state implicated in these curricula and in students' experiences?

I discuss in the next section the specific methods that I used to address these questions, which generally encompass the scope of my findings in the field. At the same time, my research certainly took unexpected (although often not surprising) shapes and directions. For example, while I had known from the literature that Singapore worked to attract foreign expertise, known as "foreign talents," I found interesting tensions around the competition that these foreign talents produced in computer science, and Singapore more generally, as well as with the aspirations and desires of many students to leave Singapore to work in such places as Silicon Valley despite the government's ongoing and significant promotion of the local tech industry and "ecosystem." I discuss these issues in Chapters 6 and 8. At the same time, I also discuss below how I came to realize many insights and reflections about my research only after leaving Singapore and spending time going over the various materials I collected.

2.2 Doing Fieldwork

I arrived at 6am in Singapore's award-winning Changi airport after a long and boring twenty-seven hours of flying and transiting. It was still dark out. There is a consistent sunrise and sunset around 7am and 7pm throughout the year rather than the long summer and short winter days I am accustomed to in Canada. My experience at Changi airport was smooth, although it was also instantly noticeably hot and humid. The lineup was short at immigration, I was given no hassle and asked few questions. I was also waved through customs without any scans or checks, even as a group of Indian men were made to put their luggage through x-rays.

My first few days in Singapore were filled with jet-lag, awkward encounters as I figured out how to order food at hawker centres, and hours spent scouring various online sites looking for an apartment. I eventually agreed to an over-priced but conveniently located room in a shared apartment in a Housing Development Board (HDB) flat, one of the many government built and run apartment buildings that house approximately 82% of the island's citizens and permanent residents (Singapore Department of Statistics 2011, xi). In the meantime I also started sending emails to computer science professors asking if I could sit in on their courses in order to conduct participant observation and "begin" my fieldwork.

My research largely took place from August to May of the 2013-2014 academic year. The "fall" semester starts in August in Singapore and consists of thirteen weeks of instruction followed by exams and a short break. The second semester starts in January. My research during this period consisted of participant observation in all levels of undergraduate computer science courses, tutorials, and labs, and at numerous co-curricular events such as hackathons, workshops, speaker series, project showcases, and career fairs organized by the computer science department, the university, and Singapore's tech community. I conducted in-depth interviews with students, professors, university administrators, professionals, and industry recruiters. I also regularly read Singaporean state and independent media, technology news and blogs, and government and university policy and curriculum documents, along with computer science curriculum recommendations provided by the Association for Computing Machinery (ACM). Finally, I conducted a brief return visit to Singapore for the Society for the History of Technology (SHOT) annual meetings, as well as follow-up interviews and further library and archival research from June to July of 2016. I address each of these methods separately and outline the details of my approaches and experiences.

2.2.1 Participant Observation

Most of my initial emails asking professors if I could sit in on their courses went unanswered. As a result, I eventually started knocking on office doors and requesting a meeting. After speaking with professors in person, most were willing to let me participate in their classes. Most had no objections to my presence after I assured them I would not interfere with their class or students' learning, and since most courses I attended were large lectures in theatres, I became just another body in the room. A few professors were concerned about approval for my research in the University hierarchy. I thus also met with the appropriate Dean to discuss in greater depth my plans, which led to a small change in my consent process for classes (students were to be sent an email in addition to my giving a presentation in class) and the department wanted a written copy of professors' consent. This change was mutually agreeable to myself, the professors, the computer science administration, and my ethics board, and I encountered few other concerns or objections throughout the course of my research.

I was able to arrange observation in four undergraduate computer science courses for each of the two semesters during my fieldwork. Altogether, I sat in on three first-year courses, two second-year courses, two third-year courses, and one fourth year course. All of these courses included a lecture component, while several also included tutorials, and labs. I arranged observations separately with tutorial leaders for these tutorials. As a result, I sat in on three separate tutorials for the same first year course, enabling me to compare teaching styles, content, and student-TA interactions. Yet, there were some courses where I did not attend any tutorials. In most cases, professors also gave me access to the online learning system where I could access the slides, handouts, forum discussions, announcements, and other materials for the course.

I had originally planned to focus on first- and fourth-year courses to compare and explore the changes that students experience and perform in becoming a computer scientist between when they start their programs and when they are close to finishing. I ended up observing courses across all levels, partially based on which professors I could connect with to ask for consent, but also so that I could explore in greater detail the progressions of students' experiences in learning and becoming computer scientists. Given that course planners could likely assume few students would be taking both first- and upper-year courses, I also had some difficulty in arranging my schedule to attend courses that did not overlap. The majority of courses I observed were core computer science classes that all students in the discipline were required to take. However, two of the upper-year courses I observed were electives, allowing me to consider the different specialities and knowledges in computer science and the different ways students could become computer scientists. To help maintain professors' anonymity, I do not specify the titles or codes of the courses I observed.

During lectures, I would usually sit near the back of the class and note generalized observations (Fife 2005, 72–77), including: the design and features of the spaces; what students were doing; how students performed and presented themselves through their dress, their postures, their seating choices, and the technologies and tools they were using; the content of the lecture; how the lecture material was presented; the language being used, including pronouns, metaphors, and examples; and how professors presented themselves. Sometimes these notes shifted to becoming more like lecture notes that I would have taken when I was a computer science student and it was difficult not to simply lapse back into that role. Near the end of each term, I conducted focused observations. These centre on one or two particular topics of inquiry and provide additional examples or counterexamples of behaviour, exploring the scope and depth of certain phenomena, and testing hypotheses and assumptions (Fife 2005, 83). Topics for focused observations included the ways technologies were treated as actors; metaphors used to explain algorithms and computer science concepts; qualities and values such as independence or critical thinking emphasized by professors; and gendered performances. At the beginning of every class I would also make a diagram of student seating arrangements, which provided a sense of which students frequently sat together and the ways students organized themselves and socialized in terms of gender, race, or religion.

I stood out as one of the few non-Asian students in the classrooms. Still, my presence was not that unusual, as Temasek University and the computer science department regularly hosted exchange students from Canada, the US, the UK, and many other countries around the world. I even met a couple of students from the University of Waterloo, where I had done my undergraduate studies. My presence was thus rarely marked in any significant way, although there were several exceptions. Professors did occasionally come to speak to me during lecture breaks, and a few professors commented out loud about whether I was taking observations on what they were saying in that moment, although that was often on the day when I did my original introductions. After these initial introductions to professors and students, I often simply blended into the background, becoming just another student sitting in the lecture hall.

In one case a professor asked me to give a brief (approximately 5 minute) presentation on ethnography since it was being discussed in class in relation to user-studies.[12] I also once suggested "gender and computing" as a possible topic for group presentations about various social and technological issues relating to computing, which the professor gladly accepted. I also attended several final-project presentations that were otherwise only attended by the professor and TA who were grading the presentations. Finally, in one of the tutorials I observed, I frequently sat with a student who was

[12] User-studies encompasses a variety of methods from surveys to ethnographic research that seeks to understand how users interact with a particular technology (broadly defined).

struggling with the material and so I often worked with her during the tutorial to help explain concepts or work towards the solution of a given question or problem. I was thus sometimes in an ambiguous and ambivalent position of having and displaying input and expertise more akin to a teaching assistant in ways more representative of my background outside of fieldwork, albeit with none of the same responsibilities or power.

It varied based on context as to whether I would discuss my background in computer science. I would generally introduce myself as an anthropologist studying computer science education. There were, however, various contexts where it was necessary or useful to indicate that I had some expertise in computing, such as gaining access to upper-year computer science courses, and asking students or professors to discuss technical details in greater depth. As such, I did not hide my background, and would occasionally highlight it when it was strategic to do so. At the same time, it often worked to my disadvantage that others knew I had studied computer science, particularly in discussing the role of gender, since many would ask what I thought about why women left computer science instead of offering their own opinions and perspectives. Additionally, there were many facets of computer science that I was not familiar with, either due to developments over the decade since I had been an undergraduate student or related to specializations that I had not pursued. As a result, there were sometimes aspects of computing that others assumed that I understood and I would have to explain why I did not. These disjunctures were informative, but also often awkward and tended to break the flow of conversation.

Where possible, however, I endeavoured to participate in classes as a student. This was generally easy in lectures, as mentioned above, since regardless of whether I was conducting observations or taking lecture notes I appeared to be an especially attentive student. I also completed several individual programming assignments and wrote three midterms, one of which was even graded. Where there were in-class activities, I endeavoured to participate with and on the same basis as other students. For example, I became an item to be sorted in an embodied demonstration of sorting algorithms in one case. One course featured a variety of weekly hands-on group activities, which I joined in. My insights from in-class observations and participation are discussed in Chapter 3 – where I consider the construction and learning of computer science knowledge and practice and how these are tied to the construction of computing "worlds" – as well as in Chapter 5, where I discuss some of the values and behaviours that are cultivated through computer science education.

Several of the class projects, however, were done through group-work. Since I felt that I should not have a (potentially negative) impact on students' grades, I was generally unable to participate in this aspect of students' academic experiences. Through attending tutorials and listening to questions posed in class I gained some sense of what concepts students were struggling with on assignments and how they worked through questions individually and in groups. I could also clearly observe some of the social connections and social groups that were (per)formed in class, with certain students always sitting with one another in class and tutorial or asking one another for help. In one case I participated as an extra group member for a class project at the suggestion of the professor and prompting of students who thought I could provide useful input. I participated in group meetings and discussions and helped proof-read the group's reports, but the work for the project was done by the students. My participation in this group was particularly valuable as it allowed me to see the various ways students coordinated group projects, using numerous different tools for communication and joint-work such as Google Docs, WhatsApp, Facebook Groups, Facebook Messenger, and e-mail alongside face-to-face meetings and discussions. It also allowed me to see negotiations among students about what makes a good project, efforts at interpreting assignment instructions and the professors' intentions, and different goals among students for project and class work.

I had also hoped to shadow and observe students while programming to understand how students relate to/intra-act with technical actors such as code, compilers, and computers, as well as their thought and coding practices and challenges. To some extent, I was able to watch students program during first-year lab exercises, as I usually sat at or near the back of the classroom and could watch either the screen of the student in front of me or beside me. The majority of students' programming work, however, was completed on their personal computers and often in private, such as at home. Even when students worked on assignments in public spaces, such as in common areas, there was often no convenient or non-disruptive way to watch students in the midst of doing programming. I tried a couple of methods to otherwise observe students while programming, including asking students to complete a small programming problem during interviews, or to record their screens while programming at home. In all cases, however, no student seemed enthusiastic about the idea or volunteered to participate and so I ultimately dropped these ideas.

Fortunately, as I discuss in Chapter 5, many professors either posted copies of students' solutions as examples for others to learn from, or asked students to display and explain their solutions in class to discuss and critique. These illustrations provide some insight into students' programming thought processes and coding practices. I could also see how students positioned and performed themselves in relation to computers as technical objects, since students frequently used their computers in public spaces, including in lectures, to program among many other tasks. I also asked students about their computer use and programming practices in interviews. These various methods do not provide the same in-depth and detailed observations about how students intra-act with computing technologies as long-term shadowing likely would have, but nonetheless did provide some insights that I discuss in various parts of this dissertation.

I spent a great deal of time hanging around common areas in the computer science department and around the university, particularly as it was often not convenient to return to my flat between classes or meetings. This practice seemed common among students who did not live on campus, some having up to two hour commutes on transit to and from home. I often saw students taking naps in these spaces, along with working individually or in groups, and socializing with friends and colleagues. I often would use these spaces as a convenient place to read news or catch up on fieldnotes, but I also conducted generalized observations on these spaces, how they were used, by whom, and the various individual and social practices that took place in these common and public areas (see Fife 2005, 77–80).

I also attended numerous events, lectures, and meetings organized by the computer science department, as well as other departments, institutions, and co-curricular groups around the university. These included lectures from visiting computer scientists such as Kent Beck, who I discuss in Chapter 3, hackathons, student project showcases, and lectures and panels on the state of education in Singapore and other issues and topics relating to the city-state. Most weeks I attended meetings of the university's student hacker group (discussed in detail in Chapter 6) that ran demonstrations and hosted talks by students, tech industry professionals, and local and multinational company representatives and recruiters. I also participated in free tours around Singapore for exchange and graduate students offered by the university and different student bodies. These provided additional insight into the culture and history of Singapore that the university and tour operators seek to display and promote. The number of events that were of interest to me taking place at the university could be overwhelming.[13] I often found it difficult to choose what to attend and struggled with not having enough time or energy to attend more.

---

[13] These different activities provided a variety of insights. Attending the student hacker group meetings, for example, allowed me to consider and explore one particular way that students and others promoted and performed being a "good" computer scientist that has in many ways gained hegemonic status at Temasek University and beyond, as I discuss in chapter 6. Attending different events also offered more general insights into what the university could afford to provide. Many events were catered and offered free gifts. For example, over the course of my research I received from these events, or from just walking around the campus, free candy and snacks, two laptop sleeves, several magazines, newspapers, and books, a pencil case, several bookmarks, two t-shirts, two water bottles, and a day planner, which are just a subset of the items available to undergraduate students through their departments and faculties.

My position as a "non-graduating student" provided me with similar access and privileges to other students, including access to most university events, activities, and facilities. As a temporary international student, however, I was also often categorized the same as an exchange student. This was convenient in some cases, where I participated in events providing introductions to Singapore culture and local tours. Yet, as I was focused primarily on my research, I treated my time in Singapore much differently than most exchange students who often focused more on self-organizing for adventurous weekly excursions around South East Asia. At the same time, I was reminded that I was not the same as a full-time undergraduate (or graduate) student – and particularly not a computer science student – every time my student card would not open the various electronically locked doors around the computer science building or let me log on to certain electronic systems that students enrolled in computer science courses had access to by default. In Singapore, everyone must also register for events prior to attending, regardless of whether you need to pay for tickets or not. This process worked to filter or notify people, like myself, who are unwanted or who do not fit within the desired categories. These small frictions and barriers were often frustrating and sometimes flustering. I occasionally accidentally attended an event for which I was not the intended audience, for example. But they also illustrated the different ways students and other persons were categorized, the differential access and privileges accorded to different categories, and the ways that electronic systems worked to enforce categorical boundaries.[14]

[14] Some students suggested to me that my lack of access, particularly in the case of doors, was simply an oversight and that I should be able to request permissions be granted. I never did so, however, because I felt uneasy asking for additional privileges. While I had asked for and been given permission necessary for my research in the computer science department, I did not want to test the limits of my somewhat ambiguous

Although I have no way to substantiate it, I also think my ambiguous status, along with the lack of an official relationship between my home university and Temasek University, contributed to my unsuccessful attempts to stay in on-campus housing which similarly operated through an opaque electronic application process. While on-campus housing would have been more convenient for my research, and likely would have facilitated interaction with students in more casual and informal settings, it was also common for students (particularly Singaporean students) to live at home. After another unsuccessful application to live on-campus during my second semester and despite the amount of time and stress it took over the course of a month to find a new apartment, I moved to a room in a shared condo flat with two foreign professionals that was less conveniently located but that provided amenities like a pool and, more importantly, greater freedom.[15] While moving took time away from the direct focus of my research, it allowed me to explore the variety of housing options available to students, to experience a different neighborhood in Singapore, and I had many discussions with international and local students about the struggles with housing, landlords, and commuting.[16]

---

and tenuous position and relationship to the department and, more significantly, to the larger university hierarchy. In most cases, I also found the barriers I encountered more of an annoyance (along with a source of insight) than a hindrance.

[15] I struggled with my decision to move since my first apartment was well located and my roommate/landlord was a Singaporean of a similar age to myself. We had had some useful and insightful discussions about Singapore. It also required breaking my lease, which came with financial consequences. However, as I discovered is common with renting in Singapore, there were rules such as restricted timing for doing laundry or using the air-conditioner that I struggled with. These rules and restrictions were also quite minor compared to those I heard from some other students, particularly in relation to cooking. I also felt constantly monitored in how I spent my time and in my use of the apartment, which worked to compound my stresses about my fieldwork, research, and integration in Singapore.

[16] I spent a great deal of time over the course of a month looking at different rooms for rent as well as privately-run student accommodation. I was astounded in most cases at the cost of a small room, sometimes shared, in cramped and not always clean or maintained buildings or apartments for students. One of the first ones I visited, for example, was $1,200 SGD for a small dormitory-like room (private) with access to a shared bathroom in an albeit clean student housing facility located near downtown Singapore. This I found

I focused most of my time on participant observation at the university. Yet, I also attended a variety of tech and computing related events around Singapore more generally, including meetups, tech talks, recruitment talks, and workshops. These were organized by several different groups and companies and generally took place either in the Singapore start-up hub "Block 71", at the offices of different tech companies around Singapore, or at Singapore's "hackerspace." These events were public, although occasionally they were limited by the size of the space. They allowed me to explore different spaces associated with the tech industry and computing-related activities in Singapore and to see how computer science education at Temasek University relates to the local tech community and industry. Some of the values and practices promoted throughout the Singaporean (and international) tech scene and that I observed at these meetups are discussed in Chapter 6 and 7. There were, however, many meetups and events that I was unable to attend, including the meetings of the Singapore Computer Society, which runs a variety of professionalization activities and which could have provided useful insights on what it is like to work as a tech professional in Singapore.

As part of my participant observation, when appropriate, I also took photos of spaces and aspects of the physical environment around the department, university, and Singapore more broadly. These photos provide quick and detailed documentation of the

---

to be relatively standard pricing for private student accommodation. Speaking with other international and exchange students who had, like me, not been offered a space in on-campus housing, I found students pursued a variety of housing options from sharing rooms among two or three people in order to afford access to a posh condo flat, renting rooms (shared and private) in HDB flats, or renting a room in these private student facilities. I was fortunate (and privileged) not to experience any overt discrimination while searching for housing, but this was not the case for many others who were rejected outright with housing ads simply stating "no Indian" or "no PRC" (referring to the People's Republic of China).

physical look, design, and layout of spaces, as well as of signs, posters, and other meaningful decorations. I used one of these photos for elicitation in interviews. However, while they help inform my analysis, I only include a few of these photos throughout this dissertation to help maintain the anonymity of the university and department, as I discuss below.

My return visit to Singapore in 2016 enabled further participant observation. For the length of the conference I was for the first time able to live on-campus, providing me with better insight into the facilities and spaces where many students live during their studies. For the remainder of my visit I stayed in a new area which introduced me to a new neighbourhoods in Singapore. I experienced numerous similarities from my earlier time in Singapore, such as taking the same buses, following the same routes, and going to the same eateries, as well as differences such as new buildings and facilities. Previously blank walls in computing common areas had been painted with sayings that promoted familiar values in computer science such as innovation and progress. I again experienced the boundaries and privileges of different categories: I lost my campus library access and could not stay in on-campus accommodation except during the conference. At the same time, as a public institution, I was able to move through campus quite easily with few obvious barriers or hindrances, although I also knew which areas and facilities I could likely access or not. I also attended a public tech meetup like those discussed above to see what had changed in the two years since my main fieldwork. Many of my observations from around the university and around Singapore inform my analyses and discussions throughout this dissertation, even if I often do not address them directly.

2.2.2 Interviews

I started conducting interviews with students during my second month of fieldwork, earlier than originally planned, after a student introduced himself to me after a class and offered to speak with me. I conducted in-depth semi-structured interviews with students, professors (relating both to teaching and administration), graduates, university administrators, professionals, and industry recruiters. Interviews were helpful in several ways, including: learning information that I could not easily gain through observation; exploring various academic and career paths through computer science education and beyond; exploring the values, goals, and perspectives of different persons relating to learning and teaching; and to discuss, confirm, or adjust suppositions and theories that I had developed throughout the course of participant observation. Interviews can thus provide a great deal of information, but they are also performances where interviewees (and interviewers) enact particular and partial selfhoods and identities (Haraway 1988; Khan 2011; Khan and Jerolmack 2013). Shamus Khan and Colin Jerolmack (2013) show the value of combining interviews with observation for exploring the relationships between what participants say and what they do, and why. I discuss similar meaningful incongruences in Chapter 6 where participants say that passion and a "hacker culture" is lacking in Singapore, but in practice the value of passion is hegemonic in the ways students relate to and assess one another.

I conducted interviews with a total of thirty students, including undergraduate, post-graduate, and recently graduated students, over the course of my fieldwork and return visit. Among these interviews, seven are with first-year students, ten with second-year students, two with third-year students, eight with fourth- or fifth- year students (including follow-up interviews conducted during my return visit with students whom I had previously interviewed as first- or second- years), three with recent graduates, and three with PhD students. Eighteen of these students presented as men and twelve as women. Thirteen of these students are Singaporean or Permanent Residents (PRs) and seventeen are foreign students of various nationalities including Chinese, Sri Lankan, Malaysian, Swedish, Indian, and Indonesian. As part of these interviews, I also talked with students in different roles such as Teaching Assistants (TAs) or participating in the student hacker group or other computing related student groups.

These interviews include seven students who are in programs related to computer science including Computer Engineering, Information Systems, and Data Analytics. Students in these programs take several of the core computer science courses, but their programs also entail a variety of other courses that students majoring in computer science generally do not take. I conducted interviews with these students to understand the ways students with different majors perceive computer science as a discipline, why they chose different but related fields, and to explore other potential ways of becoming a computer scientist or computing expert. I thus sought to conduct interviews with students from a variety of backgrounds and with different approaches, perspectives, and goals in relation to computer science. Yet, interviewees were also selected based on access and convenience, including those who volunteered for interviews, those who had been referred to me by others, those who responded to my emails requesting an interview, and those I had come to know through classes, tutorials, or social events.

Interviews lasted approximately one to three hours. As suggested above, some of these interviews were with students I had met repeatedly and come to know well. Yet, a majority were with students I had encountered once or twice in class, or with others who I had been referred to or contacted but who I had not necessarily previously met in person. While there were many students studying computing (i.e. potential participants; there were 1480 students in computing for the 2013-2014 academic year), as I discuss further below, students did not form a single community. I thus did not have access to every single student. Even among those who I shared classes with, many students often did not attend lectures, the large lecture theatres were not conducive to meeting others, and my own shyness often made it difficult for me to simply introduce myself to students that I did not know from other social circumstances. Additionally, several weeks into the term, when students' assignment load grew and then students had midterms and finals to worry about, there were fewer students to be seen in general and many were focused on studying or finishing their work. Not only was I hesitant to take away from their time "mugging" – local slang for studying hard – but several students said they felt too busy to take time to meet for an interview.

My interviews were semi-structured, following an evolving list of questions but with flexibility where I would add questions that I thought of or that emerged over the course of the interview, or leave out others that had been more or less covered already or that seemed irrelevant or inappropriate depending on the context.[17] For example, I had

[17] See (Fife 2005, 95-101) for a thorough discussion of semi-structured interviewing practices that is reflective of how I used them with participants.

specific questions relating to being a foreign student in Singapore or working as a TA that I would ask if they were relevant to the student I was speaking with. I also had begun to include questions about how a person identifies in terms of gender that I left out when one of my interviews was taking place at a shared table in a common area. Singapore is strongly heteronormative, as I discuss in Chapter 4, and I did not want to accidently out or put an interviewee on the spot in front of others. I struggled asking questions about gender identification, feeling extremely awkward doing so since most interviewees found these question highly unusual, although this generally only earned me a confused look and a hesitant response of “male” or “female” or a question in response about what I meant. In some cases, however, I received surprising and valuable explanations about the different ways students thought about their gender and why. These encouraged me to continue asking these questions, despite (and because of) my feelings of awkwardness and some participants’ confusion. An interview schedule for students can be found in Appendix A.

In student interviews I also used photo elicitation with two images, displayed in Figure 2-1 and Figure 2-2. Douglas Harper has suggested that photo elicitation can offer two significant insights: first, in presenting unexpected framings of an event or activity it can lead interviewees to reflect on their experiences of reality; and second it can offer bridges of understanding across different social worlds (Harper 2002, 20–22). Figure 2-1 is a comic that was shown in a class that I was observing, which I used to elicit students’ perspectives on mathematics and computer science, as well as on gender issues in relation to these subjects. Figure 2-2 (edited for anonymity) was taken prior to a career fair in the computing building and I used it to discuss students' role models, their perspectives on the career fair, and on emotions and computing (e.g. do they feel like super heroes when they program?). Students at first seemed unsure about the process, and many started by saying they were not sure what to respond in relation to these photographs, but after I asked a related question they would generally have something to say. These responses often did not provide the depth of discussion that Harper suggests they could elicit, but they allowed me to touch on topics and issues that otherwise would be difficult or confusing to address.

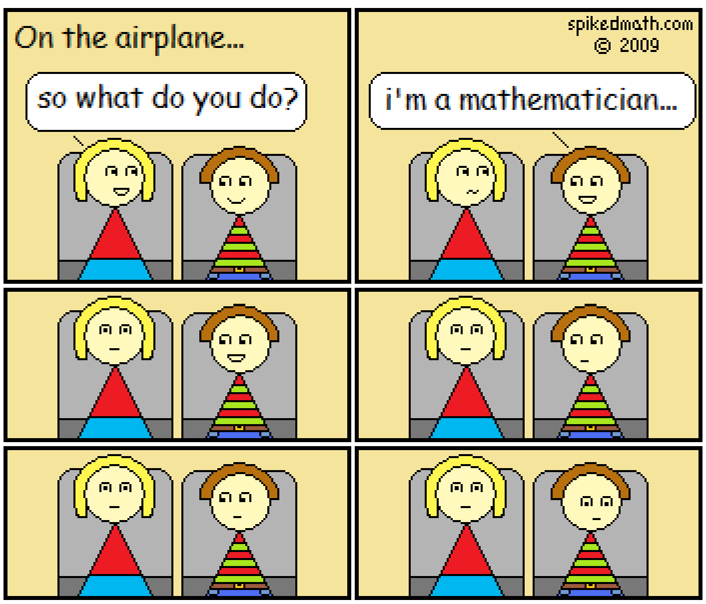

Figure 2-1: Image for photo elicitation

Figure 2-2: Second photo elicitation image (edited for anonymity)

I conducted interviews in locations and times chosen by participants. Generally interviews with students took place in common areas around the university during the day, either in between students' classes and meetings or on one of their days off. Occasionally we would find an empty classroom for a quiet and private location. Most interviews with professors took place in their offices, again during a time that was convenient for them. I also conducted two interviews via Skype, one with a recent graduate who was working outside of Singapore and another with an industry recruiter who I had met in Singapore but who had left to continue her recruiting work elsewhere.

I interviewed a total of thirteen professors. Six interviews focused on professors' administrative roles, such as curriculum planning or entrepreneurship programs.[18] However, I would also often ask about these professors' educational backgrounds and career paths to gain additional perspectives on learning computer science in Singapore and abroad, and to gain some personal context for my questions relating to their administrative roles. The other seven interviews were with professors of the courses I attended as a participant observer. In these interviews I focused on professors' education and career trajectories, as well as their teaching goals, values, and methods. I often also asked questions about things I had observed in class or decisions about course content and design. Many parts of these interviews, such as discussing professors' educational backgrounds and paths, followed a similar structure to that of students allowing for further comparison across time and persons. An interview schedule for professors can also be found in Appendix A.

Through a family connection, I was also able to arrange five interviews with professionals at a multi-national tech company with offices in Singapore. These were an exhausting series of successive one-hour interviews that took place over the course of a single day at the company offices in one of their meeting rooms. They had been arranged and scheduled on my behalf, although I worked to emphasize to interviewees that they were welcome to decline and that I would protect their confidentiality. Interviewees were somewhat surprised by my questions, focusing significantly on the trajectory of their

[18] I discuss both curricula and government policies relating to entrepreneurship in Chapters 5 and 6. Courses on entrepreneurship were offered to a select group of students who showed interest in and potential for developing a startup company, but entrepreneurialism was promoted more broadly as part of a hidden or implicit curriculum in computer science.

careers and their perspectives on working in the tech industry in Singapore, since my research had been presented to them as being primarily about women in computing, but none raised any issues or objections. While not the focus of my research, it provided me with the opportunity to ask about and explore how efforts at education and professionalization continue after students graduate and different experiences of working in the tech industry in Singapore.

2.2.3 Media and Library/Archive Research

When I was not busy with participant observation or interviews, I spent a great deal of time reading Singaporean news, as well as local and international tech news. I frequently did this while sitting in common areas around the university, conveniently combining such research and participant observation. I continue to read relevant news, although less frequently than I did during my fieldwork. Reading news provides me with insight into social, cultural, and political issues and debates in Singapore and their public presentation, as well as government policies and practices, and local and international technological and business changes and developments. This information is thus significant for providing a general context for my research, as well as for seeing some of the changes and developments that are occurring in Singapore since I left in 2014. At the same time, news media in various forms provide a partial picture that is itself situated socially and politically in particular ways.

In relation to local news, I read mainstream state-owned Singaporean news media including the *Straits Times*, *Asia One*, *The New Paper*, and *Channel News Asia*. The first three of these are owned by Singapore Press Holdings, a publicly traded company with its senior executives appointed by the Singaporean government. *Channel News Asia* is operated by Mediacorp, a government-owned corporation. While the government does not directly control these media outlets, there was a strong sense that what is published centres on and is shaped by government interests. While the same could be said of the Canadian Broadcasting Company in Canada, censorship laws are stricter and government influence over media is stronger in Singapore where removing controversial or offensive content from media in general is common, and journalists report various forms of self-censorship, as well as pressure from editors to limit coverage of opposing political parties and controversial topics (see, for example, Hicks 2013; Jaswal 2017; Leow 2016; WikiLeaks 2009). I also draw on archives of these news sources provided online by Singapore's National Library Board (http://eresources.nlb.gov.sg/newspapers/), which offer insights into historical policies and practices relating to computing in Singapore.

I also read Singapore alternative news media and commentary such as the *Online Citizen*, *The Breakfast Network* (shut down December 2013), *The Middle Ground* (established in June 2015 by some of the same people who ran the Breakfast Network, shutting down at the end of 2017), and *The Real Singapore* (shut down May 2015).[19] As suggested by these changing platforms, independent media in Singapore is small and struggles with many legal and administrative restrictions. *The Breakfast Network*, for example, closed when the Media Development Authority (a government board that

[19] *The Real Singapore* (TRS) did not have the same favorable reputation as the other independent media sites mentioned, and has been critiqued for publishing unverified or false information and plagiarised articles. Yet, I read articles published on the site as it provided a different, even if inaccurate, perspective. That is, which inaccuracies, falsehoods, or exaggerations were published and how they are presented are also ethnographically interesting.

regulates media in Singapore) asked *The Breakfast Network* to register with the board and prove that they did not run with any foreign funding or foreign involvement. They decided to close after finding the forms and paperwork overly onerous for a largely volunteer run, relatively new, and small news and opinion outlet (Tan 2013). These alternative outlets sometimes provide critical perspectives on and detailed analyses of government policies and practices, and also often cover topics less often discussed in mainstream news media, such as issues relating to foreign and domestic workers and LGBTQ communities in Singapore.

While I did not originally plan to read tech news, after asking students about their media consumption habits, I learned that many of them followed tech news more than local news (mainstream or independent). One of the more frequently cited sites for following tech news was *Hacker News* (https://news.ycombinator.com/), which is a social news aggregator and link-sharing platform focusing primarily on computing and entrepreneurship, although members can share any link they find interesting. The site was established by the American startup incubator Y-Combinator and so provides international (although largely American) tech news. For local and Asian tech news I came to read *Tech In Asia* (https://www.techinasia.com/), an independently run "media, events, and jobs platform" with headquarters in Singapore that covers tech and entrepreneurship news primarily across Asia (TechInAsia 2016). These sites provided me with information about and context for understanding developments in the tech industry, future careers and goals that students could pursue, and insight into the relationship between computing education and industry in Singapore and beyond.

In addition to following news media, I collected and read a variety of Singaporean policy documents, particularly national policy and planning reports relating to computers and information technology. These are discussed in detail in Chapters 5 and 6 and provide insights into how the Singaporean government perceives and plans information technology and its role in Singapore society. I also read additional policy documents that were either suggested to me or that I found to be of relevance. In particular, a few participants had mentioned the "Population White Paper" as a significant document. This policy report is also discussed in Chapter 5 where I consider how technology policy in Singapore has sought to cultivate Singaporean citizens and workers to be entrepreneurial, creative, and risk-taking, while promoting and hiring "foreign talents" – workers and experts from abroad – as representative of these qualities.

Finally, I read curriculum documents relating to computer science at Temasek University, including university calendars from 1975 to the present, to compare the course descriptions and changing requirements for a degree in computer science from Temasek University. This historical exploration was not originally a significant part of my research plan, but as I considered the similarities among computer science in Singapore, in Canada where I had done my undergraduate studies, and in the US, which I read about in the literature, I became increasingly intrigued about how this sameness was built and constituted. I thus also read the Curricula Recommendations that have been published by the Association for Computing Machinery (ACM) joint with the Institute for Electrical and Electronic Engineers (IEEE) approximately every ten years since 1968. The ACM and IEEE are international, but originally and primarily American, associations for professional computing and engineering. The curriculum documents are provided as guidelines or suggestions for computing departments around the world on what to include in their curriculum. I discuss these multiple documents in Chapters 4 and 5 where I explore this question of the historical construction and movement of computer science curriculum, and of computer science as a discipline.

2.3 Methodological Reflections

Upon returning to St. John's from Singapore I began the long and laborious process of transcribing interviews and coding field-notes. I used the MaxQDA qualitative data analysis software package to code various forms of data I collected – interview transcripts, fieldnotes, photographs, social media posts, among others – with emergent themes that I could then categorize and compare. After leaving Singapore and my immersion in the computer science discipline, I also began to reconstruct my sense of self in order to again think like an anthropologist rather than some kind of (generally confused) computer scientist/fieldworker hybrid. I only came to fully realize and articulate this struggle, along with the disjuncture between my idealized visions of fieldwork and my actual experiences, upon "leaving" the field.

2.3.1 Fieldwork and "Community"

It was March, nearing the end of the second semester of the academic year. After wandering around the computer science department for a short time I sat in one of the large but quiet common areas. Around me there were a scattering of people in their early twenties, sitting in disparate places at the large common tables and staring intently at the laptops or printed notes in front of them. I was nearing the end of my fieldwork, but still felt I need more interviews; I had said I would do 40-50 interviews with students in my proposal but had not completed near that number. I opened my email and sent a message to a student I had met earlier at a workshop, asking if he was willing to be interviewed for my research.

There was little talking around me. The large laser printer on one side of the room suddenly started up, its whirring and chugging occupying the sound in the room. I didn't recognize any of the students around me, even though I had been hanging out in the computing building for almost eight months now. I heard back from the student a few days later, he wished me luck with my research but said it's "crunch time" and he had many projects due and so he couldn't meet. His response was not unexpected. The common areas could be bustling early in the semester: students were social and events abounded. But like the silent common area, once midterms and then finals came students evaporated or fell into silent "mugging" – studying hard for their upcoming exams in their quest to beat out other students for their place on the bell-curve.

This situation is of course informative for my research, but it also exemplifies ongoing problems with the way anthropological fieldwork is conceptualized and done. The idea that there are bounded and localized communities where anthropologists can study "other" cultural practices has been heavily critiqued (Clifford and Marcus 1986; Gupta and Ferguson 1992, 1997; Wolf 1997). Formulations such as "multi-sited" ethnography (Marcus 1995) and fieldwork as an "assemblage" of sites (Reddy 2009) have worked to complicate the notion of the field as bounded in time and space and help researchers look towards interconnections among places and groups and the role of the anthropologist in defining and making “the field.” The concepts of multi-locality and ontology in anthropology have also pointed to the multiple ways that places can be understood and experienced (Carrithers et al. 2010; Kohn 2013; Rodman 1992; Viveiros de Castro, Pedersen, and Holbraad 2014).

At the same time, anthropological research at the graduate level continues to centre largely on an extended but bounded period of time for fieldwork, often in a particular place.[20] Additionally, while anthropologists are generally more open and reflexive about the process of doing fieldwork since the publication of “Writing Culture” (Clifford and Marcus 1986), anthropological texts continue to present a generally cohesive picture of fieldwork teeming with the ethnographer’s insights. As Robert Desjarlais argues, the notion of “experience” remains central to much anthropological research and, while it is rarely defined, experience is treated as a coherent and progressive process of personal growth and development (Desjarlais 1996, 74–75). In this way, while I knew the many critiques of anthropological methodology – in theory – in terms of both research and writing, I nevertheless struggled with the feeling that my integration into the computer science “community” and my knowledge about life as a computer science student should be developing in a more or less cumulative and

[20] The internet and social media have blurred boundaries between fieldwork and non-fieldwork time. Yet, time-zone differences and differential access and use of internet applications and platforms across spaces work to reinforce these boundaries. Additionally, ideal degree completion times and institutional funding practices, particularly in Canada, place constraints on the time students can and should spend “in” and connected to the field.

progressive fashion. Computer science students, however, did not form a single cohesive community and my fieldwork "progressed" in fits and starts.[21]

The computer science program and department can work as a community for some, and various institutional groups and clubs run by both students and professors sometimes refer to computing students as a "family" or "community." Yet, for myself and many others, such "groupness" did not define our experiences (Brubaker 2004). Many students knew about me as a result of my presentations and emails about my research in the courses where I was conducting observations, sometimes to over two-hundred students in a class. Nevertheless, throughout the length of my fieldwork I was continually reintroducing myself and my research. In only a few circumstances, mentioned below, I did feel sense of belonging or integration among students or professors within the computer science department.

There were instead multiple "communities" or groups that students formed and re-formed. When I asked students in interviews where and how they met friends, many suggested that this occurred in their first year and often before classes even started, at various orientation camps. Additionally, students formed groups around class projects, nationality, multiple shared classes, and participation in institutional groups and committees such as the student hacker group, or faculty-based sporting competitions, to name a few. In one of my classes during my second semester I frequently sat with a group made up primarily of women students after being invited by one particularly social

[21] An increasing number of texts offer frank accounts of anthropologists' struggles in the field (e.g. Gardner and Hoffman 2006; Pollard 2009). These provided some comfort and guidance throughout my fieldwork.

and out-going student who I had met several months prior. My participation in this particular "group" was then somewhat fortuitous although potentially gender-based. Other areas where I felt part of a "groupness" include the group project discussed above; some gatherings of exchange students; and my participation in an archery class for several weeks. My return trip to Singapore also highlighted to me the connections with individual students that I had made through my research when several students I contacted were enthusiastic to meet again.

The bulk of my time, however, was spent outside or beside various "groups." My position as not-quite-a-student who was not regularly completing assignments, and particularly not completing them for grades, often placed me outside the social connections made among students through their shared interest in and struggle with course-work. I did not have the same motivation or stake in doing assignments and finding the answers, and I found it difficult to manufacture the interest or urgency that I remember experiencing when I was an undergraduate student myself. I therefore also did not seek out help from other students in completing assignments, as I might have done as an undergraduate student.

This also placed me outside the flow of time of the academic semester. My research was ongoing despite midterms, assignments, and exams, even though these formed part of the subject of my research. Students, however, generally had different experiences of time. Once midterms started and then moving into final exams, time seemed to become compacted and in short supply for students: full of project meetings, study sessions, and never-ending last-minute assignment work. This situation provided some insights: in writing down all the assignments, labs, tutorials, and other work that students had in a week, for example, I was astounded by the number of things students needed to work on (or even simply keep track of). Yet, it also shaped the "progression" of my research. For the second semester, for example, I tried to conduct multiple interviews in the first few weeks of school before students were loaded down with work, and I reserved library work for exam time. These shifts in scheduling were helpful, but I nonetheless continued to feel throughout my fieldwork like any momentum I gathered was lost a week later.

In this way, my fieldwork fits with the idea of an "assemblage" of places, times, and encounters (Marcus 2006, 1995; Ong and Collier 2005; Reddy 2009). As with many facets of my research, I have only come to this realization after much post-fieldwork reflection and writing; it is thus a retrospective coherence imposed on or pulled from my research. I used a similar framework to interpret my Master's fieldwork, which entailed a collection of different "sessions" – regular but discrete musical gatherings and events (Breslin 2011). "The field" was then defined by my performance of the role of "anthropologist." Perhaps because my doctoral fieldwork was more localized, taking place primarily in a single building and certainly largely encompassed by the university campus and temporally bounded by the schedule of an academic year, however, I was unfortunately less attentive to my own role in "making" my "field" at the time.

Nevertheless, while much of my fieldwork may have been conducted in a bounded place and time, my "field," as such, can be seen as a collection of disparate pieces. As Deepa S. Reddy considers the "field":

> As an almost random assemblage of sites that come into coherence through the process of fieldwork itself: the field as deterritorialized and reterritorialized, as it were, by the questions brought to bear on it in the course of research. This process necessarily entails much movement, as much between physical locations closer or farther apart as between ideological positionings or frames of reference (as I call them). Tracking this movement, understanding the relationships between sites, one's own positioning within each, and the demands placed on the ethnographer coming-into-being – these I believe are the means by which the field is made, quite alongside the objects of study that it yields then to ethnographic attention (Reddy 2009, 90).

Through these pieces, I made connections with certain persons, groups, and places, for varying lengths of time and varying reasons. Many of these reasons (for me) related to my fieldwork, but are also inevitably caught up with other purposes and practices such as forming friendships, sharing interests or goals, participating in organized activities, and seeking or giving support or aid. In many ways, this process also led to a shift in my sense of self. The process of writing this dissertation has imposed an overall conceptual and narrative coherence to my fieldwork that I did not experience at the time.

2.3.2 Fieldwork, Selfhood, and Politics

Dorinne Kondo's recollection of her shock and mis-self-recognition upon spotting her own reflection in a shop display case near the end of her fieldwork is well-known (Kondo 1990, 16–17). Kondo explains how she experienced a collapse of identity and fragmentation of self as her sense of self was being rewritten in order to "make" her Japanese (Kondo 1990, 17). While I never experienced the same kind of shock or self-othering, increasingly throughout my fieldwork I had this nagging feeling that I could no longer really think like an anthropologist and I had forgotten (or had lost) the purpose and meaning of my research. However, I only came to (partially) understand and articulate this process after returning to Canada and spending long months re-reading my fieldnotes, and re-reading and reconnecting with anthropological theory and literature. I discuss these feelings further in Chapter 3 in relation to my background in computer science as I came to easily fall into patterns of thought relating to the construction of computing words and the process of rendering technical. Here, I consider the personal and political implications of this slow loss of self in relation to researching and performing gender.

As I discuss above, beginning my research I was interested in exploring how gender was constructed and experienced by students studying computer science. I wanted to follow a queer and intersectional approach to gender to understand the ways gender is made and remade through computer science knowledge practices, education, and through students' everyday performances, as well as how gender intersects with race and citizenship. However, what became quickly apparent to me was overwhelmingly pervasive assumptions of binary genders and heteronormative relations in computer science and in Singapore. These assumptions are precisely what I was interested in exploring for my research. However, investigating them presented challenges to discussing gender with students, professors, and other research participants, and to maintaining my original approach to gender while conducting research.

I discuss the construction of gender in computer science and in Singapore in Chapters 4, 7, and 8. Briefly, however, gender binaries and heteronormative relationships were taken as a given and became integral to students' and professors' understandings and conceptualizations of computer science knowledge. Teaching examples that relied on gender binaries were casually and repeatedly used by different professors for multiple and different concepts. Government rules, regulations, policies, and discourses in Singapore similarly rely on assumptions of heteronormative gender binaries. Until January 2016, for example, there was government funding for student groups and activities, such as orientation camps, that "maximise opportunities for undergraduates to meet and interact with the opposite gender." As I mentioned above, in planning my research I thought the variety of policies that seek to govern gender in Singapore would make the topic easy to discuss. In practice, however, I found the opposite to be the case.

In introducing my topic and interviewing students and professors I worked to keep my language open. I would ask whether they thought gender was relevant to computer science and in what ways. My questions, however, were widely interpreted as asking about the role of and difference between men and women in computer science. These answers provide insights in and of themselves about how participants see and understand gender in terms of differences between men and women. In particular, my topic was often read as being about women in computer science, with gender generally taken as a synonym for "women." The ongoing issues of disparate numbers of women in the discipline were known to many participants, even if the disparity was not as marked in Singapore as in the US or Canada. The issue therefore did provide a narrow bridge and means of interpreting my research to participants, as I thought it might. Yet, the centrality of this issue highlighted a tension in terms of perspectives and politics between myself and my research participants, and faithfully representing my research intentions and questions in relation to gender presented both practical and ethical challenges.

While students and professors were generally willing and open in speaking with me, few showed much interest in the details of my research. Given an opportunity, I would also clarify that I was looking at both men and women (sometimes saying other genders as well) and was taking a broad approach to gender and its role in computer science, yet that was generally the end of the conversation. This was partially a matter of time; both students and professors seemed constantly busy and overloaded with work and activities. In many cases, to extend the discussion would have also meant elaborating in detail on gender theory. There were few circumstances where this seemed appropriate since it would have prioritized my perspective over that of my participants and I often felt it to be a very awkward topic to broach and discuss.[22] Highlighting the performativity of gender (even within a heteronormative context) also meant implicitly contesting the discourse of the Singaporean government and critiquing practices of professors in teaching and doing computer science, professors who had been generous enough to let me observe their classes and their teaching. As such, to the extent that the issue of the disparate numbers of women in computing stood in for gender analysis, it also stopped discussion in a variety of ways.

The translation of research between informants and participants is a common occurrence in fieldwork. In doing research, anthropologists and their work are often interpreted in different ways based on the social and cultural contexts, as well as based on

[22] Professors and PhD students tended to ask more detailed questions about my research. Related to my research interests, but not directly part of my research, I also developed a collaboration with Dr. Bimlesh Wadhwa in the Department of Computer Science at the National University of Singapore, considering how gender and feminist theory are applied in Human-Computer Interaction research and could be incorporated into computer science curricula (Breslin and Wadhwa 2014a, 2014b, 2015, 2017).

personal relationships and individual and community goals (Kondo 1990; Reddy 2009). In many ways, this type of translation is a necessary part of doing fieldwork. Yet, translating my research to be about women in computer science also often entailed a translation into a problem-based experimental framework. As I discuss further in Chapter 3, I was continuously asked what I was trying to find out, to answer. What was my hypothesis? In this context, my response that I was interested in how gender is involved in computer science felt unsatisfactory, unsubstantial. There was no problem to solve, *except* when my research was frequently reinterpreted as being about the lack of women in computer science. As my research continued, I also began to think about how I could go about proving various facets of my research, for example that women more than men pursued careers in fields other than computer science following their degrees.[23] That is, I was beginning to think about research more in numerical and statistical terms, rather than in terms of students' experiences, values, and behaviours.

I was thus coming to reinterpret my research and myself, to "fit" in computer science. This often meant implicitly accepting norms about gender, and about how to do research on gender in computer science. In planning my research, while I thought the topic of "women in computer science" could provide a connection and shared concern with participants, I was determined not to be "drawn back into the heteronormative black-hole," as I wrote in my proposal, of focusing on statistics and the number of women in computer science. Yet, the hegemony of gender binaries and problem-based approaches in computer science and in Singapore made this a difficult endeavour, one that often felt

[23] Notice how such a research question also relies on an assumption of gender binaries.

impossible while actually doing my fieldwork and speaking with people every day who would translate and reconfigure my research, my perspective on gender, and my perspective on knowledge production. I attended a few talks by local scholars doing work in sociology and science and technology studies that notified me of the stark differences in approaches and questions, and provide me with a glimmer of a reminder of what it was like to think like an anthropologist. In general, however, towards the end of my research I felt like I was working in a haze, knowing that some form of anthropological analytical and theoretical clarity must exist, but that it was just out of reach.

Issues of "going native," the positionality and power of researchers in relation to research participants, and the significance of cultural relativism versus political activism are all topics of lengthy debate and discussion in anthropology (Abu-Lughod 1990; Clifford and Marcus 1986; Narayan 1993; Rosaldo 1993). In researching education and subject-making, the process by which one becomes a "native" through deliberate education, effort, and training – and what that means - was central to what I sought to explore. Yet, in the case of my research, these challenges also occur within a context of "studying up," in relation to the university as an institution, the discursive power of computer science as a discipline, and the governance of the Singaporean state. Laura Nader (1972) argued that studying powerful institutions and actors would provide valuable and necessary insights; by studying up, anthropologists could gain insight into the meanings and practices of those operating within and through often opaque institutional and cultural systems, providing a means to critique operations of power. Yet, in doing my fieldwork, I was also caught up in those operations of power. I was coming to perform – in how I spoke, thought about, and presented my research – according to computer science norms. My summoning to embody the norms of computer science was made all the more complicated (or maybe more simple) due to my past studies in computer science (Foucault 1982; Mahmood 2005).

Rebuilding my sense of self and theoretical direction post-fieldwork – a process that has only been possible through distance and time – has then been a significant process for working to recognize the ways that I was summoned to think and behave in particular ways (Kondo 1990; Mahmood 2005). This process has also been essential in allowing me to explore and interpret the many subtle ways that students and professors interpreted their gender identities while learning and doing computer science, and the struggles they face in "becoming" computer scientists. As I discuss in Chapter 8, while my participants may not have studied gender theory, some students nevertheless expressed perspectives on the fluidity of their gender identities based on circumstance, or the ways that gendered norms and expectations – and personal interests and desires – conflicted with the demands of studies and a career in computer science. These students were interpreting and performing gender in ways that subtly worked to reconfigure hegemonic norms.

While my theoretical approach to gender, as well as knowledge production, differs from the norms of computer science and the Singaporean state, in using these distinct "categories of analysis" that are also "categories of practice," I am able to acknowledge and explore the constitution of those norms, and resistances and challenges to them, by students and others (Cooper and Brubaker 2005). The challenges I faced in following a marginalized theoretical approach – as a person who mostly fits within hegemonic gender norms – are negligible in comparison to those who inhabit identities on the margins of computer science and of the Singaporean state. The power of anthropologists to "leave" the field has also allowed me to maintain or rebuild my approach to gender, or my categories of analysis, after fieldwork, in ways that I hope have allowed me explore students' hopes, fears, and struggles, that would otherwise be submerged by the hegemonic constructions of gender. At the same time, understanding the ways my selfhood is entangled with computer science and Singaporean state discourse, although to a lesser degree in the latter case, is a process that I have continued to grapple with – and work to make explicit at different points – throughout writing this dissertation.

2.4 Writing Decisions

During fieldwork it often felt like the triumphs and insights portrayed in many other ethnographies never came. Those moments where the ethnographer suddenly has an epiphany never materialized. Yet, I have retrospectively built clarity, structure, and insight into my research through the writing process. I have also made deliberate decisions in my writing in relation to the anonymity and confidentiality of participants.

I have chosen to maintain the anonymity of the university where I based my fieldwork. This decision was not a requirement by either my ethics board approval or my arrangement with the computer science departmental administration. Anyone who is familiar with the higher education landscape in Singapore will also likely be able to easily determine at which university I did my research. Using a pseudonym for the university throughout this dissertation is instead meant as a symbolic gesture to emphasize that my arguments, reflections, and critiques focus on systems and structures – relating to the discipline of computer science or universities as global institutions, for example – rather than the specific people, department, or university that was the site of my research.

Treating anthropological fieldwork as based in particular places, where places are treated as unproblematized containers for people and events, has been heavily critiqued (see, for example, Gupta and Ferguson 1997; Nespor 2000; Rosaldo 1993; Wolf 1997). I have discussed above my own struggles with conceptualizing my field-site throughout my research. Jan Nespor (2000) has also critiqued anonymization practices for the way they decontextualize people and places, turning them into theoretical and analytical entities outside of place and time. He suggests that:

> Giving people or places pseudonyms and strategically deleting identifying information turns them into usable examples or illustrations of generalizing theoretical categories… in which form they can stand in for social classes, ethnic groups, genders, institutions, or other theoretical constructs (Nespor 2000, 550).

In my case, anonymizing the university does indeed hide some of its particular history and local social, political, and economic context. It also focuses my analysis more on theoretical processes than the details of the particular department and school. Yet, my purpose is instead to highlight the connections that work to constitute how computer science is conceptualized and taught both in Singapore and through its connections to other places, particularly the US.

I also use pseudonyms in place of participants' names in order to protect their anonymity and confidentiality, except where statements are already found on the public record or the individual explicitly gave me permission to use their real names. While use of pseudonyms is not a guarantee of anonymity, which was mentioned to me by some participants and is another critique by Nespor (2000, 547–49) of the practice, it nonetheless helps avoid definitive identification of participants. I often provide the year of study, gender, and nationality of students, as these are often important details for interpreting their discussions. At the same time, there are various ways that participants could be identified even with a pseudonym, as small numbers of individuals organize certain events, run certain groups, or teach certain courses. As a result, in most cases I will only provide limited background information to participants' words, or alter minor details about participants where I judge it will not impact the analysis but will help mask the person's identity. In the case of professors, I often provide very limited contextual information because professors are much more easily identifiable. In some instances I will use also different pseudonyms for quotes by the same person, again aiming to protect the anonymity and confidentiality of participants.[24] As a result, I do not provide dates when interviews took place, indicating only the year of the interview and only for extended quotes by students.[25] In this way, throughout this dissertation I hope to convey the detailed and nuanced construction and performance of subjectivities by students and others, along with my own, while also working to maintain the anonymity of those who shared their stories about their lives and selves with me.

[24] I use this practice sparingly and only when I felt it was necessary to protect the anonymity of participants.
[25] Because the break between Semester 1 and Semester 2 is the end of December/beginning of January, the year of the interview also indicates which academic semester during my research the interview took place.

Chapter 3 : Initiating Programming

I was sitting in an air-conditioned classroom in a new part of campus on a Saturday morning, a couple of months into my fieldwork. I was attending a student-run workshop. The student hacker group was teaching about Git, a popular version control and distributed programming system. Put simply, it is a much more elaborate and powerful version of the "track changes" found in Microsoft Word that can be used for multiple files and by multiple people simultaneously. The workshop was hands-on: we all had our laptops, had been given instructions to download Git, and were walked through the various commands for initializing our code repository and checking files in and out. The organizers commented at the beginning that they were aiming to teach us a particular skill, namely how to use Git, rather than how to become computer scientists more broadly. That is, they were not going to discuss how to write a good program, different algorithms or data structures, or how to analyze efficiency, all core topics to the discipline of computer science. Rather, we were simply learning a tool which might help us with our academic and possible future industry work.

In order to explain what was being done by certain commands, the organizers used various metaphors. One of these was the idea of a multiverse, where there are multiple parallel universes with minor differences from one another.[26] This was used to explain the process of "branching," whereby a body of code is "branched" usually for the

[26] The idea of multiverses is tied to quantum theory, originating from the work of physicist Hugh Everett III. The idea was enthusiastically adopted in the sci-fi genre, illustrated, for example, by the "mirror universe" throughout the Star Trek franchise. The mirror universe features all the same people, places, and things as primary universe in the franchise, but everyone in the mirror universe is the opposite (and usually evil) versions of themselves from the primary universe.

purpose of making significant modifications without affecting the original body of code. As a result, the two branches (original and new) can develop along different paths simultaneously. I noted at the time to think about how metaphors relate to programming and computer science knowledge. But, only after sifting through notes from class and tutorial observations back in Canada did I realize how the idea of having separate realities as suggested by a multiverse succinctly captures the type of cultural and epistemological production that was happening throughout my fieldwork, and as part of computer science education and practice more broadly.

This chapter explores the content and practices of knowledge learning and production that students undertake in studying computer science. In particular, I consider how beginning in their first year, students and professors collectively build symbolic and material worlds as they learn (for many) their first programming languages. These worlds are complex, rule-based, and often mutually supportive. I also consider how these worlds become naturalized – their existence and logic becomes matters of fact and simultaneously represented as based in the natural evolution of human thought and practice. My goal here is not to critique the representations of reality done through code, programs, or modelling languages in relation to their fit or accuracy. Rather, I seek to explore and understand how these representations are part of computer science practices and perspectives and to explore the "diffractions," as "the processing of small but consequential differences," entailed in the intra-actions among students, programming languages, and technologies (Haraway 1992, 318). This process operates as a means of initiating students into human and nonhuman networks or phenomena – and thereby systems of thought and practice – that are part of the computer science discipline.

3.1 Programming Worlds

Many first-year students in Singapore have no prior experience in programming. Their first class is then sometimes a shock. One first-year student commented "So, it's like, first few weeks we learned a lot of things. Like, in the first week we learned all the stuff about recursion, iteration, and stuff like that… so basically last semester I just feel like I'm being pushed off the edge of the cliff" (Susan 2014). Students are introduced to their first programming language, as well as a variety of technologies and programs needed in order to write, compile, test, and submit their programs. Like the multi-verse metaphor discussed above, I consider how these programs and languages can be construed as "worlds" that operate according to particular but shared rules and logics that are part of but constructed differently from the "real world," and that require students to adopt particular practices, modes of thought, and forms of representation.

3.1.1 Learning to Program

For most computer science students at Temasek University, their first two weeks entail learning many new things. They learn how to connect to and log into a remote UNIX server, how to navigate the file system using UNIX commands, how to open and use Vim (a UNIX-based text editor that students are instructed to use to write their programs; Vim relies on text-based commands rather than mouse-input), and how to compile their code into executable programs and then run them.[27] They also learn about different data types defined by programming languages. Most students start by learning the programming language C and its associated types such as *int*, *short*, *long*, *float*, *double*, *char*. Students are introduced to the idea that programming languages have strict rules that must be followed. They have to learn the basic format of a program in C and write their first program by taking user input, making a mathematical calculation, and outputting an answer based on that input. Early program examples include converting Fahrenheit to Celsius or Miles to Kilometers, as seen, for example, in Figure 3-1.

```
/* Converts distance in miles to kilometres.*/

#include <stdio.h>   /* printf, scanf definitions */
#define KMS_PER_MILE 1.609 /* conversion constant */

int main(void) {
      float miles,   // input - distance in miles
            kms;     // output - distance in kilometres

      /* Get the distance in miles */
      printf("Enter distance in miles: ");
      scanf("%f", &miles);
      // Convert the distance to kilometres
      kms = KMS_PER_MILE * miles;
      // Display the distance in kilometres
      printf("That equals %9.2f km.\n", kms);
      return 0;
}
```

Figure 3-1: Early program example

Nearly every character in Figure 3-1 is significant. For example, semi-colons must appear at the end of each command. The {} brackets indicate the beginning and end

[27] UNIX is an operating system, like Windows. Students use a secure shell to program in UNIX, which is a text-based secure window that operates like a tunnel that connects students to a UNIX machine, which resides and operates in a separate room or even building from where students are sitting. The shell allows students to use and interact with this machine. All interactions are text commands. For example, to see the files in a given directory students type “ls” in the window. A list of the file names is then printed out.

of the program. The % sign at the beginning of "%f" indicates it is a variable that is being output, rather than plain text. These are just a few of the many meaningful symbols contained within this short program that students learn to understand and write. This syntax forms the basic foundation of programming and works alongside the language's grammar through which students can build functional programs. Over the course of their studies students learn to build programs of greater and greater scope and complexity, which can include multiple files, languages, tools and Application Program Interfaces (APIs) (for incorporating or using other languages and programs within a program). In later years students learn to create task-organizers, mobile games, data analysis tools, and web applications, to name a few. If readers are feeling confused, take it as a small glimpse into students' early experiences learning computer science.

Each of the technologies/programs that students learn and create can be construed as "worlds" unto themselves, with their own rules, operations, logics or modes of thought, and sanctioned behaviours. Paul Edwards uses the similar concept of "microworlds," explaining that "every microworld has a unique ontological and epistemological structure" (Edwards 1990, 109). The term microworlds was previously used by computer scientist Seymore Papert in relation to the educational programming language Logo, which uses a "turtle" that responds to programmatic commands (Papert 1980).[28] Papert explains:

> It is in fact easy for children to understand how the Turtle defines a self-contained world in which certain questions are relevant and others are not… this idea can be developed by constructing many such "microworlds," each with its own set of

[28] See (Edwards 1995) for a discussion of different uses of microworlds as a term and concept, particularly in relation to mathematics and science education.

> assumptions and constraints. Children get to know what it is like to explore the properties of a chosen microworld undisturbed by extraneous questions. In doing so they learn to transfer habits of exploration from their personal lives to the formal domain of scientific theory construction (Papert 1980, 117).

Sherry Turkle also explores how hackers seek to create microworlds because of the capacity for control and mastery that they enable (Turkle 2005, 204); such worlds are made up of internally consistent and closed logic and so are necessarily partial representations or (re)creations of life and reality as they are continually lived and (re)made (Edwards 1990, 1997; Turkle 2005). My use of worlds is also comparable to that of multi-verses suggested by the Git workshop metaphor.

Some worlds are big and complex, such as UNIX, which has a multitude of commands and operations and can contain other worlds such as Vim, or students' own programs, which have their own sets of rules.[29] At the same time, these computing worlds operate within and according to the rules and logics of the digital computer, which constitute their "universe."[30] These metaphors suggest the operation of distinct realities – of different ways of being and doing – that provide insights for learning computer science. Yet, while different worlds are more or less distinct from one another and operate according to their own rules and processes, they can also interact. Programs are more or less self-contained, yet they can and do pass information to one another in specified ways and often rely on each other's functionality. I use "universe" instead to

---

[29] If we extend the metaphor of worlds, UNIX could constitute a solar system, or the sun at the centre of the solar system, which contains multiple orbiting planets or worlds that operate within that system. These details are not significant or necessary, however, for the purposes of this chapter and dissertation.

[30] Throughout this chapter I use the terms "computing worlds," "computational worlds" and "programming worlds." These terms refer to the same processes. However, I use programming worlds to talk specifically about the code and programs that students use and develop, and computing and computational worlds to encompass these programming worlds and all other worlds of technology and computation.

refer to the collection of these computational worlds that follow similar operations of logic, rule, and shaped by the operation of digital computers.[31]

In order to use programs such as UNIX and Vim, and to create their own programs – in order to function adequately in these worlds and create new ones – students must learn the rules and commands. Varying degrees of mastery or skill in these worlds are possible. One professor, for example, showed off in class his skill at using Vim, demonstrating the speed with which he could write and modify a program through simple text based commands, such as "w" for write, "dd" to delete a full line, or "gg=G" to fix the indentation of a program. In comparison, at the beginning, students have trouble remembering not to use the mouse, which does not work in the UNIX window, in addition to remembering the proper commands to achieve what they want. From personal experience, it can be tremendously frustrating when trying to move along a line to type, done by reflex using the arrow buttons only to have Vim interpret this as adding extra useless characters.

While the skill demonstrated by the professor shows the possibility for speed and accuracy, in the beginning the dizzying array of new commands and operations students

[31] These metaphors are imperfect. What I aim to highlight in discussing computing worlds is how making and using programs, programming languages, operating systems, mathematical theorems, and so on, entail learning, using, and creating distinctive realities in terms of both thought and practice. These worlds differ in particular ways from one another, and from the actual world. At the same time, a key point to understanding programming is the ways these programming worlds exist and operate within the same "universe" of thought and practice. The discipline of computer science centres on understanding, building, and expanding this "universe." What is less obvious with these metaphors, but is no less important, is the ways the computational "universe" and worlds within it significantly affect and intra-act with the "real" world.

need to learn can feel like smashing your head against a brick wall. Susan further described her struggles in her first semester:

> Sometimes I really got really frustrated, that I really don't want to continue doing it anymore because it's really painful looking at the computer every time and see the sea of red run through. You know like the error, right, for Python, whenever you got a recursion error then there would be this sea of red running through, yeah running through the screen. Then you got really scared (Susan 2014).

Susan was taking a different version of the introduction to programming module, where they learn the Python programming language instead of C. Yet, if specific rules are different, the basic struggles are the same: the need to master many details of syntax and grammar. In Python, for example, commands need not end with a semi-colon, but white-space such as tabs are meaningful, so careful attention needs to be paid to indentation in addition to the other syntax, grammar, and program logic. When students were working on their first programs in the school's programming lab, there was a constant dinging throughout the room as students struggled to grasp the right commands and the computers responded to invalid input.

In discussing how students must learn the appropriate logical, mathematical, and programming rules one professor referred to German philosopher Arthur Shopenhauer who said that "it is not possible to have a dispute with someone who denies the principles." The professor was emphasizing to students how you have to agree on the rules to play or program, as in a game. He further explained:

> It's like if you are playing Monopoly with your friends and you don't agree with the rules of Monopoly, or you are playing Scrabble with your friends and you don't agree on the rules of Scrabble, then it's very likely that the game is going to end up with a fight. So this is the same, it's like a game and I'm going to show you the rules of the game, and while we are playing the game we accept them. Outside of the game we can always disagree.

Learning to program is about accepting and following the rules of the game, or of the world. Students are encouraged to think critically, to question what professors tell them, to try things out for themselves, and to evaluate the advantages and disadvantages of particular coding styles and algorithms, a topic I will discuss further in Chapters 5 and 6. Yet, in the end they must do so within the rules of the world and follow them, otherwise they will not be able to complete their programs and have them compile and run successfully; they will not be able to play the game or operate in that programming world.

Language operates to constitute reality in particular ways, to create worlds of meaning and implication (Cohn 1987; Huizinga 1949). In discussing play and human language, Johan Huizinga discusses how every expression and metaphor is a play on words in which humans construct a "second, poetic world alongside the world of nature" (Huizinga 1949, 4). Computer languages evoke similar forms of play that Huizinga also sees in human language, or so-called "natural language" – languages not deliberately and formally constructed for working with computers. Yet, with computer languages, the "second, poetic world" is realized in the operation of the machine (Breslin 2013). Students and professors speak of these worlds as though they exist in space, beyond the physical space in memory and computational time that a program takes up. Programs and code are talked about as though they have a shape and substance. For example, students are told functions, which I discuss below, have a "territory" or a "scope." Certain functions have property, variables that they own and know about, but that other functions do not. Some data structures are in the forms of trees, with branches that can be traversed breadth-first or depth-first as different searching algorithms.

These abstractions help programmers conceptualize and manipulate data through their programs, but also work to build up shapes and structures in these programming worlds. Programmers build levels of abstraction on top of one another, so that machine code can be translated from higher level languages, which can be organized into “objects” or “classes” sometimes representing constructs from the actual world, and this object-oriented code (among other types) can be written to orchestrate financial transactions, the opening and closing of doors, the time displayed on a digital watch, simulations of life, or 3D graphical realizations of virtual worlds such as *Second Life* and *World of Warcraft*, for example (Berry 2011; Breslin 2013; Helmreich 1998; Mackenzie 2006; Malaby 2009). In learning to program, students are thus learning to become fluent in particular languages and particular modes of thought that constitute and enable particular worlds and realities.

3.1.2 Learning to Think

In addition to languages and programmatic tools, students learn algorithmic problem solving. Frequently, the significance of learning how to write algorithms to solve problems is emphasized over the particular technical skills – the particular languages or platforms – that students learn in computer science. Students are admonished, particularly at the beginning of their studies, to understand and think about a problem and create a plan for solving it before typing a single character. “Programming is not just about writing code” one professor emphasized as he was discussing an in-class exercise, moving on to say that students need to spend time thinking about the algorithm first. "Jumping into code first is sometimes anti-productive." A few weeks later, he told students that their Practical Exams (midterms done through coding at a computer) are designed so that students have a great deal of time to think, even if they only need to write a couple of lines of code. The basic practices of defining an algorithm and learning to write one in pseudocode (a description of an algorithm in near plain-English) are also covered in students' first week of learning programming, before introducing the basic syntactic symbols for students to write their first working program.

I saw and heard repeated evidence of how students took the significance of algorithmic problem solving to heart. Even students in the second semester of their first year emphasized how they had learned to think algorithmically, to analyze a problem, to break down a problem to smaller steps, and to devise a solution with step-by-step instructions for a computer to follow. Susan, who discussed her struggles to learn programming above, was emphatic about how she had learned to think differently by programming, and in ways that she transferred to other aspects of her life:

> I feel that programming, it has changed the way I look at things because I don't just jump straight to conclusions. Because programming has, it allows me to look at the problem and then solve it systematically and then, yeah, step-by-step approach… I'm not so hasty. Yeah. I don't take that hasty decision anymore. I learn how to analyze the problem properly and then think up a solution that works. It's just like how you look at a programming problem and then you think of the pseudocode and then from the pseudocode you can type out the code and all that (Susan 2014).

Another first-year student commented on understanding the thought processes entailed in writing a program:

> You want a computer to say take 23 out of 1000, and you need to know that a computer can only do things step by step, they can't be like us and look at the entire thing and take it out. Actually I think we do the same thing as what computers do – it's just being able to think of what you do and translate it into computer (Naomi 2014).

Students in later years echoed these statements. A second-year student, in discussing what she liked about computer science, commented: "Another side is about the thinking process, it's more [than] about programming, like algorithms and data structures, how you come up with smart ways to solve a problem. So it's more about problem solving and thinking strategies" (Alicia 2013). While not all students emphasized changes in their thinking, some also suggested that they already thought over problems carefully and in detail before learning computer science.

Both students and professors, however, emphasize the systematic, step by step, abstract process of problem solving as key to programming and, more generally, to doing computer science. Students are told when they start to learn that "it's all about logic. Every step must be clear to you." Unlike syntax errors, compilers generally do not catch logic errors, often leading programs to produce unexpected results. Compilers are programs that translate "high-level" languages (languages more or less understandable to humans) into machine language (binary). In doing so, a compiler checks that the syntax of the program is correct, because otherwise it would not be able to translate the program. Alongside programming, many students also learn discrete mathematics whereby students learn to produce proofs of particular mathematical theorems, similarly emphasizing step-by-step and logical thought. In doing so one theorem builds on another, eventually building the logical worlds of mathematical and programmatic thought discussed above. Yet, students must pay attention to the minute details of programming syntax and logical reasoning in order to build these worlds.

There is also a significant focus on problem solving as a key facet of computer science, as seen in Susan's and Alicia's discussions. Students were told in their first class that an algorithm is "a set of instructions to solve a problem." Early programming courses usually entail several forms of assessment, including weekly labs where students do multiple exercises, midterm exams (both practical and paper based), participation marks, and final exams. Labs or practical assignments are a key component of learning to program. Everything can make sense conceptually, but the experience of programming often makes students wrestle with the meaning of concepts, as well as the details of actually writing out the algorithm and making the program work. Most assignments are problem-based. Students are given a set of specifications and a problem that they have to solve with their program. Ideally, students must think about the key components of the problem and develop an algorithm to solve it and then implement that solution using code, while also testing the program to see if it gives the desired result in all cases.

Of course, not all students follow the ideal thought process and method. Often students will delve into writing out a solution in code before having developed a conceptual algorithm first. Regardless of which process students follow, however, the focus is on solving a particular problem. One of students' first assignments, for example, is to write a program that calculates an investment growth given a certain principal and interest rate. As seen in Figure 3-2, students are given precise instructions, including the name of the file and the name of the variables they should create, as well as the format of the output such as "two decimal places," which is meant to make the number appear as a monetary amount. While assignment instructions are not nearly so detailed in later years, the clear focus in learning programming is on following a set of rules and procedures to solve a particular problem, mobilizing computers' computational abilities, mathematical formulas, and various libraries and capacities that have been created and accumulated for computers such as text input and output, among others. These task statements thereby frame programming as a means for finding solutions to problems.

1.2 Task Statement
If you invest *principal* amount of money (in dollars) at *rate* percent interest rate compounded annually, in *numYears* years, your investment will grow to

$$\frac{principal\ \times (1-(\frac{rate}{100})^{numYears+1})}{1-\frac{rate}{100}}$$

dollars.

Write a program **invest.c** that accepts positive integers *principal, rate* and *numYears* and computes the amount of money (of type `float`) earned after *numYears* years, presented in two decimal places.
You may assume that the interest rate is always smaller than 100.

Figure 3-2: First-year problem statement

Progressively, students also learn about various forms of abstraction and encapsulation. These are presented as practical solutions to programming challenges, such as repeatedly rewriting the same code, which is generally seen as a waste of time and as producing code that is messy and hard to understand. Instead students learn to write functions, which are independent pieces of code that can be called on repeatedly to do something. The point is to abstract and factor out the commonalities between repeated pieces of code and write a function that is flexible and versatile in order to achieve that functionality in different circumstances. If successful, the programmer will then only have to write out one line of code that calls their new function multiple times, rather than repeating the many lines of code that make up the function. In later years students learn different "patterns" they can use in larger software development projects (Gamma et al. 1994). "The key is to recognize when things are the same" one professor explained. Students learn different ways of encapsulating pieces of code and creating particular relationships between them so that each piece can operate more or less independent of the other and pieces. For example, the graphical user interface, can then be changed or replaced without having to rewrite the whole program, at least in theory.

Entire programming languages, such as Java, are designed around these concepts of abstraction and encapsulation. Java is known as an Object Oriented Programming Language, where everything is an Object based on Classes with particular properties and functions, some of which are public and can be accessed by other objects, and some of which are private, hidden within the internal functioning of the object and inaccessible to others. Most students learn Java beginning in their second programming course and are thereby taught this mode of thought whereby code should be abstracted and encapsulated so that a program becomes a series of ever larger and encompassing black boxes with particular public functionality accessible to those outside the black box. Information can be piped or passed from black box to black box to produce the ultimate desired results. Each black box works as a tool to produce a particular end.

This concept of encapsulation – creating black boxes – extends to other facets of students' studies in computer science. In discussing how students should write a mathematical proof, one professor suggested that students should use the best tools available to them to answer the assignment questions. In particular, students can "use a theorem even if you don't know how to prove it" saying it was like most students do not know how to build a TV, but they use one all the time. The TV and the theorems are black boxes that can be used to particular ends. Similarly, the programs – the worlds students have learned – themselves can be seen as black boxes with only public functionality that is accessible while the internal operation, the code, is hidden. The hidden functionality is partially why Open and Free Source Software is so significant to many programmers, because they can look inside the "black boxes" of particular programs, see how they work, identify their limitations and the possibilities they afford, and even change them (Coleman 2013; Kelty 2008).

In 2006, Jeannette M. Wing introduced the term "computational thinking" as a kind of manifesto for an "analytical ability" that should be added to children's learning alongside "reading, writing, and arithmetic" (Wing 2006, 33). There have, however, been previous calls for programming or computing to become an ubiquitous part of education; Turing Award winner Alan Perlis, for example, called in 1962 for all students on campus to learn to program (Guzdial 2008, 25). Wing and Perlis both argued that thought processes associated with computing and computation could and should be applied to understanding a wide variety of disciplines and subjects, and there are other and earlier examples of proponents making similar arguments (Denning 2009, 28–29). The arguments about the thought processes involved in computer science and the application of these thought processes to other facets of life by students and professors, discussed above, echo those advocating "computational thinking" and related processes. While

Peter J. Denning argues that computer scientists are valued for their "computational doing" more than a process of thinking (that is not necessarily unique to the discipline) (Denning 2009, 30), the promotion of computational thinking as underlying the thought processes of multiple (if not all) disciplines and practices supports and encourages the value of rendering technical and related practices.

The thinking processes that students learn in studying computer science thus may not be unique to computer science (Denning 2009), but they are performative. Through discussing, learning, and practising these particular modes of thought, students learn to understand and build within the computational universe. They learn to create algorithms, focusing on creating step by step instructions that solve particular problems. Concomitantly, what they learn to do with their programs or algorithms is just that, to solve problems. This approach necessitates a frame of reference as formed of problems and solutions. Students also learn about abstraction and encapsulation, creating programs such that they are formed as a series of black boxes that interact in only particular pre-defined ways, with internal workings hidden from view.

3.2 Rendering Technical and Rendering Natural

"I hope this kind of logic is natural to you now" a professor commented to students as he was reviewing a particular programming method. The technicality of programming and computer science thought – the rules, the systematicity, and the problematization – becomes naturalized as students learn and reproduce it. Students learn how to represent and translate reality into models, algorithms, and code. The rules of writing programs themselves are also presented as an inherent part of how computers work and, by extension, how humans work with computers. As a result, they appear natural and immutable, creating a tension between computer science ideas about independent learning and critical thinking, and the need to accept programming rules as "just the way things are," discussed further in Chapter 5. Students who cannot accept the rules are unable to understand and operate in the different worlds; they cannot play the game, to put it another way.

3.2.1 Translating Reality

Along with programming languages and algorithmic problem solving, students learn various forms of representation for both code and for reality. At first, students learn how to trace through the operation of code, to model by hand what would happen if the code runs. This is displayed in various diagrams drawn on whiteboards, printed on PowerPoint slides, or scribbled on pieces of paper when answering an exam question or trying to figure out why a program is not working. Generally variables are given little boxes. A string of characters has multiple little boxes beside each other. A particular variable type, known as a "pointer," refers to other variables. In terms of computer operation, pointers contain the address of another variable in the computer's memory, but when tracing a program these are represented by their own little boxes with arrows that point to the boxes of other variables. Arrays, essentially lists of elements, are represented similar to strings of characters with multiple boxes beside each other. Arrays can be multi-dimensional, however, so a two-dimensional array is then represented by a table or matrix.

The representations become more complex as the data structures that computer scientists conceptualize become more complex. Tree elements, known as nodes, gain colours. Variables are no longer small boxes, but Objects represented by large boxes with other smaller variable boxes inside of them. When tracing a program, the contents of a variable are scratched out, and replaced with the new values reached after an operation such as an addition or subtraction. As mentioned in relation to the discussion of spatial metaphors, these representations go beyond the physical form of data in the computer and work as abstractions and representations that help programmers and computer scientists conceptualize the data and their programs.

Both the variables and their representations often stand in, not just for a data structure or an extra-local abstract representation of machine operations, but also for actual world constructs. As seen in Figure 3-2, variables stand in for various concepts such as a principal amount of money, a rate of interest, and the number of years for an investment. Students later learn to model relationships between variables, and between such actual world constructs. Figure 3-3, for example, shows a representation of Marriage in Unified Modelling Language (UML), which all students learn over the course of their studies. UML provides a representation of the various objects, variables, and functions, and how they interact, within a program. This example was given to students in course notes to show them how to model relationships between classes (abstract versions of objects), such as the “married to” association embodied in the Marriage class between the Man and Woman classes. In the UML representation there is a Man, Woman, and Marriage box, each corresponding to a separate class in the program code. A separate class “Marriage” is created because a marriage has properties and operations that are distinct from either Man or Woman.

It is also an over-simplification of how such relationships would likely be modelled in a working program, provided in this way to illustrate a concept. Nevertheless, it clearly shows how particular assumptions about the construction of reality can be written into programs. Marriage is represented here as occurring only between a man and a woman and each Man or Woman can only have one Marriage to one entity of the “opposite” gender.

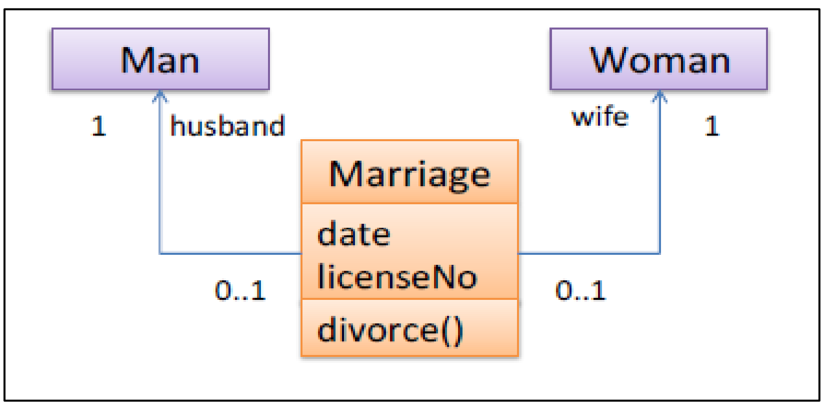

Figure 3-3: UML representation of marriage from second-year course notes

A great deal of research has explored and critiqued the ways in which particular biases and assumptions are both implicitly and explicitly built into the design of technologies, often in relation to gender but also considerations such as how human emotions are constituted and represented, physical abilities and disabilities, and the constitution of expertise and knowledge (Alsheikh, Rode, and Lindley 2011; Berg 1999; Forsythe 2001; Huff and Cooper 1987; Oudshoorn, Rommes, and Stienstra 2004; Suchman 2011a; van Oost 2003). Much of this research focuses on Human Computer Interaction (HCI) and, in particular, the interfaces and other areas of encounter between human users, and computers and other technologies. Yet, as seen in the example above and shown by science and technology studies scholars, the underlying theories, designs, and code developed by scientists, engineers, and computer scientists also render the world in particular ways (Keller 1985; Li 2007; Martin 1991; Myers 2014, 2015).

Representing "nature" and "reality" is an integral part of scientific and technological practice (Coopmans et al. 2014; Lynch and Woolgar 1990). Computer science students are taught how UML diagrams such as the ones above are important for planning and designing large programs and specifying their functionality. Representations in UML, algorithms, data structures, code, however, are also all performative renderings (Mackenzie 2005). In particular, many (although not all) computer languages abstract and objectify facets of the actual world constructs. Every entity represented is turned into an object in Java and in other object-oriented languages, for example, similarly seen in UML. Through these modular computational worlds, things and relationships become explicitly specified and solidified into stable representations.

Moreover, these representations work as ways of developing "solutions" to predefined "problems." Horst W. J. Rittel and Melvin M. Webber identify "wicked" versus "benign" problems in relation to the politically and socially complex challenges that face planning and policy work (Rittel and Webber 1973).[32] John Law further

---

[32] They outline ten properties of "wicked" problems, which essentially explain the ways that such problems cannot and should not be rendered technical. Their sixth property, for example, states that "wicked problems do not have enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-describable set of permissible operations that may be incorporated into the plan" (Rittel and Webber 1973, 162). While programming languages do have a clear and describable set of operations, in their

suggests that "the problems of the world are *always* wicked" and that the only way to handle or "tame" them is to treat them as benign, and so the interesting question is *how* wicked problems are rendered benign (Law 2014, 9–11). In this regard, defining a problem is for computer science the first step of this "taming," which is done, in part, through these various forms of representation. Ultimately, these representations – and the computational worlds that they produce – become the solutions to the problems that were defined and created through their making.

There are multitudes of technological "products" that address "problems" of varying complexity that exemplify the results of this process; examples from international and Singaporean news media include programs/apps that are meant to eliminate unconscious bias through creating anonymity (Bereznak 2017); that stop women from apologizing frequently and thus undermining their authority (Cauterucci 2015); that help women walk home safely at night (Russon 2015); that address issues related to haze (Chia 2015); and that determine when laundry machines are free in a dorm or apartment (Jeffrey 2016). The preponderance of "solutions" relating to women listed here is largely due to my interest in the topic. However, the framing of these solutions as intervening in or helping women's identities or behaviours illustrates the ways "problems" and "solutions" become defined through technical renderings as based on individual encounters mediated by technologies or programs, treating symptoms rather than addressing underlying causes such as patriarchal norms or, in another case, the

---

workings as world-making actors, code, programs, and computers intra-actions with reality are not reducible to a predetermined set of problems and solutions.

social, political, and economic circumstances that lead to the production of environmental haze.

Additionally significant here is how these diagrams, code, and programs are meant to represent both machine and human practice and language. It is a mediation and a translation between the complexity, nuance, ambiguity, and detail of human life, captured and modelled in variables, classes, objects, and relationships that can ultimately be translated into series of binary instructions that are transmitted through multiple circuits to produce computations that are then retranslated by computer hardware and software to register a marriage in a database, display a congratulations message on a screen, or print out a marriage certificate. In other words, the worlds that computer scientists build are both filtered reflections of constructions in the actual world and performances that constitute part of that world; these worlds intra-actively constitute reality as it is being represented, the technologies involved, and the persons developing and using them.

Following Donna Haraway, these worlds are diffractions: "the noninnocent, complexly erotic practice of making a difference in the world, rather than displacing the same elsewhere" (Haraway 1994, 63). Tara McPherson (2012), for example, suggests that the tendency towards modularity embedded in computing, and stemming in part from the development of the UNIX operating system, is intertwined with the constitution of race in the US. Computer scientists learn to sit at the middle of and orchestrate this mediation, manipulating the representation of reality and then its reconstruction through computer operations rendered into computing worlds. Feminist scholars have thus argued that computing practices and the technologies they create need to be accountable to the worlds they produce (Rommes, Bath, and Maass 2012; Suchman 2002; van der Velden and Mortberg 2014). I discuss in the next chapter the gendered renderings of computer science knowledge and practice. Here, I consider the implications of this world-building in relation to how the history of computers and computer languages are taught and experienced by computer students.

3.2.2 Ahistoricism and Acontextuality

Computers, operating systems, programming languages, text editors, and compilers were all created by humans in particular historical and cultural contexts. Alan Turing is well known for developing his Turing Machine in 1936 which provided the theoretical basis for conceptualizing modern digital computers (see Ensmenger 2010, 30–31). Early electronic and digital computers such as the Z2 and Z3 in Germany in 1939 and 1941, the Atanasoff-Berry Computer in the US in 1942, Colossus in the UK in 1943, and ENIAC in the US completed in 1945, were all created and developed during World War II. The Z3, Colossus, and ENIAC were the first electronic programmable computers, enabling calculations to be done at much greater speeds to quickly solve mathematical equations such as basic linear mathematical equations, or integrating ballistic equations and trajectories of naval shells in the case of ENIAC (O’Regan 2012, 43). Colossus worked towards cracking German coded communications. These were literal computers, working to produce computations, sometimes replacing women who previously worked as computers to calculate equations such as ballistic trajectories (Abbate 2012; Haigh 2010; Light 1999).

Later, throughout the 1950s to 1970s, computers were further developed for military purposes against the perceived threat of the Soviet Union during the Cold War, as well as for a growing commercial industry in making computers for business purposes (Ceruzzi 1998; Edwards 1997; Ensmenger 2010b; Hicks 2017). The computer's functionality continuously expanded, moving from their focus almost exclusively on calculations to focus on information processing, including payroll processing, data processing, and information management (Ensmenger 2010b, 58). A movement for "automatic programming" was also taking place as an effort to reduce the difficulty of writing programs and reduce or eliminate the tedious work of writing code (Ensmenger 2010b, 83–84). This movement never succeeded in eliminating programming work, but did help make programming languages more human-readable compared to machine code (Ensmenger 2010b, 84).

Modern digital computers rely on electrical signals to indicate bits of 1s and 0s – binary signals. To perform operations, processors contain a series of "gates" that perform various logical operations in inputted binary numbers. For example, AND checks if the bits of two numbers are the same, XOR checks if the bits of two numbers are exclusively different, OR checks among each of the bits if one of them is 1, and so on. These operations can be combined together to produce various calculations such as addition, subtraction, and multiplication. All of the functionality of our current technologies is based around layers of binary data and operations, with specific locally defined meanings translated into hardware and further layers of software functionality, as discussed above, to achieve everything from storing a text file to the display of a video on the screen. Early computers were programmed in their native machine-language, consisting of binary. These are read directly by the computer, with the various bits representing different operations and data.

However, humans do not easily read machine code and so writing it is very tedious and prone to errors. Additionally, machine code is dependent on the machine hardware. A program for one machine cannot be used on another. “Higher-level” programming languages were then created that could be more easily read and written by humans. Instead of being sequences of 0s and 1s these contain keywords representing operations such as “Add,” as well as variables used to represent data. The most basic form is known as assembly languages. While these types of languages are much more readable than binary, they remain difficult to use as they still closely resemble the basic operations of the computer. Later languages are much more intuitive for humans with variables that can be named for easy identification and structures such as “if” and “repeat” that make certain common functions much quicker and simpler to write. Both assembly and “high level languages,” as the latter are known, ultimately need to be translated into machine code, which is the work of an assembler or compiler. Thus, computer languages and compilers or assemblers are often co-created.

FORTRAN was one highly successful language developed by IBM and released in 1957 primarily for scientific and mathematical calculations (Ensmenger 2010b, 90–91). The language was created with a focus on the compilation process, such that the machine code produced would be at least as efficient as that created by humans. As a result, there was less effort placed on the design of the language itself (Ensmenger 2010b, 91). These foci meant certain other functionalities, such as manipulating alphanumeric data, were quite difficult (Ensmenger 2010b, 91). The C programming language was first written at Bell Labs from 1969-1973 for the purpose of developing the UNIX operating system. As one of the primary authors of the C language explains, the C language was developed by a small number of people, drawing inspiration from previously existing languages such as FORTRAN and ALGOL 60, and building on BCPL, B, and NB which were previous iterations of languages created alongside and for the development of UNIX (Ritchie 1996).

Early developments of C and UNIX contended with significant limitations such as the size of memory and access to useful software for development. The capabilities of computing hardware, and the functionality provided, also significantly shaped the language that was developed: “They are ‘close to the machine’ in that the abstractions they introduce are readily grounded in the concrete data types and operations supplied by conventional computers” (Ritchie 1996, 673). Many resulting features of C that are critiqued and that are difficult for beginners were also a result of the history of development of the language, carried over from FORTRAN or from the previous iterations of C, which needed to be maintained to allow for compatibility with programs that had already been written using earlier versions (Ritchie 1996, 683).

Despite the challenges of the language, C and UNIX are used as part of first-year education at Temasek University, as well as many other places, partially as a result of their history also as tools for computer science pedagogy. Christopher Kelty discusses how UNIX became the “paradigmatic object” for computer science students to use and learn because, with its spread in the 1980s, it was the first operating system to offer the source code along with the program itself. Additionally, as an operating system purposefully built to be small, it was simple enough for students to explore the inner workings within the framework of one or two academic courses (Kelty 2008, 129). C is the main development language for UNIX. While students do not learn directly about operating systems until later in their studies, they are inducted from the beginning to using UNIX as an operating system for doing programming work in C. These are not necessary choices for learning programming or computer science. One professor explained to me that they have considered and tried teaching different first-programming languages, including Java. Additionally, other versions of the introductory programming course, for science students or for accelerated learning, use different languages including Python and JavaScript. Nevertheless, C persisted at Temasek as a frequent choice for teaching the core programming course to computer science students, likely in part because of the historical uses and associations of C and UNIX.

It is clear that the particular purposes and circumstances, including the hardware available and its functionality, as well as the ongoing development and iterations of the language shaped how C works and its various functions. Similarly, computing hardware itself developed out of international competition during World War II and the Cold War when there was much focus on completing computations for code breaking, military purposes, and scientific research. Students, however, do not learn this history. Often professors will mention who created or proved a particular theorem or developed a language or technique. Students also learn about computer hardware and work to understand and create parts of their own operating system in an upper-year course. However, the history and social and political context of why particular languages operate in particular ways is largely excluded from discussion with students who are first learning to program, and even in later years for those who do not specifically take upper-year courses on Programming Languages or Compiler Design.

There are obvious reasons for this silence. As a largely technical course, the focus is on learning *how* to program as a particular technical skill. Covering the basic structures and processes for building a program within a semester, amounting to 13 weeks, leaves little room for seemingly superfluous topics such as historical context. Yet, in presenting the rules of programming independent of this context, they become naturalized. The particular syntax, structure, operations is just how things are, and how they have to be. In practice, in order to learn to program in C or another language, this is precisely how students have to understand and approach programming.

For example, near the beginning of the semester, as students were learning basic syntax, one student asked what would happen if he did not include the curly brackets, “{“ and “}” at the beginning and end of the program, as seen in Figure 3-1. The professor answered he was not sure, since he had never tried, having himself accepted the rules of coding in C and following them with dedication. Similarly, in discussing Boolean operators (False and True) the professor explained that these are represented by numbers (0 and everything but 0, respectively) because of the particular version of C they are using, commenting to students “Very strange right? But this is how things are like.” He thereby emphasized that students need to accept the rules, even if they are strange, because that is how this version of C works. C thereby becomes a natural world with inherent and immutable rules, a "magic circle" wherein students must accept the rules of the game in order to play, as discussed above (Huizinga 1949).

Agency for making and enforcing the rules has also been transferred to computing worlds – to the programs themselves. The professor who says above he never tried leaving out curly brackets, further commented to the student that likely the compiler will complain about the missing characters. Compilers were created by humans with specific rules and limitations to ensure that programs can be successfully translated and operate on the machine. The compiler therefore works to enforce the rules of the game; it is the arbiter of correctness.[33] It is also common in speaking about code to delegate the action to the program itself: it is this program, that function, or that variable that is acting – a computer science performance of Bruno Latour's nonhuman "actors" (Latour 2005). In discussing the operation of a function, for example, one professor explained "this guy [this function], his job is to get two guys [two other functions] to help him." Another professor explained the elements and operations on a tree-like data structure using the university administration structure as part of the example, explaining it contains paper-pushers, professors, "alcoholic Deans," and so on, as anthropomorphized allegories of the tree node operations. He suggested that, for example, the Dean wants to do something relating to Sustainability and does a bit of work, then sends it out to the Department Heads who are lower "nodes" in the tree. Or, a boss wants a report done, passes it on to

[33] Programs that pass the compilers' tests will not always operate as expected, since the functionality also depends on the logic and correctness of the program itself. Occasionally compilers will give a "friendly warning" if it detects something that might indicate such a logical error.

his subordinates, who continue to pass it on until it reaches the poor guy who has nobody else to call on anymore – a "leaf node" – and he will have to do the work.[34]

Similarly, most of students' programs in their first year are first marked by an auto-marker, a program that tests students' programs on various input to see if it functions correctly. The agency for arbitration of correctness, as with the compiler, is then shifted to the marking program rather than to the professors and TAs, although the latter do provide feedback on coding style and program structure. The minutiae of rules indicated in early problem statements seen in Figure 3-2 emphasize the significance of following rules. The output of students' programs, in particular, must be precise, down to the correct number of spaces between characters, in order for the auto-marker to be able to analyze the output. These are part of the rules of the auto-marker "world." This practice works to accustom students to following particular rules and attending to minute details, even if these particular rules are more obviously created by the professors and TAs in alliance with the auto-marker. The concomitant effect is to naturalize computers and their functionality as technical objects. The human agency and social, cultural, and political context for creating machines and languages in particular ways is hidden and blocked from view.

Latour (1987) argues that scientific and mathematical equations are forms of translation, integral to the process of network building. They work to increase the mobility and combinability of encounters by creating "traces," by translating reality into

[34] The use of masculine pronouns here is deliberate and reproduces the language used by most professors in class. I discuss this gendered language usage in Chapters 4 and 7.

different forms. Yet, while mobility, stability, and combinability can be gained – Latour's properties of immutable mobiles (Latour 1987, 223) – often something is also lost or, as Barad suggests, excluded (Barad 2007, 19-20). Computers and computer languages are translations and diffractions, of social/political/cultural relationships, into technical form (Haraway 1997, 11). These translations encompass multiple intra-actions of technologies, persons, organizations, and practices. Creating the C language involved past languages, past operating systems, the capacities of particular hardware that was available, the support of Bell Labs, and the expertise of the people who contributed to the development of the language and of UNIX (Kelty 2008; Ritchie 1996). Throughout the 1980s, various versions or "dialects" of the language were associated with different machine hardware and associated compilers (Ritchie 1996, 681).Yet, eventually efforts were made to formally standardize and stabilize the language (Ritchie 1996, 681–83), where networks were "cut" and C solidified into an entity or phenomena – an immutable mobile (Latour 1987; Strathern 1996).

In addressing the concept of networks from Actor Network Theory, Marilyn Strathern considers how these networks entail endless flows or connections of humans and nonhumans, but that, at some point, these flows must be stopped, not least for the sake of analysis (Strathern 1996, 522–23). Drawing on Jacques Derrida, she uses the concept of "cuts" to describe how one phenomenon "stops" another (Strathern 1996, 522). As discussed in the introduction, Barad similarly addresses the question of the boundaries of "apparatuses," which are similar to but not the same as networks, pointing out that the boundaries and properties of an apparatus are not clearly defined and only determined in their relation as part of broader phenomena (Barad 2007, 160).[35] These boundaries are produced through "agential cuts," in intra-action with other apparatuses (Barad 2007, 348).

In this case, decisions around curricula and the boundaries of the computer science discipline cut these extended networks to encompass the language and technologies in their present states, but not the earlier versions that they are built on or the organizations, persons, and practices that contributed to and intra-actively shaped their development. Similarly, apparatuses of production based on the intra-actions of silicon, metals, chips, electronics assembly workers, plastics, operating systems, software, and programmers are continually being cut by apparatuses of branding, marketing, and sales to create "digital computers" as "products" that work themselves as immutable mobiles that can be bought and sold.

These cuts and renderings also rely on and reproduce a separation between social and technical frames. In his exploration of computer engineers in the US, Gary Lee Downey discusses how dominant images of technology see it as an external force "rescuing humanity or guaranteeing human progress through automation" (Downey 1998, 3). Alternatively, but with the same implications, technology is seen as a negative destructive force. In either case, technology is imbued with deterministic and independent qualities; it has the power to shape society while not being social or cultural in and of

[35] Barad explains the essential difference between apparatuses and networks: "Apparatuses are not assemblages of humans and nonhumans; they are open-ended practices involving specific intra-actions of humans and nonhumans" (Barad 2007, 171). The key difference is that, for Barad, humans and nonhumans are not separate stable entities, but are themselves co-constituted and distinguished as part of and through their intra-actions.

itself. Similarly, Bruno Latour (1993) has argued that the "modern constitution" is built upon continuous work of "purifying" and separating categories of "nature" and "culture." This separation brings about disciplinary distinctions between science and sociology/politics, each involved in the "distinct" realms of nonhumans and humans, respectively (Latour 1993, 27–29). From this "modern" perspective, computer science is concerned primarily with nonhumans in the form of technologies.

Most students had a more complex approach to technology than one of simple technological determinism or one where technology was conceived of as completely separate from society and culture, particularly if I pressed the issue. For example, second-year student Xiaowen discussed how he wanted to help society, saying:

> **Xiaowen**: It could be a lot of ways like you know all the social problems like poverty, you know freedom of speech, communication, so many problems in the world, medication, all the diseases. It's not easy, it's not going to be easy to solve – I don't have any ideas how to solve it, but that's why I'm here. Get that knowledge, get the insights and eventually hopefully I'll get there one day…. Computing is everywhere. You can do anything, you can do *anything*. That's why I *love* computing. That's why I'm in this field, because you can do anything, you're not limited.
> **Sam**: Do you think computing can solve every problem there is out there?
> **X**: Not just computing. Computing cannot solve every problem by itself. Yeah, so you still need the others – you still need like all the other sciences, mathematicians, uh like psych people, you know, those people. (Xiaowen 2014)

Xiaowen acknowledges that computing is insufficient to solve all social problems, although he is enthusiastic about the power and possibility it provides. Likewise, I encountered multiple visions of technology associated with the Internet and social networking that provide different models for understanding the relationship between technology and society. I was surprised while sitting at a product demonstration by Palantir, a Silicon Valley company with offices in Singapore, when they explained their approach to technology as something they developed to work in tandem with humans. “Reducing friction [between human and computer] is what Palantir is all about,” the presenter explained, suggesting a model for technology different from one where technology is seen as a distinct “thing” that works as an overriding positive or negative force. Rather, technology was being portrayed in this case as a resource, but also almost as a partner in working on “problems that ‘really matter’ without clear solutions.” Technologies here are mobilized as separate entities, but ones that work with humans more or less successfully depending on “frictions.”

A common thread in Xiaowen and Palantir’s discussions, however, is the focus on problems and solutions. As discussed above, approaching the world in terms of “problems,” even if the solutions are not clear, is intertwined with the practice of rendering technical, which works to remake reality in terms of problems that students can know, manipulate, and solve, and defines the universe of possible solutions. This practice helps effect a boundary between technical and social, by effecting the “purification,” discussed by Bruno Latour (1993), between humans and non-humans, culture and nature, and social and technical. The clearest example of this process comes from the way computer science treats text algorithms related to information retrieval and searching (like Google search), social media analysis, classification (such as spam filters), and developing knowledge-based systems, among other applications. Semantics are generally not included as part of the processing and analysis of text. Rather, individual words are made comparable and countable. Generally words such as “and,” “am,” “you,” “he,” “where,” and so on that have little inherent specific meaning are discounted and eliminated. For the English language, in order to make words comparable, they are also stripped of their conjugations, a process called "stemming." "House" and "housing" become simply "hous." The words are then counted, weighted, and compared to identify similarities, distinctions, and to categorize, select, or discount.

Rendering words comparable and countable is just a small part of complex algorithms. Algorithms such as Google Search are highly useful and accurate. Nevertheless, it is a clear illustration of the literal stripping of meaning that occurs through a process of rendering technical. N. Katherine Hayles (1999) similarly discusses how the very definition of information that became prevalent in computing and cybernetics was developed to be entirely separate from informational meaning: "a simplification necessitated by engineering considerations becomes an ideology in which a reified concept of information is treated as if it were fully commensurate with the complexities of human thought" (Hayles 1999, 52). In this way information, text, money, people, categories, and ultimately reality can be manipulated in the technical/computing worlds.

As discussed, however, this practice entails cuts where often something is excluded – history, meaning, context, and embodiment, for example – in order to gain mobility, stability, and combinability of knowledge/reality (Barad 2007; Latour 1987).[36]

[36] As discussed above, the "hybrid" networks or apparatuses that are built in learning and doing computer science, are cut as technological practice is translated into "products" or technological "objects" (Barad 2007; Latour 1993; Strathern 1996). As a third-year student commented, after I asked about the role of computer scientists in determining the impact of their software: "Once you actually release something, some code or like some program or some software, you cannot handle it. You cannot determine whose hands it should go in or whose hands it cannot go in. Of course, you can regulate, place regulations. But we know, despite that, it's not going to stop people. Like for example you've got peer to peer, you've got bit-torrent, for example" (Aindri 2013). Latour suggests that it is the paradox of the "modern constitution" to

Returning to the making and teaching of particular technologies and programming languages, in making tracings and cuts, the historical process, the military context of creating calculating machines, the encounters with limited hardware capabilities, the experience of using language predecessors, the logic and reason applied in creating particular language rules or structures, the historical pedagogical processes of computer science as a nascent discipline, and the context, meaning, and construction of reality being processed by programs are excluded. Yet, in the process, C, digital computers, and other computing worlds gain extensive mobility, stability, and combinability, moving across geographies, nations, time, context, and cultures to form the context of computer science education in Singapore in the 21st century. The diffraction of these movements is seen as computational thought and practice are reinterpreted as based in the evolution of human thought and natural environmental processes.

3.2.3 Computing as Natural History

In addition to technological and computing language apparatuses becoming solidified into objects that are treated as worlds unto themselves, computational and mathematical thought and practice are often represented as natural and essentially human in and of themselves. One event in particular stood out to me that exemplified this perspective. I attended a public lecture by Kent Beck, a well-known American Software Engineer who worked at Facebook at the time and who has promoted software

---

create such hybrid networks while disavowing the connections that are built: "the modern Constitution allows the expanded proliferation of the hybrids whose existence, whose very possibility, it denies" (Latour 1993, 34). As in Aindri's comment, once a technology is "released," it is an object that can be used for any purpose its users can find for it. It is seen as separated from the intentions and networks of developers and other humans and nonhumans involved in its creation. Technologies are thereby rendered natural, as I discuss next.

development methods such as Extreme and Agile programming. Other students and I were told of the talk in one of our lectures and encouraged to attend given his prominence. The title of his talk was "The Nature of Software" and at the beginning of his presentation he explained the "heuristic" for his talk would be "one startling sentence" that "programming is best viewed as a natural process." He provided a comparison, which he repeated several times throughout the presentation, that just as river deltas were created with no master plan but have developed into large scale and beautiful patterns, the same is true of programming. His presentation further elaborated how multiple programming practices follow statistical Power Law distributions, which could be found almost everywhere in nature. He made comparisons to other natural phenomena such as typhoons and hurricanes, and he was amazed by how such patterns just "happen."

While it was not a topic that arose frequently in classes or among students, with teaching and study focused largely on technical issues, in the few other circumstances when the nature of computing was discussed, it was often seen as a natural part of human thought and practice. In particular, the modes of reasoning entailed in doing computer science are seen as a part of natural human thought processes, and a result of human evolution. There is thus an interrelationship assumed between human thought and computational thought. As Naomi suggested above: "Actually I think we do the same thing as what computers do – it's just being able to think of what you do and translate it into computer [language]" (Naomi 2014). In this view, humans are like computers, but the particular thought processes, the steps to achieving certain goals or actions, are not explicitly conceptualized, which is necessary for computers.

Sherry Turkle has explored how the computer works as an "evocative object" that promotes human reflection of our selves and leads us to develop images of ourselves as machines or as "feeling computers" (Turkle 2005, 285). Artificial intelligence researchers, for example, develop AI processes based on analyses of how the human mind functions in terms of information processing, such that AI is "not about building machines but building a new paradigm for thinking about people, thought, and reality" (Turkle 2005, 244). Similarly, Lucy Suchman has explored how Artificial Intelligence researchers' attempts at making human-like robots that mimic or reproduce human emotion also reproduce and normalize universalized essentialist, reified, and categorical understandings of emotions and their expression (Suchman 2011a). The demonstration of these emotions – following emotional categories such as anger, fear, and excitement – (re)create emotions as based on these categories (Suchman 2011a, 128-129). In this way, the human interactions with robots (re)produce particular – often narrowed – ways of understanding and creating ourselves as humans, as much as they are scientific and engineering productions of entities with certain abilities or properties.

In similar ways, thinking about the reasoning entailed in programming and doing computer science leads professors and students to think about processes of human reasoning. One professor, in particular, in trying to help students understand the thought processes entailed in a particular form of mathematical reasoning discussed how, although he is not a psychologist, he guesses that very early on the brain starts developing the ability for logical reasoning. The point was that logical thought is something students are inherently capable of, even if they are finding it difficult at this particular moment. The professor located this ability for logical reasoning as part of early human evolution. He told a story about how when human ancestors were living in caves, they would go outside and while they may not yet have conceptualized numbers, they could see groups of horses, their tribe, apples, and oranges, and distinguish these as different categories or sets. They could likely also understand correspondence, such that each man and woman could ride one horse, or match horses with each of their fingers. "I think [it's a] basic mechanism of human thinking" the professor commented, and to make these comparisons and groupings as "quite a natural thing to do." Part of doing computer science is then seen as conceptualizing this sort of basic human reasoning in a formal way.

The professor was clear in emphasizing that these were just "stories" or "pseudoscience." Yet, these stories are performative in that they suggest and produce a picture of human thought and of computer science as based in human reason from "prehistoric times." Additionally, the abstraction and formalization entailed in mathematics and algorithms – the computational universe – is thereby tethered to the construction of reality.[37] As seen above in the discussion of various forms of representation in computer science, the constructs from the actual world become translated into diagrams, symbols, computational objects through multiple processes. Reality is rendered technical. By situating computational thought as part of human

[37] Cross-cultural research with the Pirahã who have an anumeric language and often cannot distinguish numbers greater than three suggests that numeracy and set correspondence develop through the use of language that includes numerical concepts (Everett 2013; Everett and Madora 2012).

evolution, or programming practice as river deltas, computer science is then rendered natural.

"Nature" is not a neutral category. As Latour has argued, ideas and practices of "modernity" have been strongly implicated in creating and maintaining a distinction between "nature" and "culture" (Latour 1993). Such ideas are tied to multiple other dichotomous distinctions including subject-object, human-nonhuman, and self-other, among others, as discussed in the Introduction. From this "modern" perspective nature is something out-there to be discovered, appropriated, managed, and controlled (Haraway 1991a, 1994; Keller 1985). Practices of naturalization then work to reproduce these distinctions, positioning computer-scientists-in-the-making in relation to computational worlds as the objects of their study and practice. In other words, naturalization does rhetorical and performative work to position technologies, computers, code, programs in particular ways in relation to computer scientists and in relation to the construction of reality.

Despite the significant affective dimensions associated with learning and doing computer science, some of which I discuss in Chapter 6, explicit discourses about computer science knowledge production prioritize logic and reason. Professors often emphasized the significance of logical thinking and reasoning for many tasks including writing algorithms, proving the validity of theorems, and writing clear and good code. In emphasizing how to write a good algorithm, a professor for a first-year course commented, "if your algorithm is messy, if your idea is messy, then your code is messy. You have to start with a clear mind." Throughout the course this professor would often comment how students need “logical thinking” or to “think logically.” Code, algorithms, programs, and computing in general is meant to be constructed from formal mathematical and logical thought; computing worlds are meant to *be* logical, and therefore knowable and able to be assessed, evaluated, and judged.

As suggested above, this is not to say that there is another distinct or objective reality “out there” separate from code and programs. Rather, naturalizing programming worlds, as much as the programs and code themselves, “*rend* the world in particular ways; they pull, tear and torque the world in some ways (if not others)” (Myers 2014, 155). Seeing code as part of nature shapes the ways in which programmers intervene in coding worlds and the ways in which coding worlds intersect with the ongoing construction of reality. Rendering technical and rendering natural are two facets of the same process, each feeding into the other. Reality is rendered technical as it is translated into mathematical theorems, diagrams, data structures, algorithms, and code. These translations are then rendered into “natural” computing worlds as if reality was always mathematical, categorical, and computational – as if the computational universe and the actual universe are one and the same – and this is now being realized by computer scientists and their work.

Similar perspectives are held by cyberneticists and artificial intelligence researchers, some of whom envision the universe as a giant digital computer. Human thought and action are thus simply one process among many organizing the overall program of the universe (Hayles 1999, 240–41; Helmreich 1998, 65–83). Rachel Douglas-Jones and Christopher Gad, for example, identify an ontology constructed through “slippages” in meaning about computational thinking, which work to “enact computational thinking simultaneously as a foundation and a means for molding the world in its own image” and, moreover, as a way to “engage a computational natural world” (Douglas-Jones and Gad 2015, 9). Similarly, for some cyberneticists we *are* computer programs, as are animals and other organisms and, of course, computers and other machines. Some Artificial Intelligence researchers have argued that they have created new life-forms out of digital simulations, which becomes understandable when we consider how code, programs, and computing languages can be understood and experienced as parallel and self-contained worlds which are then seen as reflections or instantiations of the nature of reality itself (Hayles 1999, 225–44; Helmreich 1998, 1-5-68).

In their introduction to *Naturalizing Power: Essays in Feminist Cultural Analysis*, Sylvia Yanagisako and Carol Delaney state that “cultural domains are culturally specific, but they usually come with claims of universality, which are part and parcel of their seeming to be given-in-nature and/or god given” (Yanagisako and Delaney 1995, 12). Computational thinking and computing practices, while not necessarily domains in themselves, in how they stand in for and are taken as part of “nature” and “reality,” and even as religion and the sacred as I discuss below, are taken and enacted as universal and given-in-nature. The ahistoricized encounters students today have with computers, programming languages, and code reinforce this perspective and form of encounter. The challenge for students is to function in these natural computing worlds, perhaps to explore the underlying functionality, but not to interrogate the context of their construction. Computer science education then works as an initiation into these worlds and into the mysteries of rendering technical and rendering natural.

3.3 Initiations

Initiation rituals have long fascinated anthropologists. Such rituals can work as rites of passage through different types of transitions: from childhood to adulthood or from single to married, for example. Students' first programming course and their university computer science education overall can be seen as an extended initiation ritual as part of their transition from being a "normal person" to a "programmer." I will discuss the making of distinctions between programmers and others in Chapters 5 and 6. As will also be discussed further throughout this dissertation, these initiations are not exclusive to the students at Temasek University. Part of the significance of the first-year computer science education at Temasek University is how it fits with computer science education transnationally.

Arnold van Gennep (1960) and Victor Turner (1969, 2008) have each provided detailed elucidations of the different facets and roles of rites of passage. While we should be cautious about transposing theoretical constructs to a significantly different cultural and social context from which they were developed, the insights developed by both van Gennep and Turner offer significant insights into the processes involved in learning computer science. Van Gennep defined rites of passage as "rites which accompany every change of place, state, social position and age" and having three phases: "separation, margin (or *limen*), and aggregation" (Turner 2008, 92). University studies as a whole can be construed as a liminal phase as part of a rite of passage into professional adulthood, from untrained and unspecialized persons into professional computer scientists (Field and Morgan-Klein 2010).

Turner further explores what happens in this marginal or liminal phase of initiation (Turner 2008). In particular, he suggests that a key purpose of these rites is "the communication of the *sacra*" – the transmission and incorporation of cultural and social symbols and practices to the initiates or students (Turner 2008, 96). This significantly includes teaching the main facets of theology, cosmology, and myth that are part of a society, and students' duties for the future:

> The term 'archetype' denotes in Greek a master stamp or impress, and these *sacra*, presented with a numinous simplicity, stamp into the neophytes the basic assumptions of their culture. The neophytes are told also that they are being filled with mystical power by what they see and what they are told about it. According to the purpose of the initiation, this power confers on them capacities to undertake successfully the tasks of their new office, in this world or the next.
>
> Thus, the communication of *sacra* both teaches the neophytes how to think with some degree of abstraction about their cultural milieu and gives them ultimate standards of reference. At the same time, it is believed to change their nature, transform them from one kind of being into another (Turner 2008, 99).

Rites of passage can therefore serve as a period of training in cultural norms, values, and beliefs. In this way, through their studies, computer science students are communicated the *sacra* of computer science. Programs such as UNIX, Vim, and the C compiler, learning syntax and grammar, and attending classes, completing assignments, and writing exams serve the functions played by masks, figurines, effigies, learning the names of deities, and ceremonies in Turner's discussion of initiations (Turner 2008, 96–97). Through their studies students thereby learn symbols and practices, gaining the "mystical power" of programming and computer science abstraction.

Computer programmers and scientists often present themselves as mystical wizards proficient in the power and magic of programming. This perspective was popularly espoused by Steven Levy in his well-known book *Hackers: Heroes of the Computer Revolution* (2010), who referred to and portrayed MIT hackers as wizards but also to the MIT academics and computer administrators as a priesthood with privileged access to the computers. In either case, the programmers were seen to have access to arcane knowledge and near magical powers to operate and control computers. This perspective has proliferated internationally and through various media, with both programming practice being seen as magical, particularly when it is poorly understood, and programmers themselves as having magical powers, as seen in Figure 3-4. Comic strips (such as Figure 3-4), memes, blog posts, and other facets of popular culture circulate and promote this idea. Episodes of *Star Trek* such as "The Apple" and "Return of the Archons," for example, present cultures who have forgotten the functioning of the advanced computers their ancestors had made and were thus treating these computers akin to deities to be worshipped without question. Professors at Temasek University would also sometimes refer to a complicated or difficult to understand algorithm as doing magic: "We do it by magic, that's the power of recursion" one professor commented before going into greater detail about how a sorting algorithm worked. Certain algorithmic techniques are described as "powerful" or certain operations as "dangerous."

Figure 3-4: Programming as magical (CommitStrip 2014)

Nathan Ensmenger also discusses how early programming was seen as a "black art, a private arcane matter… in which the success of a program depended primarily on the programmer's private techniques and inventions," spurring later efforts to professionalize and manage the software industry in more "scientific" ways (John Backus cited in Ensmenger 2010b, 16). Yet, the portrayals of programming as artistic and mysterious was also part of a contest over power between early programmers and managers in the 1950s, in relation to their growing indispensability to the industry and the difficulties of managing their work (Ensmenger 2010b, 48–49). Similarly, contemporary portrayals of hackers of having wizard-like virtuosic skill in computer science and tech industry discourse can also be seen as a means for hackers to promote their own special or "superior" status, which also has gendered implications where, in many cases, the distinct hacker culture or ethic that is promoted within computing is also perceived as male dominated and masculine (Håpnes and Rasmussen 1991; Ensmenger 2015). Processes of rendering technical and rendering natural play a key role in this regard. Through the naturalization of computing worlds as the same as reality, the performative diffractions of computing create a reality where magic is possible and computer scientists are the magicians.

My intention here is not to reproduce these metaphors or portrayals of programmers or hackers as having special powers, but rather to explore how understanding computer science as a process of initiation into seemingly mysterious practices helps illuminate the intra-actions that constitute computer scientists, and computer science as a discipline, and the diffractive effects of these intra-actions. Jan Nespor (1994) argues that learning a discipline is not about acquiring knowledge, but rather about becoming integrated into disciplinary actor-networks. Using Physics as an example, he argues that students learn to mobilize and become tied to networks of textbooks, equations, formulas, and expertise operating beyond the classrooms themselves. Similarly, as they are initiated into the *sacra* of computer science, of programming worlds, students in my research learn to mobilize or intra-act with particular programming languages, program documentation, programs, data structures, algorithms, internet resources, networks of academic and professional expertise, and so on, that work to constitute them as members of the computer science discipline. Students learn several languages over the course of their academic programs, and each of these connects students to different (transnational) networks of programmers and programming practice. These networks constitute various galactic networks or solar systems, clusters of worlds with shared and interrelated technical and social rules that students learn to follow and reproduce.

Nespor's approach to disciplinary actor networks, however, ignores the ways in which these networks or apparatuses become solidified or cut as part of students' experiences and the formation of their subjectivities (Strathern 1996). To Nespor, students and experts are nothing without their networks. He suggests, "drop a student or a physicist or a manager on a deserted island without their tools and colleagues and the questions of what they 'know' and in what sense they've 'learned' are rendered moot" (Nespor 1994, 11). Certainly having disciplinary tools at hand are helpful and programming is difficult without a computer, but computer science initiation works as training in a mode of thought and mode of practice, as discussed above. As Latour suggests, while "every competence, deep down in the silence of your interiority, has first come from the outside" these become "slowly sunk in and deposited into some well-constructed cellar whose doors have then to be carefully sealed" (Latour 2005, 213). Thus, while learning computer science certainly involves mobilizing "outside" actors such as lectures, notes, slides, documentation, and compilers when learning proper syntax, for example, aspects of these networks become internalized. Jean Lave similarly argues, "learning to act on the basis of any craft… requires practice to come to inhabit the practice and its conception of the world" (Lave 2011, 156). In other words, the cuts that constitute computer scietists as persons encompass (rather than exclude) aspects of these "other" actors who students intra-act with as part of computer science education.

The struggles that students experience when first learning to program, as discussed above, usually give way to progressive realizations and successes. Students were told early on that programming takes a great deal of practice, and they were given numerous problems, many of which were not for credit, to complete as part of this practice. About a third of the way through the first-year programming course students were told they should have written fifty to sixty programs already, although it is unlikely most students had in fact completed that number. One third-year student explained this early learning process:

> The way that they taught it was that, apart from the technical aspect of learning the C language itself, the way that you think about solving problems I think was pretty similar as to what you have done in maths before, right. You think in a certain way. You're given a problem and then you learn, you try to figure it out… The difficult part came in when you have to implement it using a language, when you actually have to program it. So, that takes some practice. A lot of practice actually. A lot of debugging, and yeah. So once you get over that then it starts going a bit more fluently (Aindri 2013).

The repetitive practice and work is accompanied by break throughs that can approach epiphanies, and joys at success. Sitting in the programming lab as students were working on their assignment problems, I watched as as one student was repeatedly looking back and forth between the course slides and the program he was working on on his screen. Suddenly, he smacked his head in realization, having figured out what was wrong.

Extended practice punctuated by these moments of realization work to initiate students into the particular thought processes necessary for computer science and programming, working as both a routine part of students' experiences and as a process of discovery and self-realization. Both processes help solidify particular competences and bring "fluency" in the use of particular languages and in programming practices. Over time, students increasingly remember syntax without referring to documentation. They learn to structure programs and write algorithms, to translate reality, and solve problems in particular ways that would continue if they were stuck on a "deserted island without their tools and colleagues." In this way, computational thinking, discussed above, is itself cut from the technologies it is seen as derived from (but not from students and other persons doing the thinking); it is treated as a distinct way of being and approaching the world, whether one is using a computer or not. Students are thereby performatively transformed throughout their initiation from a "normal person" to a "programmer" or "computer scientist."

3.3.1 Possibilities for Thought

After starting this initiation into computer science thinking and practice it becomes remarkably hard to think around. As I joined various computer science classes for my fieldwork observations, I remember feeling that *what* was being taught seemed so normal and natural. I recognized and understood the material from my undergraduate studies, although it had been at least five years since I had done any substantial programming. Thus, while I took fieldnotes that included the content of computer science knowledge, at the time I did not – or possibly could not – think to question it. During observations I was more interested in *how* computer science was being taught and learned. I simultaneously felt as though I had forgotten all of my anthropological theory. It did not seem to make sense in the context of computer science. As mentioned in the preceding chapter, I was continuously asked what I was trying to find out, to answer, what was my hypothesis? My response that I was interested in how gender is involved in computer science felt unsatisfactory, insubstantial. There was no problem to solve, except when my research was frequently reinterpreted as being about the lack of women in computer science, a topic I return to in Chapter 7.

In her article on conversion to Fundamental Baptism, Susan Harding explores "the consequences of listening" (Harding 1987, 168). She argues and shows through her own struggles with ethnographic positioning and belief, that conversion is a process of acquiring a specific religious language and that if a person is willing to listen, then they have begun to convert. They have begun to accept the reality of a believer (Harding 1987, 178). In listening to and accepting the discussions about computer science knowledge and programming languages, I was accepting and believing the reality of computing worlds.

Scholars studying the use of specialized language also discuss how it shapes possibilities for thought – that particular jargons or lexicons allow or prevent a person from saying particular things (Cohn 1987). For Carol Cohn, during her fieldwork with nuclear defence analysts, she found that in listening to and speaking the "technostrategic language" of the defense intellectuals "I could not use the language to express my concerns because it was physically impossible. This language does not allow certain questions to be asked or certain values to be expressed" (Cohn 1987, 708). In the computer science system of belief, reality can be and is rendered technical and framed in terms of problems with technical solutions. Such problems only need to be made explicit, broken down into smaller parts, each with particular, logical, and detailed steps to solve according to the various algorithmic, data structure, and programming language *sacra* that students have learned, along with students' own ingenuity and passion, which I discuss in Chapter 6. These technical solutions are seen as a "natural" part of human practice and thought, distinct from social/cultural practice, as has similarly been discussed in relation to a wide variety of scientific practices (Downey 1998; Haraway 1991a; Keller 1985; Latour 1993). While I was not participating in the same rigorous initiation and testing with assignments and examinations marking my progress, although I had already done this in the past, I was nevertheless being initiated into the mysteries and system of belief of computer science and being transformed "from one kind of being into another" (Turner 2008, 99). For me, computer science discourse was shaping the fields of possibilities to think about my own research as needing questions and problems for which I could find answers and solutions.

While all facets of reality are subject to technical interventions and renderings, these renderings also produce and enact exclusions and diffractions. As a result, as seen in the literature from development studies, the "solutions" developed for "problems" may not have the intended effects (Ferguson 1990; Li 2007). Nevertheless, for computer science student initiates, their induction into programming and computing worlds opens up new possibilities of thought. The "communication of *sacra* [which] both teaches the neophytes how to think with some degree of abstraction about their cultural milieu and gives them ultimate standards of reference" entails a rendering of reality as naturally technical. Later chapters discuss how this initiation and perspective influences students' experiences and their future lives and aspirations, while also considering how they negotiate, contest, and rework the renderings of their subjectivities, and of reality.

3.4 Conclusion

When first learning computer science, students are "pushed off the edge of the cliff," as Susan suggested. They are surrounded by new programming worlds that they must learn to understand and use. Students are also taught particular forms of thought, namely algorithmic problem solving. In extending these modes of thought and modes of practice through accumulated layers of encapsulation and abstraction, students are taught how to model and program in relation to both programming worlds and to reality. The human and nonhuman networks that are mobilized in the process, built both historically and as part of ongoing practices, are cut as these are turned into immutable mobiles, rendering code, programs, and even auto-markers as autonomous and worldly actors. These are ongoing processes of diffraction that render technical both computing worlds and reality, which are also sometimes rendered natural where computational thought is interpreted as originating in the evolution of human thought and digital computation part of processes of the "natural" world, or inherent to the whole universe's makeup.

This process can be seen as an initiation into the networks and *sacra* of computer science thought and practice. As Turner discusses of the liminal period entailed in such initiation rituals:

> The first is the reduction of culture into recognized components or factors; the second is their recombination in fantastic or monstrous patterns and shapes; and the third is their recombination in ways that make sense with regard to the new state and status that the neophytes will enter" (Turner 2008, 98).

Students learn new worlds of programming language syntax, of compilers, of technological environments such as UNIX and Vim, and the modelling potential of UML. They learn how to recognize and translate the technical operation of computers and the social, cultural, political, and historical operations of reality as "fantastic or monstrous patterns and shapes" in these new languages and technological systems. They recombine these patterns and shapes to create new programming worlds, re-constituting the possibilities for thought of the new computer-scientists-in-the-making in the process. The next chapter explores how these computing worlds are gendered, showing the ways that heteronormative gender binaries are (re)produced as part of computer science knowledge, thought, and practice.

Chapter 4 : (Trans)National Heteronormativity

Most of my memories from my undergraduate studies are a blur of continual weekly assignments – in algebra, calculus, *and* programming – in the early years, and then multiple all-nighters to meet project deadlines in later years. After first year I stopped buying most of the textbooks for my math and computer science courses, since most of the information we needed was given in slides or course-packs, or available online. Yet, one textbook stands out in my memory. Known colloquially as "CLRS" for its authors Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, it is simply titled "Introduction to Algorithms" (2009[1989]). A massive 3 to 4-inch thick heavy hard-cover tome, it was one of the most useful and memorable texts throughout my studies in computer science. Perhaps I should not have been surprised then, although I was, when I saw that the textbook was being used and referred to extensively in courses I observed in Singapore. During my research I saw students carrying around the only marginally smaller and lighter soft-cover version, hauling it with them to their open-book exams in the hope that the book's numerous pages would offer up the secrets to algorithmic problems. It seemed to be the default reference for learning, studying, and teaching algorithms in Singapore as much as in Canada, where I studied computer science ten years prior.

This chapter explores the movement and social construction of computer science knowledge and practice, using the lens of gender norms. I consider how heteronormative binaries were assumed and enacted among students and professors in computer science. I also show how these norms connect in many ways to gender discourse and policy in

Singapore. Many computer science discourses, however, extend beyond the national borders of Singapore and are closely tied to those of the international tech industry, as well as the discipline of computer science. I consider the ways gender norms are constituted in computer science knowledge and practice, which are transnationally mobile in the form of textbooks, algorithms, and theorems, for example.

While performances of heteronormativity are the focus of this chapter, there are certainly many other values and practices through which computer science knowledge is significantly constructed and understood. As one professor pointed out to me about students' story-writing projects, which I discuss below, the student-teacher relationship featured prominently. In these cases, knowledge such as mathematical theorems or computer algorithms were treated as entities – immutable mobiles – that can be transmitted from one person (the teacher) to another (the student). Economics, measurement and geometry, and mobility are all themes that also feature prominently in examples and problems through which professors teach and students learn computer science. In this regard, the intra-actions of gender and computing are one story – one facet – which I have chosen to focus on, among many that I could have explored.

I am also not implying that those in computer science should seek to "purify" all references to gender in their teaching or in their definitions of problems and theorems. A great deal of research explores the ways computers and technologies are "scripted" to accommodate certain values, behaviours, uses, and users (Akrich 1992; Berg 1999; Bray 2007; Forsythe 2001; Suchman 2002; van Oost 2003; Winner 1980). Rather, as discussed in the previous chapter, computer science – from the construction of computers themselves, to the design of programming languages, along with the design and analysis of algorithms and data structures – all constitute "wicked problems" in that "they rely upon elusive political judgment for resolution. (Not 'solution.' Social problems are never solved. At best they are only re-solved – over and over again.)" (Rittel and Webber 1973, 160). That is, the performance of gender in computer science, and the doing of computer science more generally, is inseparable from the historical, social, and political relations of which it is part.

In this chapter, I bring to the forefront the selective traditions of particular gender norms that are portrayed and performed in computer science at Temasek University, and more generally, through intra-actions constituted by teaching and movements of experts, textbooks, code, programs, and other actors (Williams 1977, 115-120). In considering the relationship between humans and robots in Artificial Intelligence research, Lucy Suchman, drawing on Karen Barad's (Barad 2007) concept of entanglements, explores how they "become with" one another (Suchman 2007b, 285, 2011a). Similarly, students and computing technologies are produced through their mutual intra-actions; these are continually producing boundaries in terms of their relationship with technologies and computing worlds and with gender, in ways shaped by historical context and bringing about social and material consequences. I explore the selective meanings and boundaries that are performatively (re)drawn among particular intra-active assemblages or phenomena in terms of gender. Yet, I also consider how these boundaries and norms are connected with national and transnational discourses and practices.

4.1 Heteronormativity and Computer Science at Temasek University

Many I spoke with in computer science in Singapore did not consider gender to be an issue or related to computer science at all, although I explore in Chapter 7 the ways gender is explicitly addressed in relation to the issue of the number of women in computing. As discussed in the preceding chapter, however, computer science practices of rendering technical work to separate technical and social frames. In this context, gender is generally placed squarely on the "social" side, separate from the nature of technical knowledge. Despite these assertions and separations, however, performatives of a heteronormative gender binary were common in terms of language usage, and teaching practices and examples, and often went unmarked. I also discuss below how these performatives have become intertwined with the construction and understanding of computer science knowledge. In many ways the construction of knowledge cannot be separated from the way it is taught or the use of language, but I make this distinction to highlight practices that may be specific to individual professors or tied up with gender norms in Singapore, compared to gendered assumptions that circulate transnationally through definitions of standard data structures and algorithms, for example.

4.1.1 Language Choice

There were several subtle ways that professors tended to assume and reproduce gender binaries when teaching. The most basic of these is seen in one professor's habit of beginning every class with the phrase "good morning boys and girls." This simple phrase, so often repeated, I have no doubt was meant to be inclusive and friendly. Yet, it nonetheless works to assert an immediate distinction between "boys" and "girls" and to erase other gendered possibilities.

When speaking about technical actors, such as pieces of code, an algorithm, or a data structure, or about students in a general sense, professors would most often use a gender-neutral pronoun: "we," "us," "this," "that," "you," or "it." In three different courses, each with different professors, I counted the use of different pronouns over a 10-minute period, and in all cases no gendered pronouns were used. Nevertheless, there were infrequent but regular instances when professors would start discussing the actions of code or algorithms, refer to elements of a data structure, refer to a generic programmer or user, or to students in general, with a gendered pronoun or gendered reference, generally "he," "him," or "guys." For example, in answering a student's question about whether two variables ("*a" and "a") could be declared separately, the professor answered that you cannot because the two "guys" are the same. In another class, when discussing the different tools for testing available in the programming language Java, the professor commented that C++ "guys" may feel left out in the discussion. In yet another class, the professor was instructing students about doing user-research and explained they should "empathize with the user to understand *him*" (emphasis added). This happened among professors of different genders and varying national backgrounds.

Perhaps most illustrative is, on the day I presented myself and my research to the class, one of these professors was discussing different levels of competencies and responsibilities in programming with the highest level being a program "architect." In discussing the architect position and what "he" does, the professor then paused for a moment and looked up at me sitting at the back of the lecture hall and asked whether I was taking observations on whether the professor says “he” every time in the lecture and then commented that he means “he and she” when he says it. The next time, the professor then made a point to say “he and she” when talking about the generic architect position. A similar occurrence happened in a different class with a different professor. There are two things happening in these examples: first, a masculine/male figure and pronoun is taken as the unmarked representative of technologies such as code and program elements, and of programmers and users. Second, when the professor realizes and acknowledges this language usage, they address it by saying “he and she” which includes women but maintains a binary distinction. This is partially a feature of the English language, one which I myself repeat sometimes. Yet, there are also numerous ways that these statements could have been and are made, such as “good morning, everyone,” that would avoid the gendered opposition altogether.

4.1.2 Teaching Examples

Assumptions about gender, along with gendered stereotypes, were also present in the various examples and stories that professors used for teaching. This practice varied among professors, with some using many examples that related to different facets of society while others discussed computer science knowledge with limited or no recourse to overt social references. These references also depend on the level and branch of computer science being discussed; it is much easier to avoid social references when considering the technical facets of algorithms and data structures – their operation and efficiency, for example – than when discussing user interfaces or social networks.[38] At the same time, I argue that technical and social are intertwined in all branches of computer science, and such gendered examples become part and parcel of understanding multiple facets of computing worlds, just as they are a more acknowledged and recognized part of human-computer interaction and related fields.

Most simply and most commonly, gender/sex was used as an aspect of social reality to be drawn upon as a resource for teaching examples. Figure 4-1 shows a programming problem given to students to teach them about writing functions. The ultimate goal was to draw a rocket ship, a female stick figure, and a male stick figure. Students were supposed to break this task down into drawing circles, triangles, and "legs" (drawn as a triangle without a base); they were thus learning to factor code into functions that could be reused rather than rewriting the code multiple times for each piece. Here, a distinction between male and female is taken as part of "objective" reality, something that can be named and depicted just like a rocket ship.

[38] The focus on efficiency is itself related to social and cultural values that prioritize increasing the speed and decreasing the size of technologies, along with valuing efficiency and productivity more generally in relation to labour and economic value. These values, in turn, have gendered implications (e.g. Yeoh, Huang, and Willis 2000). As considered throughout this dissertation, all computer science "problems" are inherently saturated with social and cultural values and norms. I am focusing my analysis here, however, on explicitly gendered references.

Figure 4-1: Gendered problem used to teach functions

In a later problem related to manipulating arrays (a particular type of data structure), students were asked to design a program to select a "compatible team-mate," defined as a person who is the same age but "opposite sex" as a player inputted by the user. In another problem for practising input and output, set by a different professor, students were asked to write a program that takes as input the age and gender of a user, and based on the inputted gender output either "Hello [age] dude!" or "Hey [age] girl!". The only valid input here was then either 'm' or 'f'. Yet another problem was framed around heteronormative marriage:

> Michelle and Gary are suffering from cold feet just before their wedding. They wanted to confirm if they are really suitable for each other. Michelle remembered a game she played when she was young. The game is as follows: strike off common letters found in both names (each letter can only be struck off once). If both names have an even number of letters remaining or both names have an odd number of letters remaining, then they are said to be compatible.

The game works such that any names could be input, allowing for the possibility of many different kinds of non-heteronormative pairings, but the test examples given include "Michelle April Tan" with "Gary Anand Tham" and "Tarzan" with "Jane." These examples do not tie heteronormative gender binaries directly to programming concepts but do work to naturalize and reproduce the normalcy of such gendered assumptions.

4.1.3 Computing and Mathematical Concepts

In other cases, gender distinctions were taken as directly illustrative of mathematical or programming concepts. "All of the girls go on the right side of the room, all the guys go on the left side of the room, nobody is both a boy and a girl" one professor exclaimed as an example of a "partition," with the assumption that no person was both a girl or a guy and that every person would undoubtedly fit in one of the two categories. Similarly, in discussing the idea of an exclusive "or," where something can be one or the other but not both, the same professor told a story that he had heard from his teacher, about the logician Kurt Gödel who went to visit his grad student, who was pregnant in the hospital. He asks her if it is a boy or girl. The student answers "Yes." The students in class laughed, and the professor explained how Gödel gives her a PhD because she is clearly a good logician. The point of the story is that the baby is either a boy or a girl, and assumed that it cannot be both or neither. Thus, the student's answer is clever and has demonstrated her grasp of the "exclusive or" concept. An equivalent example given for the concept is ordering tea *or* coffee at a restaurant.

This professor, in particular, often used heteronormative gender as a way of explaining mathematical and computing concepts. He was also aware of and reflexive about his usage of these examples. When discussing gender and computing in an interview, he commented to me:

> I do sexist jokes sometimes hoping nobody is going to be offended and complain. And also I use old-school examples that were very normal when I was a student. Then all of a sudden there are people like you sitting in the theatre I'm wondering should I have said that or not.

At the same time, such gendered examples were not unique to this professor's class. In Chapter 3, I showed the example of a (heterosexual) marriage relationship – depicting "Man," "Woman," and "Marriage" objects – modelled to students to teach them about UML diagrams. In another class, as an illustration of a "Flow Model" showing the flow of work and communication among different actors, the actors included "Husband," "Mom," and "Daughter" and focused on the activity of grocery shopping. The example continues through to other models such as the "Sequence Model" which includes the sequence: the Mom "invites husband to go along [shopping] as a shared activity" and the Husband "had 3 other things planned and is overwhelmed" then an argument ensues. These were again provided by different professors, who were from different ethnic and national backgrounds, were educated in a different countries, and had varied interests in computer science research and education.

Students (re)produced similar normative perspectives on gender and sexuality when doing computer science. In one class, students were asked to develop stories around particular classical mathematical and computer science problems or theorems that they first needed to research and understand. In a variety of cases, the stories students told similarly relied on and portrayed heteronormative binary genders. In one example that seems particularly illustrative of issues of national concern in Singapore, the government (albeit the US government figures in this story) notices a problem: people are working all day and then going home to sleep and then repeating the cycle; they have no time to date, and marriage and birth rates are falling, causing a "crisis in the nation." The government works to implement a national match-making programme to achieve "stable marriages" for (heterosexual) couples. The story proceeds to describe the Gale-Shapley algorithm that solves the "Stable Marriage Problem," a well-known computer science problem, as part of this match-making programme.

I discuss below the ways the stable marriage problem, in particular, works to reproduce heteronormativity, not least through its naming. Yet, this was not the only mathematical problem where students drew on norms of gender and sexuality as part of their stories. Another tells of "Alice and Bob" who like to write "mushy" love letters to one another, but use public key cryptography to ensure that the "postman" does not read their letters. I also discuss this story further below in discussing how the retelling of this problem works as a tradition in computer science. For the purposes of this section, it is clear that students, like their teachers, draw on and assume binary genders and heterosexual partnerships/marriages as ways of understanding and explaining computer science knowledge.

4.1.4 Discourse in the Department

Heteronormativity, including heterosexuality, were also assumed in general discourse throughout the computer science department. The department's Facebook page demonstrates this clearly. For a series of posts, for example, they used a simple repeated format: they have an image with two sides, one labelled "programmers" and the other labelled “non programmers.” Occasionally these divisions are alternatively labelled “programmer” and “normal person.” The meaning of these labels is further elaborated through a title stating the topic of comparison, and then an image for each category illustrating the distinction. One of the more meaningful distinctions is made under the heading “Commit” which shows a representational depiction of a heterosexual couple holding hands with a heart between them, but for programmers shows “SQL>Commit,” which is the command to commit items to a database in the language SQL.

This image does place programmers outside of the interest for heterosexual intimacy, as they are separated from non programmers or “normal people.” I explore in Chapter 6 how some in computing and particularly in relation to hacking have argued that things like “sex, money, and social approval” are mere distractions that hackers and “good” computer scientists should avoid (Raymond 2001). While some in the student hacker group broadly support a passionate dedication to computing to the exclusion of many other facets of social life, in general the students and professors that I met and spoke with pursued what would be broadly construed as normative lives. As one student explained: “The normal outlook for people: [they] will get a job, get married, get a nice house. As in live a good life, lah, like what we have now with our families. That is a common view for everyone. I think I’m following that common view for now” (Ariff 2014). I discuss gender norms in wider Singapore society further below. I focus here on the ways gender is constructed within the department of computer science.

This participation in and performance of heteronormativity is seen in other posts on the department’s Facebook page. An image posted wishing “Happy Valentine’s Day,” shown in Figure 4-2, for example, depicts a man (illustrated by his short cut hair, blue coloured shirt, and lack of any depictions of makeup, broad chest) holding a video game controller and asking a woman (illustrated by her below shoulder-length blue hair, pink coloured shirt, skirt, mouth drawn to depict red lipstick and eyes drawn to suggest eyeshadow, and breasts), “Will you be my player 2?” Other posts related to the release of “#AndroidLollipop” and the night of the “#BloodMoon” show the same pairing. Notice the ways that men and women are distinctly depicted in this image with differences in hair, dress, color, makeup, and physical morphology to emphasize a binary separation, along with particular depictions of masculinity and femininity, in this heterosexual relationship.

Figure 4-2: Image from CS Department's Facebook page (edited for anonymity)

Yet, perhaps one of the more naturalized ways that binary genders are assumed and normalized is through the use of statistics. Statistics were often difficult to access in

Singapore where information about different groups, such as enrollment by nationality and ethnicity, are contentious topics. Yet, Temasek University does provide broad enrollment statistics by faculty, available online extending back to 1994. These are very general statistics, but they provide one subdivision: “Male” and “Female.” This division is thus taken as basic, and seemingly apolitical, at least compared to issues such as the number of university spaces being given to “foreign talents” versus local students, for example. Most professors I spoke with said they paid little to no attention to gender in their teaching or course planning, or even more generally in curriculum planning and department administration. At the same time, these statistics are nonetheless repeatedly produced and thereby reproducing a seemingly natural division between men and women in computer science and in the university. In facets of computer science research, such as user research, these naturalized binary divisions are again reproduced and used to statistically analyze “user” behaviour. In this way, a gender binary is taken as a resource or natural division for illustrating programming concepts, creating example problems for students to solve, or analyzing user behaviour. I return to the significance of statistics in relation to gender in Chapter 7 when I discuss the transnational focus on women in computing.

Gender is thus an implicit part of the computer science curriculum, as well as part of understanding and doing computer science. I explore how the reproduction of a heteronormative framework is tied to Singaporean state policies, as well as to transnational computer science knowledge and practice. However, these actors and structures do not determine the performance of gender among students and professors in computer science, as will be explored in Chapter 8. Additionally, the influence of the state and the computer science discipline cannot be reduced to one another.

In the case of heteronormativity, discourses and practices of the Singaporean state and the transnational computer science discipline dovetail and intra-act to powerfully summon those in computer science to fit within and enact heteronormativity (Foucault 1982; Mahmood 2005). That is, they operate through what Eric Wolf, drawing on Michel Foucault (1982), terms “structural power,” which “shapes the social field of action so as to render some kinds of behaviour possible, while making others less possible or impossible” (Wolf 1990, 587). In this case, doing gender in non-heteronormative ways is not rendered impossible, but heteronormativity is rendered as part of the socio-technical field of action such that other ways of thinking about and doing both computer science and gender are indeed made less possible. Yet, the intra-actions of national and transnational discourses have also entailed conflicts in relation to homosexuality and LGBTQ rights that point to the potential to destabilize the mutually supportive reproduction of heteronormativity.

4.2 Heteronormativity in Singapore

Heteronormative performances in computer science relate in part to gender norms in Singapore more broadly. In particular, government rule, regulation, and policy, and popular discourse in Singapore rely on assumptions of heteronormative gender binaries, extending through a variety of different areas, including laws and policies on reproduction, marriage, and government-run housing, along with various facets of national education, which insist on particular gender, marriage, and kinship relations among the nation-state's citizen-subjects. Underlying many of the government's policies is a valuation of heteronormative family units as "the basic unit of society" as outlined by the Ministry of Education in relation to sexuality education (MOE 2016).

In Singapore, homosexual acts between men remain illegal under Section 377a of the Penal Code under "outrages on decency," which dates back to British colonial rule in Singapore. Conviction under this law entails imprisonment up to two years for "any male person who, in public or private, commits, or abets the commission of, or procures or attempts to procure the commission by any male person of, any act of gross indecency with another male person" ("Outrages on Decency" 2008). While, in practice, the law is no longer enforced and movements such as "Pink Dot" (a group and annual celebration that "support the belief that everyone deserves the freedom to love") are growing, gay and lesbian Singaporeans continue to feel marginalized as Singaporean citizens and as part of Singapore society (Pink Dot sg 2016; R. F. Phillips 2008, 4–8).

Recent attempts to have Section 377a repealed have also failed since, according to the government, "segments of Singapore society continue to hold strong views against homosexuality for various reasons including religious convictions and moral values" (Chan 2015, 21). These "strong views" have certainly been visible in a variety of ways in Singapore, including the "Wear White" campaign each year to "promote traditional family values" in response to Pink Dot (Lee 2016), or the National Library Board restricting access to certain children's books following a complaint, including a book entitled "Tango Makes Three" based on a true story that featured two male penguins raising a young penguin (Chua 2014). Several scholars have argued that the illegality of homosexuality is also tied to the state's focus on reproducing and creating a competitive and talented population seen as essential to the survival of the nation (Lyons 2004; Peletz 2007; Yao 2006). At the same time, a variety of research has explored the ways gay and lesbian communities persevere and persist in Singapore within this restrictive legal and social context (C. K. K. Tan 2009, 2011; K. P. Tan and Jin 2007; Phillips 2008, 2013; Yue 2007).

Yet, if homosexuality in Singapore has gained some measure of tolerance, transgender persons remain largely invisible and face significant discrimination and marginalization. Singaporean surgeons performed Asia's first gender confirmation surgery in 1971, completing a total of 413 of such surgeries until 1990 (Ho, Sherqueshaa, and Zheng 2016, 54). In 1973 the government also allowed post-operative transgender people to change their sex on their National Registration Identity Card (NRIC), a person's form of identification in Singapore, and in 1996 the Women's Charter was amended to allow heterosexual marriage by a post-operative transgender person (Ho, Sherqueshaa, and Zheng 2016, 55). At the same time, the Women's Charter prevents marriage between persons of the same sex ("Avoidance of marriage between persons of the same sex" 2008).[39] Additionally, public hospitals in Singapore no longer provide gender-confirmation operations, with only one surgeon offering the procedure privately (Hoe 2014). Many transgender and gender non-conforming people in Singapore cannot or do not want to undergo surgery, creating a mismatch in their identities and legal

[39] The Women's Charter was first passed in 1961 in Singapore and was meant to protect the rights of women, focusing on rights and protections in relation to marriage, family, divorce, and "offenses against women and girls" including prostitution, trafficking, and brothels ("Women's Charter" 2008).

documents, and rendering them invisible and often marginalized and discriminated against across multiple facets of society, including experiences of violence and job discrimination (Ho, Sherqueshaa, and Zheng 2016, 55; Mosbergen 2015; Sherry 2015).

There are also a variety of more subtle ways that government laws and policies insist on and promote heteronormativity, including heterosexuality, simultaneously reproducing gender norms and contributing to the continued marginalization of LGBTQ persons. Rules of the Housing Development Board (HDB) in Singapore, which housed 82.4% of the city-state's population in 2010, only provide housing to (heterosexual) married men and women, unless they are over the age of 35 (Singapore Department of Statistics 2011, xi). This age was reduced from 50 years for men and 40 years for women, at which point it is assumed the person would never marry (Huat 1997, 323–24). Housing policies are very explicit about the relationship between policy and discourses supporting the family as the basic unit of society. The "HDB Fiancé/Fiancée" scheme, for example, allows two unmarried individuals to apply for a new HDB flat and gives them three months to officially marry after they take ownership (Teo 2010, 344). As Michael Hill and Lian Kwen Fee note, these policies are particularly effective because the HDB provides the most affordable housing options in Singapore. Those who do not meet the requirements must turn to the private housing market which is limited and expensive (Hill and Lian 1995, 122–23). As suggested by Ariff above, to "get a nice house," which often means an HDB flat, is also seen as a normal – and even natural – life-step for Singaporeans (see also Teo 2010, 344–46).

These housing policies work in conjunction with government pro-natalist policies (Heng and Devan 1995; Ong 2005, 343–44; Sun 2012; Yao 2006, 98–117). A recent parliamentary discussion about the state not recognizing children of unwed mothers as legitimate is particularly telling about the governance of reproduction, family, gender, and sexuality in Singapore:

> The family is the basic building block of our society. Strong [heterosexual] marriages are the key to strong families, and parenthood within marriage is the desired and prevailing social norm. Hence, benefits such as the Baby Bonus cash gift, housing benefits and tax reliefs are provided to families with legitimate children, to encourage births within marriages (Tan 2016).

As mentioned in this response, (heterosexual) marriage, and reproduction within that framework, is rewarded through a variety of incentives such as baby bonuses and priority access to housing. Yet, those who fall outside those norms, including single, divorced, and homosexual parents, among others, have little access to such privileges, and sometimes even to necessities such as affordable housing.

These policies are tied to the government's ongoing effort to increase the nation's population, which dropped below replacement level after 1975. Prior to the 1980s the government promoted a "Stop at Two" children campaign, alongside liberal abortion policies and incentives for sterilization, particularly for lower-class women with limited formal schooling (Heng and Devan 1995; Sun 2012). Yet, this approach was reversed when population decline started to be seen as a national problem threatening Singapore's future potential "talent" resource pool (Ong 2005). Early programs favoured marriage between university graduates in order to produce "genetically superior offspring" (Heng and Devan 1995: 197). These programs were later abandoned and the government now promotes procreation among all its citizens, although numerous policies continue to implicitly favour marriage among higher income couples (Sun 2012). Policy changes and promotional campaigns to promote procreation since the 1980s include: "Have three or more, if you can afford it," launched in 1987 (John 1987); a "Strong and Stable Families" campaign in 2000; a Marriage and Parenthood package, including the Children Development Co-Savings Scheme (also known as the Baby Bonus scheme) and the Third Child Paid Maternity Leave Scheme in 2001 – the package was revised in 2004, 2008, 2013, and 2015; the "Romancing Singapore" campaign in 2003; and "Singapore, A Great Place for Families" in 2004 (see Sun 2012, 30–32).

The focus on heteronormative families is also tied to national discourses of "Asian Values" that were promoted by the Singaporean government, among others.[40] In Singapore, for example, these are a specific set of five Asian/Singaporean "Shared Values" outlined by the government in a 1991 "Command Paper." These values are meant to operate as a "cultural ballast" against the alienation and individualization of capitalism as it has been experienced in "the West," as well as the onslaught of Western values through globalization (Chia 2011b, 112–13). They are also tied to policies on learning mother-tongues and the maintenance of racial categories in Singapore. These values include "nation before community and society above self" and "family as the basic unit of society," along with emphasizing community support, consensus, and racial and

[40] The term "Asian Values" has been widely adopted by researchers to discuss the promotion of Confucianist, or more broadly "Eastern" ideology and approach to governance that was seen as distinct from "the West", promoted by states such as Singapore, primarily in the 1990s (Ambikaipaker 2015; Ang and Stratton 1995; Chia 2011a). However, the concept and term was also used by the states and policy makers themselves (Koh 2008; Tan 2016).

religious harmony (Yao 2006, 21). These values are considered to be distilled from "eastern" religious and traditional values from Singapore's Chinese, Malaysian, and Indian "races" (Huat 1998, 40). Heteronormativity is produced as a moral value as the (heteronormative) family is thus framed as the basic unit of society and of the nation, which should be prioritized over the self and the community.

In their opposition to "Western" individualism and liberalism, the highly distilled and non-specific "shared values" are also used to support state repression. Lenore Lyons, for example, details how Singapore's only official feminist organization AWARE refuses to officially adopt the label of "feminist," despite its wide use by members and its inclusion in internal policy documents, due to fears of its association with "Western feminism" characterized by free love and immorality (Lyons 2004, 63–69). Members fear that the organization will be shut down or, worse, that some may be arrested imprisoned, and so members police their own behaviour and maintain an unchallenging and moderate approach (Lyons 2004, 138–61). These fears are not unfounded or unprecedented. Members of AWARE have previously been arrested and, more generally, dissidents, critics, and opponents to the Singapore government have historically faced administrative roadblocks, been sued, or arrested (Chia 2011a; Lim 2015; Lyons 2004; Rodan 2003; Sim 2015).

Recent comments by government officials, however, suggest that the strong linkage between marriage, housing, and procreation, often seen as integral to Singaporean life, is shifting (Teo 2010). Senior Minister of State Josephine Teo incited a media discussion in October 2016 when she suggested that "you need a very small space to have sex” in discussing the relation to the “Parenthood Priority Scheme” for giving first-time married couples priority for an HDB flat if they are expecting or have a child below the age of 16 (Tai 2016). She thereby suggested that couples should not wait for a flat, or even marriage necessarily, to start having children. Nevertheless, policies continue to focus on promoting procreation between heterosexual and married couples to produce a “normal” and ideal family household.

While some students I spoke with were broadly aware of policies such as baby bonuses and priority schemes for HDB flats, few felt it affected their lives in an immediate sense. One student commented to me, when I asked whether he read news or paid attention to government policies: “Nope. I try to lah. But in the end I focus on code all day, I completely forget about policies also. Cause in the end I start to do so much about programming [and] I really forget everything else, or that’s just me likely.” This students’ responses also point to the dedication in terms of time and focus that computer science demands, as discussed in Chapter 6, often to the exclusion of other interests. Most students I spoke with, even those in their senior year, were not yet preparing for marriage or a family and so, while some students had a general knowledge of different policies, for many the relevance of these policies for their lives seemed distant. Students were more immediately concerned with their studies and news from the tech industry.

Nevertheless, related policies do have some direct effects on students. Until January 2016, for example, there was government funding up to a maximum of $5000 for student groups and activities, such as orientation camps, that “maximise opportunities for undergraduates to meet and interact with the opposite gender” as part of the

“FamilyMatters@School” program. These activities were also required to have a gender ratio difference of at most 60:40 among participants. A student organizer of orientation camps suggested to me that getting the funding was not that difficult and the possible activities were “limited by our own creativity… as long as you have an activity that encourages interaction, like just talking or social dinner, and then I think you’ll be fine, lah,” although he was not directly involved in applying for that funding.

In July 2016, however, the orientation camps at the National University of Singapore were the centre of a scandal after reports of various sexualized activities being organized and run.[41] Government funding for these programs under the FamilyMatters@School program had already ended, and such activities were explicitly disavowed in the program guidelines, which state:

> MSF [Ministry of Social and Family Development] will not be held responsible for any inappropriate physical contact (e.g. stacking on top of each other, performing push-ups or rolling over each other, sitting on each other’s laps or shoulders, removing items from each other’s body using the mouth, piggybacking, mouth-to-mouth passing of food) that may occur during the events or activities (MSF n.d., 11).

The specificity of the guidelines, however, suggest that the ministry was very aware that such activities often occur in orientation camps. The funding and the activities were premised on the same values as pro-natalist government policies, promoting interaction among students of the “opposite gender” with the ultimate goal of promoting marriage and procreation among students. However, following the media report, all student-

[41] The activities including the re-enactment of a rape scene between a brother and a sister, and men doing push-ups on top of women (Sun 2016).

organized orientation activities were suspended and 30 students were ultimately disciplined (Today Online 2016; Cheng and Ming 2016).

Shamus Khan describes a similar incident at the Barclay dorm of St. Paul's, a private elite school in the US, where disciplinary action was taken after a hazing ritual on "newb night" was seen as overly sexualized and demeaning (Khan 2011, 128–35). However, Khan suggests that:

> What the hazing did was inscribe on the bodies of girls their position within the school… St. Paul's, after all, is built upon ritual. The school's abundance of such events – though lacking body paint or diapers – give shape to the school day, the school year, and, indeed, each student's entire trajectory at the school… What the Barclay girls 'got wrong' was that their acts of hazing were too extreme; they constructed too great a chasm between those at the top and those at the bottom (Khan 2011, 132,134,135).

In the case of the Singapore orientation week, the rituals or activities similarly inscribed heteronormative gender differences on the bodies of both the men and women as explicitly required by the government funding and as a part of everyday interactions at the University and in Singapore, as discussed above. They did so, however, outside the structural life-pathways seen as "normal" or acceptable to Singaporeans, with similar extremes that highlight sex and gender inequalities that are often masked by discourses of meritocracy and equality, as I discuss below and in Chapter 7.

It should thus be clear that heteronormative gender binaries are pervasive in Singapore. While these binaries are treated as "natural," they are also actively promoted by the Singaporean government. Professors' and students' uses of these same assumptions in computer science teaching and practice reinforces this gendered framework, further rendering natural this gendered national context. In a few cases this is a deliberate choice on the part of professors who consciously work to follow government policy and discourse. Additionally, a key facet of Singaporean nation building and development has centred on participating in and fostering global flows of capital, persons, resources, and information. These transnational flows are also significant to the construction of computer science knowledge and of heteronormativity in Singapore.

4.3 Transnational Computing

In studying software developers in Rio de Janeiro, Brazil, Yuri Takhteyev explores how software work operates in and through "worlds of practice," referring, that is, to "systems of activities comprised of people, ideas, and material objects, linked (and defined) simultaneously by shared meanings and joint actions" (Takhteyev 2012, 21). As discussed in Chapter 1, Singapore's focus on developing its national economy has fostered multiple connections with international institutions and scientific and technological communities of practice. I discuss here the ways undergraduate computer science education in Singapore, in particular, is part of such worlds of practice and, specifically, how it has been shaped by norms of transnational computer science discipline. I provide a brief historical overview of the computer science department at Temasek University and its connection to transnational expertise and technologies, and international organizations, as well as some background on those (largely US based) organizations. This is a necessarily brief overview and more research is needed into the detailed historical processes through which computer science in Singapore, among many other places, have become connected to these worlds of practice.

Nevertheless, based on these current and historical connections, I explore how the use of heteronormative gender binaries discussed above draws on discourses and practices originating from beyond the borders of the small city-state and, more specifically, tied to genderings that circulate throughout the discipline but often originate from the US or Western Europe. As discussed in Chapter 3, computer science hardware, software, theoretical knowledge, curricula, textbooks, are taken as mobile – able to travel across times and places – partially because details of their historical and social context are purified and cut and are not often a part of learning computer science. While the Singaporean state powerfully enacts its policies, practices in the computer science discipline intra-act with and are intimately tied to the interests and practices of multinational corporations, professional organizations such as the Association for Computing Machinery, and international institutions. I consider how these *trans*national intra-actions constitute computer science education, as well as the particularities of transnational movements – the directional frictions – across spaces and boundaries.

4.3.1 Histories and Geographies of Computer Science in Singapore

From its inception, the computer science program at Temasek University – in addition to computing and information technology in Singapore more generally – was tied to flows of technologies and expertise elsewhere. The predecessor to the current computer science program was started in 1975 with five professors, all of whom received their bachelor's degrees in Singapore, but trained for their master's and doctoral degrees elsewhere. Three earned their PhDs in Canada and two in the US.[42] This prominence of international training parallels practices in many "peripheral" and postcolonial places for developing and growing academic programs and local expertise.[43] The program at the time focused on training in particular languages (COBOL, FORTRAN, and Assembly), along with skills such as information and data processing and particular forms of computer applications.

As with expertise in Singapore, the technologies and languages being learned and used similarly originated outside of Singapore, primarily in the US. COBOL was a programming language developed in the United States in the late 1950s and early 1960s and geared towards creating a common language for business applications as a joint effort of a variety of companies and government bodies, including IBM, RCA, and the US Department of Defense and National Bureau of Standards (Sammet 1981, 200–216). FORTRAN was developed by IBM and released in 1957, primarily for scientific and mathematical calculations (Ensmenger 2010b, 90–91; see also Backus, Lee, and Ryckman 1981). In 1975 the computer centre that was part of the department and university was equipped with two IBM computers (an IBM-1130 and IBM System 3), as well as a Facommat II computer by Fujitsu in Japan. They also had several automatic and hand punch-machines.

More generally, a report in *The Straits Times* in 1979 on the growth of the local market for computers suggested that 55% of computers and peripherals in Singapore

[42] Scott Campbell explores the connections of the computing profession in Canada to the United States (Campbell 2017).
[43] While training of technical expertise for Singapore has largely centred on the US, the political elite in Singapore have historically been trained in the UK (Ye and Nylander 2015).

were manufactured in the US (The Straits Times 1979). Additionally, US-branded computers, which included those built outside the US as part of multi-national companies, accounted for 82% of computers installed in Singapore, with IBM as the primary vendor (Straits Times 1979). Other suppliers came primarily from the UK and Japan, including companies such as ICL (UK) and Fujitsu, NEC, and Hitachi (Japan). The article suggested that "US vendors have succeeded in Singapore because American products are held in high regard by Asian business men, who consider the US industry to be the pioneer in the computing field" (Straits Times 1979).

Early educational institutions dedicated to training professionals were developed in association with foreign companies, experts, and institutions. The Institute of Systems Science at the National University of Singapore, for example, responsible for training local professionals in information technology and later conducting applied research, was developed in partnership with IBM (Tan and Yeo 1981). The National Computer Board, which was in charge of early technological development efforts, modelled various practices such as the Professional Examinations Syndicate – meant to ensure professional standards – after those of foreign organizations in Britain, the United States, and Japan.

The influence of US-based or US-centred technologies, expertise, companies, and institutions continues to the present. The computer science program at Temasek University underwent a significant overhaul in 1984 to resemble much more closely the curriculum taught today, with a focus on data structures, algorithms, and mathematics. The faculty had also significantly expanded to sixteen persons, likely tied to the government's efforts to develop the national IT and computing infrastructure and expertise at the time, and the development of Master's degree courses in computer science beginning in 1982.[44] Of the sixteen persons, three were trained for their PhDs in the US, six in Canada, two in England, one in Australia, one in India, one in France, one in South Africa, and one in Singapore. The Master's program was developed with a focus on problem-solving techniques, with a curriculum explicitly "based on those developed by well-known American software and computer institutes such as the Association for Computing Machinery and Institute for Electrical and Electronics Engineers" (Foo 1984). The connection to these professional institutions was also an important part of later curriculum developments.

By 2000, the program had been restructured again both by the University's shift to a modular system beginning in 1994 and by the department gaining greater administrative and programmatic independence. This led to a diversity of programs found associated with computer science around the world including computer engineering and information systems, along with others such as computational finance that are more specific to the Singaporean context. While the Department also offered three-year degrees meant specifically to "meet national IT manpower needs," they also introduced a new four-year Bachelor of Computing program (extended from a previous three-year program with an optional honours year) which "emphasizes the fundamentals of the four pillars of computer science, viz, programming languages, theory/algorithms, architecture, and networking." A wide variety of courses were also introduced focusing on different

[44] This count does not include seven persons who were listed as "senior tutors" and who had not completed doctoral degrees at the time.

sub-disciplines. In 2005, the curriculum was revised again. The course calendar for that year explains the new curriculum:

> Designed according to the recommendations of the Association for Computing Machinery (ACM), the Association for Information Systems (AIS), and the Computer Society of the Institute of Electical Engineers (IEEE), which are the foremost authorities in the field of computing. Consequently, the knowledge units that are to be imparted with the curricula are recognized internationally, the practices instilled are sensitive to industrial development, and the pedagogy espoused adheres to IT education standards worldwide.

A later course calendar specifies that the program was structured to specifically follow the ACM and IEEE's *Computing Curriculum 2001* recommendations.

The Association for Computing Machinery was founded in 1947 at a meeting at Columbia University in New York.[45] While the ACM is an international organization, it is nonetheless centred on the US and other "Western" countries. In 2015, its membership comprised over 82,549 people, with over 43,946 from the US (53%) and another 2,556 (3%) or more from Canada. I do not have information subdividing the remaining 36,049 or more persons, but it is likely that a significant proportion were from European countries. Janet Abbate explores how the ACM (along with the IEEE) played a predominant role in shaping computer science as a "proper intellectual discipline" (Abbate 2017), debating and defining its boundaries and then working to become a gatekeeper for "proper" computer science education.

Since the 1960s, the ACM together with the IEEE's computer science branch have regularly published curriculum recommendations. The first of these curriculum

[45] The association was first named the Eastern Association for Computing Machinery with "Eastern" dropped in 1948.

guidelines was developed in 1968 by a committee of twelve people (eleven from US universities, and one from a Canadian university) (ACM Curriculum Committee on Computer Science 1968, 151). Acknowledgements for contribution and consultation were given to an additional sixty-three people (fifty-one from US universities, two from Canadian universities, nine from American corporations such as Bell, RCA, and IBM, and one from the US Department of Defense) (ACM Curriculum Committee on Computer Science 1968, 169). The 2001 curriculum guidelines involved much more expansive consultations in terms of the numbers of people involved, based on a short questionnaire that was mailed to the heads of all computer science departments, but only in the United States and Canada (Joint Task Force on Computing Curricula 2001, 7). The report also states that the questionnaire was made available on the Internet, but that "the vast majority of the responses still came from North America" (Joint Task Force on Computing Curricula 2001, 7). The committee heading the review and report was also largely composed of members based at US institutions (Joint Task Force on Computing Curricula 2001, ii).

As indicated by Temasek University's computer science course calendar, these curriculum guidelines shape courses offerings, course material, and overall degree requirements for computer science programs in the US, and in many places around the world (Dziallas and Fincher 2017, 92). Dziallas and Fincher outline how by 1983, "reports from both societies [the ACM and the IEEE] were often seen as prescriptive documents that specified the exact material that should be taught and in which order" (Dziallas and Fincher 2017, 99–100). This role for the curriculum guidelines was further reinforced as the joint organizations sought ways to mandate program requirements and accreditation. For a time, beginning in the 1980s, a Computing Sciences Accreditation Board (CSAB) operated independently, but it was later merged with the Accreditation Board for Engineering and Technology (ABET) in 2001 (Dziallas and Fincher 2017, 100–101). By 2008 the 3-year program at Temasek University that was designed to meet the national need for IT professionals had disappeared from the calendar, but the program had "gained recognition by ABET Inc.," explained in the course calendar as "the world respected US based accreditation agency." Accreditation "certified that the programme is 'substantially equivalent' to accredited programmes in the United States, which is the standard status that the agency offers to universities outside the United States."

Along with curricular changes, by 2013 (the time of my fieldwork), the number of faculty associated with computer science and related programs at Temasek University had grown significantly, to seventy-four members. Many, however, continued to be trained in North America or Western Europe: thirty-six were trained in the US (48%), nine in Singapore (12%), seven in the UK (9%), five in Australia (7%), four in Canada (5%), and the rest in a variety of countries including Finland, France, Germany, Hong Kong, India, Japan, the Netherlands, and Poland. Thus, while computer scientists in Singapore are being trained in a wide variety of places, including a larger number by local universities, the largest proportion continue to be trained in North America and specifically the US.

As a result of these multiple connections, computer science education at Temasek University has been and continues to be closely tied to computer science technologies, curricula, and expertise from the US. Moreover, the constitution of computer science knowledge as it is (re)produced and cited in undergraduate education at Temasek University draws on similar connections. Some professors will frequently refer to the inventor or author of a particular idea, concept, or theorem. One professor did this regularly, usually referring to at least one person per lecture, often more. However, all other professors whose classes I observed similarly made references to people seen as significant in the history and development of computer science, albeit with less frequency. Most of these people are mathematicians or computer scientists, but they also include software engineers, logicians, philosophers, and scientists. Figure 4-3 shows the birth places of the different referenced persons throughout the first academic term of my fieldwork (thirteen weeks).[46] As you can see, these are heavily centred on the US and otherwise Europe. Textbooks used in courses also largely originated from or were written by experts from the US. For example, three of the authors of the CLRS text are from the US and one from Norway.

[46] This count only covers half of the courses in which I conducted observations, excluding the second half of my fieldwork. It is meant as an illustration of the specific US-centric citations for computer science knowledge. While the numbers would likely be slightly different for a count of the second term, with a potential for greater diversity in countries of birth for researchers cited in upper-year courses that focus on more recent research in CS sub-disciplines. The predominance of US-based research and researchers would, however, persist.

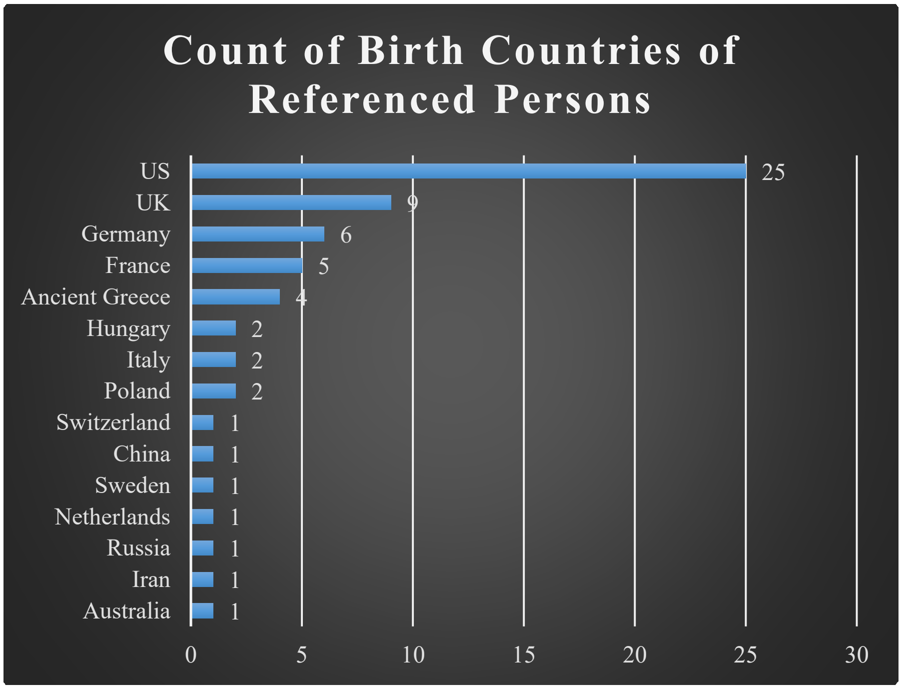

Figure 4-3: Place of birth of people referred to in class over the first term of fieldwork

In many ways these connections are unsurprising. Research on the history of programming also locates the origins and development of programmable computers and programming languages mostly in the US and parts of Europe (e.g. Priestley 2011). These connections also follow the teleological portrayal of "progress" in relation to "the West" discussed by Eric Wolf:

> We have been taught, inside the classroom and outside of it, that there exists an entity called the West… many of us even grew up believing that this West has a genealogy, according to which ancient Greece begat Rome, Rome begat Christian Europe, Christian Europe begat the Renaissance, the Renaissance the Enlightenment, the Enlightenment political democracy and the industrial revolution. Industry crossed with democracy, in turn yielded the United States, embodying the rights to life, liberty, and the pursuit of happiness (Wolf 1997, 5).

Indeed, ancient Greek philosophers and mathematicians such as Aristotle, Socrates, Euclid, and Pythagoras were cited, along with Leonardo Fibonacci in the late Roman Empire, Rene Descartes and Pierre Fermat in the late Renaissance, Johann Carl Fredrich

Gauss and Arthur Shopenhauer in the Enlightenment, George Boole, Augustus de Morgan, Ada Lovelace, and John Venn (all from the UK) in the Industrial Revolution, and then multiple persons from the US, the first of which is Grace Hopper born in 1906. These figures were certainly not presented in this order or with this explicit narrative. They were cited across several courses in different orders and for varying purposes. Yet, a (disjointed) picture nevertheless emerges that computer science knowledge (in the present and in relation to its claimed mathematical, philosophical, and logical predecessors) centres on and emerges out of the "West" and most recently the US, with only a couple of exceptions.

In relation to Silicon Valley, Lucy Suchman (2011b) discusses how it often positions itself as the vanguard of future-making – portraying itself as the centre of development while colonizing other times and places with its singular vision. The discipline of computer science is situated more broadly, but nonetheless is often portrayed and promoted as the future in and of many places, including Singapore. Computer code is "the language of the future" as proclaimed on the department's website, and student-made t-shirts and displays assert that "we code the future." Sareeta Amrute describes how programmers in India saw code and virtual reality as a kind of utopia because it offers less limitations and constraints than "embodied worlds," a vision that is meant as a critique of the borders, boundaries, and limitations that prevent programmers' own mobility (Amrute 2014, 114).

Yet, the mobility of computing worlds (and their worlds of practice) are also not without friction. As Yuri Takhteyev explores in relation to programmers in Brazil "local participants orient themselves toward such meccas in an attempt to draw on their symbolic power and to bring the local practice closer to the remote standards" (Takhteyev 2012, 208). That is, computing worlds of computing practice are structured such that they centre on places like Silicon Valley, and in order to gain prestige, recognition, legitimation, and mobility, actors must orient towards the practices and values of such centres or "meccas." On the other hand, actors developed in and from such centres are "often mobile from birth" (Takhteyev 2012, 42). Computer science in Singapore is clearly oriented towards computer science in the United States. However, the directional movement of computing worlds also (re)produces gender norms – heteronormativity, in particular.

4.3.2 Heteronormative Traditions in Computer Science

Widely circulated textbooks and problem definitions rely on and reproduce similar genderings to those discussed above by students and professors at Temasek University. One of the clearest of these is the "stable marriage problem," which was discussed in one of the courses I observed, but is also widely referred to by this name in general computer science discourse. The problem consists of two equally sized sets A and B, and the goal is to find a matching between them that is "stable," where a matching between a_i in A and b_i in B is not stable if there is some element in B that a_i prefers (i.e. gives a higher ranking) over b_i and likewise there is some element in A that b_i prefers over a_i. However, the problem is often explained in terms of men and women. For example, an article co-authored by a business professor at the National University of

Singapore under the title “Linear Programming Brings Marital Bliss” describes the problem as follows:

> Alan, Bob, Carl and Dan, the only four bachelors in oddtown, finally contemplate marriage. They approach Marx, the matchmaker, who introduces them to Alice, Brenda, Cindy and Debbie. After the meeting, each person ranks all the members of the opposite sex and hands it to Marx… Marx’s job is to find a match for each man, and his reputation depends on the number of successful marriages arranged. What should Marx do? (Sethuraman and Teo, n.d., 89).

The paper then explains the problem of an unstable marriage:

> “Suppose he matches Alice & Alan, Brenda & Bob, Cindy & Carl, and Debbie & Dan. Observe that Dan likes Brenda better than Debbie (his current partner) and Brenda likes Dan better than Bob (her current partner); so, the proposed marriage would ‘break-up,’ and Dan & Brenda would ‘elope.’ We conclude that the proposed marriage in this case is ‘unstable’” (Sethuraman and Teo, n.d., 89).

This is a problem in the mathematical and computing area known as graph theory, where graphs are generally represented by points, sometimes joined by lines that connect these points. The problem dates back to at least the 1930s, with a version of the gendered problem definition described as early as 1949 by Hermann Weyl, a German mathematician (Wagner 2009, 291; Weyl 1949).

The heteronormative and binary gender assumptions in the problem definition are immediately apparent. However, Roy Wagner (2009) extends the gendered analysis to discuss the algorithms used for solving this problem. He starts with the oldest and most popular Gale-Shapley problem and algorithm, which introduces a component of individual preferences, and shows how it follows and reproduces gendered stereotypes about men and women. “A particularly disturbing feature in the various narrations of the algorithm,” he argues, “is that in *none* of the surveyed texts do women ever say ‘yes’ to the marriage proposals. Their replies are either a definite ‘no’, a deferring ‘maybe’ or a silence, which is interpreted as provisional or permanent consent" (Wagner 2009, 295). More generally, the algorithm is strongly biased towards matchings that are best for all men and worst for all women in terms of their ranked preferences (Wagner 2009, 295).

Wagner suggests that innovations in new algorithms have been influenced by these gendered stereotypes. For example, algorithms that produce more balanced matchings were only developed much later (around 1990) and the problem itself and all solutions rely on pervasive preference for stability (Wagner 2009, 297–99). The Gale-Shapley algorithm was published in 1962 by David Gale and Lloyd Shapley, both American mathematicians and economists (Gale and Shapley 1962). As discussed above, students at Temasek University also reproduce the heteronormative genderings outlined in the definition of the problem and the various algorithmic solutions, giving them a national Singaporean flavour with the involvement of the government (albeit represented in students' story assignment as the US government) in governing "matchings" to aid in the problem of national population reproduction.

The stable marriage problem is the most explicit heteronormative gendering that I encountered during my research, but it is not the only example in computer science. Another oft-cited example is one version of the "Imitation Game" proposed by British computer scientist Alan Turing to answer the question of "Can Machines think?" The game described as follows:

> It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman (Turing 1950, 433).

In the game, A attempts to mislead the interrogator and B attempts to help the interrogator. Turing then asks: "'What will happen when a machine takes the part of A in this game?' 'Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?'" (Turing 1950, 434). This game is a recreation of a popular Victorian parlor game, with a man, a woman, and a judge of either sex/gender (Brahnam, Karanikas, and Weaver 2011, 408).

There are several implications that can be taken from this game. Several authors discuss the way this game destabilizes gender binaries since in the game gender is merely about imitation – separate from embodied sex and negotiated through language and symbols – made all the more meaningful by Turing's committing suicide not long after the article was published in relation to the British government's prosecution of his homosexuality (Golumbia 2003, 37–38; Halberstam 1991, 441–45; Hayles 1999, xiii–xiv). On the other hand, others point to the ways that the test nevertheless relies on a binary distinctions between men and women and that machine player working to imitate a woman in order to pass the test will likely draw on and accentuate stereotypical gender differences (Brahnam, Karanikas, and Weaver 2011, 408; Hayes and Ford 1995, 973). The Test also focuses exclusively on judging a machine's intelligence relative to a human's (Hayes and Ford 1995, 974–75), the definition of which scholars have argued is inherently unstable and relative (Riskin 2003; Suchman 2007a, 2011a).

Given that Turing was writing in 1950, the Turing Test cannot be expected to adhere to current concepts of gender equality, gender diversity, and posthumanism. Yet, like Wagner's discussion of the stable marriage problem, Patrick Hayes and Kenneth Ford argue that the requirements of the Turing Test have influenced the questions and definitions that have continued to constitute research and practice in artificial intelligence.

> The Turing Test indeed challenges a computer to simulate a woman, rather than be one… Turing correctly insisted that his test was not meant to define intelligence. Nevertheless, in giving us this touchstone of success, he chose human intelligence – in fact, even more peculiarly, the arguing skill of the educated English middle class in playing a kind of party game – as our goal. (Hayes and Ford 1995, 976).

They discuss how various branches of computer science and robotics have disavowed their association with artificial intelligence in search of other approaches, while those working within the field have been limited (or had their success judged) by the requirements of the Test (Hayes and Ford 1995, 974–75). Instead, Hayes and Ford argue against a focus on imitating human intelligence and instead expand the boundaries of Artificial Intelligence to encompass all techniques that "do something intelligent" or "display a cognitive ability" regardless of their techniques or relationship to human cognition (Hayes and Ford 1995, 977). While I did not observe any discussions of the Turing Test in this form during my research, it is clear from Hayes and Ford's discussion that they see the gendered form of the test as having significantly shaped and limited, in its specificity, the field of Artificial Intelligence around the world. Additionally, Turing himself was referred to several times by different professors, one of whom explained that he is often called "the father of computing," and another suggesting he was the father of students themselves as nascent computer scientists.

A final example that I will consider ties to examples reproduced in several "classic" texts, including the CLRS "Introduction to Algorithms" textbook discussed in the introduction to this chapter. The gendering is more subtle than in the cases discussed above and relies simply on the repeated use of two figures, "Alice" and "Bob" (standing in for non-descriptive figures A and B), who want to communicate securely. As the authors of the CLRS text note: "The participants 'Alice' and 'Bob' are traditionally used in cryptography examples" (Cormen et al. 2009, 959). The problem is as follows: "suppose Bob wishes to send Alice a message M encrypted so that it will look like unintelligible gibberish to an eavesdropper" (Cormen et al. 2009, 960). The authors proceed to outline different cryptographic algorithms including public-key encryption and digital signatures using Alice and Bob for their examples. In his multi-volume work "The Art of Computer Programming," Donald Knuth uses the same figures for public-key encryption and digital signatures: "As an example of what can be done when the encoding method is public knowledge, suppose Alice wants to communicate with Bob securely via electronic mail, signing her letter so that Bob can be sure nobody else has forged it" (Knuth 1998, 406). Figure 4-4 shows a popular web-comic commenting on this recurring use of names by computer scientists and cryptographers, showing its circulation and ubiquity in computer science discourse.[47]

[47] The comic also refers to Eve, a third character not discussed above, who is a short form for "Eavesdropper."

Figure 4-4: Comic about commonality of "Alice" and "Bob" in Computer Science (xkcd n.d.)

The binary gendering is enacted simply through distinct gendered names, yet when students retell the story the heteronormative and heterosexual implications become more explicit. One student group explained prime numbers and RSA cryptography similarly referring to Alice and Bob. They told a story of a student who meets Ron Rivest, an American cryptographer partially responsible for inventing RSA cryptography. In the story, Rivest tells the student a story about Alice and Bob, whose "hobby is to write mushy love letters to each other" and Rivest works as their part-time postman. Bob and Alice do not want the postman to read their letters and so they develop the system of public key cryptography so that their letters are safely and secretly exchanged and "Alice and Bob live happily ever after," including having a child together.

The repetition of problems in the same form, seen clearly in the Stable Marriage Problem and in cryptographic usage of figures "Alice" and "Bob" work as a tradition. In his well-known book *Keywords: A Vocabulary of Culture and Society*, Raymond Williams suggests that, "tradition survives in English as a description of a general process of handing down… But the word tends to move towards *age-old* and towards ceremony, duty and respect" (Williams 1983, 318–19). Among seven uses of the concept of tradition in Folklore and other disciplines, Dan Ben-Amos outlines "tradition as process" where "the process of tradition implies the dynamics of transmission of cultural heritage from generation to generation" (Ben-Amos 1984, 116–17). In another usage of "tradition as performance," Ben-Amos adds that "dynamic variations occur in performance, in speaking, singing, music making, painting and sculpturing" (Ben-Amos 1984, 124). The passing down of algorithms often occurs through written texts, as with those discussed here, unlike the "traditional" focus on orality in relation to speaking, singing, and music making, for example. However, as seen with the stories around the Stable Marriage Problem and Alice and Bob, the written retellings are nonetheless performed with variations. The handing down of algorithmic descriptions and conceptualizations along particular heteronormative gender lines thus easily fits with these definitions of tradition.

Many scholars have explored how tradition can function as a vehicle for power, considering how it is used in various projects such as nationalism or class domination (Handler 1988; Hobsbawm and Ranger 1992; Rosaldo 1993; Williams 1977; Wolf 1997). Selective tradition, in particular, is used by the hegemonic class to create a dominant history fabricated from the range of available historical traditions to enforce hegemonic ideals (Williams 1977, 115). The dissemination of this selective tradition depends on its institutionalization; that is, the socialization of the dominant view of history and perceptions of the world through schools, the family, and other areas of learning (Williams 1977, 117–18). Eric Hobsbawm and Terrence Ranger (1992) additionally introduce the concept of "invented tradition." Hobsbawm explores how certain "'traditions' which appear or claim to be old are often quite recent in origin and sometimes invented" (Hobsbawm 1992, 1). He also discusses how these traditions are used to socialize or inculcate certain ideals or values, particularly in support of nationalism. Several scholars, however, have critiqued the idea that some traditions are obviously and deliberately invented because of the implicit assumption that certain other traditions are more "authentic" (Handler and Linnekin 1984; Linnekin 1983; Ray 2005). Handler and Linnekin also point out that all traditions have an element of selection and invention (Handler and Linnekin 1984).[48]

Computer science traditions are certainly invented in that the discipline focuses on what Herbert Simon has called "the sciences of the artificial" (Simon 1996). Simon posits "artificial" in contrast to "natural," a distinction that replicates similar problematic dichotomies such as nature versus culture discussed in Chapters 1 and 3. Nevertheless, the term is useful for highlighting how "artificial things are synthesized (though not always or usually with full forethought) by human beings" – the first of Simon's four characteristics that distinguish artificial things (Simon 1996, 5). That is, computer science knowledge has been built on a foundation of human creations, some of which are tied to

[48] See my MA Thesis for a more detailed discussion and literature review of concepts of "tradition" and its performance (Breslin 2011).

the ideas about the “nature” of reality as discussed in Chapter 3, but nonetheless “synthesized” through human invention.

As Celeste Ray argues in her work on those claiming Scottish ethnicity in North America, however: “that tradition is invented does not detract from its present meaning to those who emotionally invest in its practice” (Ray 2005, 6). Computer science traditions could be distinguished from those relating to Scottish heritage, for example, in their deliberate invention for a purpose explicitly unrelated to heritage, culture, or identity. Nevertheless, in the circulation of stories and variations the process of transmitting computer science knowledge works as a traditional process, even if invented. Additionally, in both cases where the history of invention is forgotten – or excluded – from discussions and teachings, the practices become naturalized as a part of history or reality.

Through the performance of selective traditions in the telling, teaching, and definition of algorithms, cuts or boundaries are repeatedly made between male vs female and masculine vs feminine, as much as between social vs technical and developer vs product, as discussed in preceding chapter. In this way, in their continued retelling and circulation, these traditions selectively (re)produce social assumptions about the “nature” of gender as being based in heteronormative gender binaries. Research in folkloristics has explored representations and performances of masculinities and femininities through various traditions (Brandes 1980; Bronner 2005; Greenhill and Tye 2014). Simon Bronner, for example, introduces his book on *Manly Traditions* by explaining:

> By focusing on traditions, the book seeks to interpret the mechanisms by which masculine values are maintained, adapted, invented and discarded… these

> traditions are critical in explaining socially particularized states of masculinity because they are by their nature vessels for creating meaning, producing metaphors, reinforcing beliefs, and transmitting values through time (Bronner 2005, xi–xii).

Similarly, the iterative intra-actions entailed in gendered computing traditions reinforce beliefs and transmit values over time in relation to heteronormativity. In this regard, they contribute to the socio-technical field of action such that other ways of thinking about and doing both computer science and gender are made less possible (Foucault 1982, 1991; Wolf 1990).

4.4 Trans-national Transmission and Tension

While these gender norms tie in with Singaporean state discourses, laws, and policies, they are also part of transnational disciplinary discourse and practice in computer science, which have particular geographies and histories spreading from the United States and Western Europe. The directional frictions of computing worlds of practice promote the mobility of specifically rendered and gendered American computer science and its colonization of other times and places. Yet, these intra-actions across scales and national contexts also have the potential to produce tensions.

Research on gender and computing has tended to focus on a single national context (El-Bahey and Zeid 2013; Lagesen 2008; Margolis and Fisher 2002; Turkle 2005), or theorized about the relationship between gender and computing without reference to particular places or cultural contexts (Golumbia 2003; Wajcman 1991). While some research has considered gender in relation to information and communication technologies for developing countries/development (ICT4D) and human computer interaction for development (HCI4D) (e.g. Dodson, Sterling, and Bennett 2013;

Light et al. 2010; Purushothaman 2013;), Shklovski et al. point out that these approaches tend to reify national difference to determine the degree and relevance of "otherness" (Shklovski, Vertesi, and Lindtner 2014, 12). Despite their different foci, research on gender and computing and scholarship on transnational computing contend with similar "matters of concern" (Latour 2004): the reproduction of social norms and values and the marginalization of users (women or users in developing countries) where there are frictions between design and use.[49]

As seen above, these issues are embedded in and emerge out of computer science practice, beyond the level of the interface. That is, all facets of computing including the hardware, compilers, programming languages, programming problems, teaching examples, algorithms, data structures, and so on, embody wicked problems in that they intra-actively constitute and reproduce gendered social, cultural, political, and economic values and practices (Rittel and Webber 1973). Additionally, these intra-actions work across national boundaries and entail directional frictions that often privilege values and practices from the US and other "Western" countries (Takhteyev 2012; Tsing 2005). In the case of Singapore, the reproduction of heteronormativity in computer science education and as part of the computer science discipline largely aligns with national policies and interests.

In one case, however, a professor challenged these norms. The professor was using a dating app as an example for modelling processes within a program. He started by

[49] A growing body of research is also exploring the frictions, movements, and inequalities across places and countries in and through computing and IT cultures and practices (Takhteyev 2012; Amrute 2010, 2016; Biao 2006; O'Donnell 2014).

saying "sometimes woman objects want to date man objects." Then, after discussing the diagram for a short while, made a small modification to the diagram on the slides that allowed for women to date other women and men to date other men, saying "now, if we allow same gender dating." There was laughter throughout the lecture hall and a sense that students were mildly shocked, to which the professor responded "I don't see why not?" The model was further generalized to allow persons to date other persons. When I asked the professor about this example, he explained that students could be making software for people all over the world, so they need to be aware of differences. Yet, even if those differences were implicitly situated elsewhere, the students' reactions certainly indicated that the discussion of homosexuality was out of the ordinary. There is thus a tension in the intra-action of transnational computing practices with national policies and norms.

There are also indications that the potential for conflicting intra-actions in Singapore in relation to norms of gender and sexuality is growing. In 2017, the Singaporean government prevented foreign companies, many of which were multinational tech companies such as Google, Apple, and Facebook, from sponsoring PinkDot, the only local LGBTQ pride event. The government argued that these "foreign entities should not fund, support, or influence events that relate to domestic issues, especially political issues or controversial social issues with political overtones" (Chua 2017). Although such policies ignore the multiple ways that tech companies and technologies have and continue to influence "domestic issues" in Singapore in more subtle ways, they also points to a recognition of the power and influence of these companies (and potentially also of their products). The conflict surrounding PinkDot also suggests a divergence between Singaporean policy and discourse in tech companies that focuses on promoting LGBTQ acceptance and diversity.[50] There is also increasing research on queer computing and technologies, suggesting that this divergence has the potential to grow in the future (Cockayne and Richardson 2017; Gaboury 2013; Landström 2007; Light 2011).

Much research on globalization (and on technologies) has explored the local and particular ways that all things “global” have been adopted, adapted, and reconfigured in particular contexts and by particular people (Caldwell 2004; Grinshpun 2014; Tsing 2005). Karen Ho (2005) additionally points to the ways that images of being “global” are produced by multinational finance companies, but in actuality entail limited and partial connections in some (key) places around the globe. The apparent significance of transnational connections in relation to computer science in Singapore are likely accentuated by Singapore’s policies and practices that foster such connections, and more generally an image of the city-state as globally interconnected (Carver 2010; Huat 2011; Tan 2003). At the same time, while more research is needed, this chapter points to some of the specific ways that (US-centric) gendered discourses and practices are reproduced transnationally.

[50] I intentionally say that it is discourse, and not necessarily practice, among these companies that promotes diversity and inclusion, given the ongoing issues faced by women and minorities working in the tech industry, and particularly in Silicon Valley (e.g. Benner 2017).

4.5 Conclusion

A heteronormative gender binary was generally assumed and enacted by students and professors as part of teaching, learning, understanding, and doing computer science at Temasek University. Heteronormativity was embedded in language usage, teaching examples, and computer science concepts, as well as broader discourse, practice, and administration in the department and university. Gender binaries, the equation of sex with gender, and heterosexuality were thereby taken as a natural and objective facet of reality, with few exceptions. These performances of heteronormativity can be tied, in part, to gender norms in Singapore. Singaporean state policies and discourses promote (heterosexual) marriage and procreation in Singapore, where Singaporean citizens are treated as the city-state's primary resource and families as the basic unit of the society and nation. However, while the Singaporean state does powerfully manage its citizens and subjects, various actors in relation to the computer science discipline – including technologies, textbooks, theorems, algorithms, and experts – that move and circulate transnationally also intra-actively produce heteronormative discourses and practices.

Throughout its history, computer science in Singapore and at Temasek University has been significantly shaped by the discipline's development outside the city-state, centring particularly on the US. The telling and retelling of particular narratives around computer science theorems, problems, and algorithms constitute a selective tradition of heteronormativity embedded in computer science knowledge and practice. This and the preceding chapter thus point to the significance of "classic" concepts such as ritual, initiation, and tradition for the study of computing and technological cultures, and for understanding the reproduction of gendered and other norms, values, and practices. The selective traditions in computer science often originate from the US and are mobile from birth, based on their origins in the centre of computer science worlds of practice. While these traditions currently dovetail with Singaporean governance, working to limit the possibilities for thought and practice in relation to gender in computer science in Singapore, there are also tensions in the intra-actions between transnational computing discourses and practices and national policies and interests. The next chapter further considers the values and behaviours that are cultivated among students in Singapore, as well as how students (like code and programs) are judged and assessed in terms of their performance. Students themselves are rendered technical and thereby comparable and competitive.

Chapter 5 : Rendering Students Comparable and Competitive

When I was not sitting in on classes and tutorials, attending public lectures, exploring different parts of Singapore, or simply trying not to turn into a melted puddle from the heat and humidity, I spent a great deal of my time reading Singaporean official and independent news media, as well as local blogs and other social media. I found particularly useful and intriguing several social media pages for anonymous "confessions" by students and others at different universities in Singapore, and specifically Temasek University. People would send in confessions, comments, or stories, which would then be posted publicly and anonymously. When reading posts on Facebook for one of these confession sites, I noticed several references to the "Bell Curve God." In response to a "freshie" posting about a midterm that did not go well, for example, others responded "be very very good friends with the bell curve god! You will grow to love this new friend of yours.... we all do!" and "Don't worry, the bell curve god always plays fair." Another post commented: "I used to be an atheist… Now, I worship the Bell Curve God and pray frequently to it after my exams."

There were numerous similar posts, along with news reports about students who had built altars to this Bell Curve God, a Twitter account in its name that responded to students who tweeted with prayers, and one student who had made a website with an image of the God and a "Pray" button that showed an animated image of joss sticks when clicked. The Bell Curve God could have many different meanings for students, it could offer a form of fun and stress relief, a way of gaining control over a high-stakes and indeterminate situation, or a means of gaining luck. Nevertheless, in all cases, the Bell

Curve God's significance relates to the pervasive use of the bell-curve in the Singaporean education system, across several national examinations and most university courses. The Bell Curve God provides a light hearted, but also serious, instantiation of students' sense of the ways they are judged and compared throughout their studies and lives, in this case according to their grades.[51]

The preceding two chapters explored the values, behaviours, and norms in relation to the construction of reality and gender embedded in the technical knowledges and skills that students learn, such as particular technologies, programming languages, and mathematical theorems. This chapter continues this exploration of hidden or implicit curricula, considering the values and behaviours fostered among students as part of national education in Singapore, and specifically computer science education at Temasek University. I begin by providing an overview of the education system and educational policies in Singapore. As part of this national context, I then explore how norms relating to independence and cooperation are cultivated, and the relationship between these diverging qualities. In particular, I show how students learn to become independent and autonomous people who "network" and instrumentally seek knowledge through cooperation with others. As part of this process of becoming a computer scientist, students also learn to judge themselves and others, just as they learn to judge computer

---

[51] As suggested by these numerous references to the Bell Curve God, students' grades for most courses are modified to fit a bell-curve (also known as a Normal distribution, or a Gaussian distribution). The effect of the bell-curve is to modify grades according to their relative rank among all test takers. For university examinations, this is not an absolute ranking, but rather a moderation so that only a certain (small) percentage of the grades in a class will be As, the majority of grades will be Bs and then Cs, and another small percentage will be Ds or Fs. In this way, the distribution of grades roughly fits a Normal distribution.

code. In this way, students are rendered comparable – and competitive – as they are summoned to compare themselves with one another locally and (trans)nationally.

5.1 National Education

Singaporean national education operates through a streamed system where, from primary school to university, students in Singapore are continually tested and evaluated, beginning with the national Primary School Leaving Examinations (PSLE) around the age of 12 or 13. The results of this and later Singapore-Cambridge General Certificate of Examination (GCE) tests are significant for shaping the course of a student's future. This system has its legacy in British colonial administration where the GCE "was highly esteemed as it was a pathway to employment in government and commerce during colonial days" (Koh 2004, 163). Figure 5-1 illustrates the multiple tracks and paths that students can take through the Singaporean education system. As Li-Ching Ho explains of a students' academic course in secondary school, following the PSLE:

> These students are ranked according to their national examination results and the best students join the Integrated Programme (IP) – a 6-year continuous programme offered by 11 elite secondary schools…. IP students have greater exposure to a range of courses and topics, including philosophy and scientific research. The majority of secondary schools in Singapore offer the academically demanding 4- or 5-year Express and Normal Academic (E/NA) tracks, and the 4-year vocational Normal Technical (NT) track… The Ministry of Education places the least academically able students ~ 13% of each cohort, in the NT track… Unlike their counterparts from the IP, E/NA tracks, the NT students are limited to a narrower and less academic range of subjects, including Food and Nutrition, Computer Applications, and Elements of Office Administration (Ho 2012, 409).

Students in the E/NA track will take the GCE Ordinary Level, or "O-level" examinations that will then sort them into Polytechnics versus Junior Colleges. In the latter, they will prepare for the GCE Advanced Level or "A-level" examinations, which will shape their possibilities for university study. Once students have completed their national examinations and entered university they continue to be evaluated in their courses, through assignments, project work, class participation, midterms, and final examinations. These methods of assessment are then used to calculate students' course grades, which are then aggregated into their Cumulative Average Point (CAP).[52]

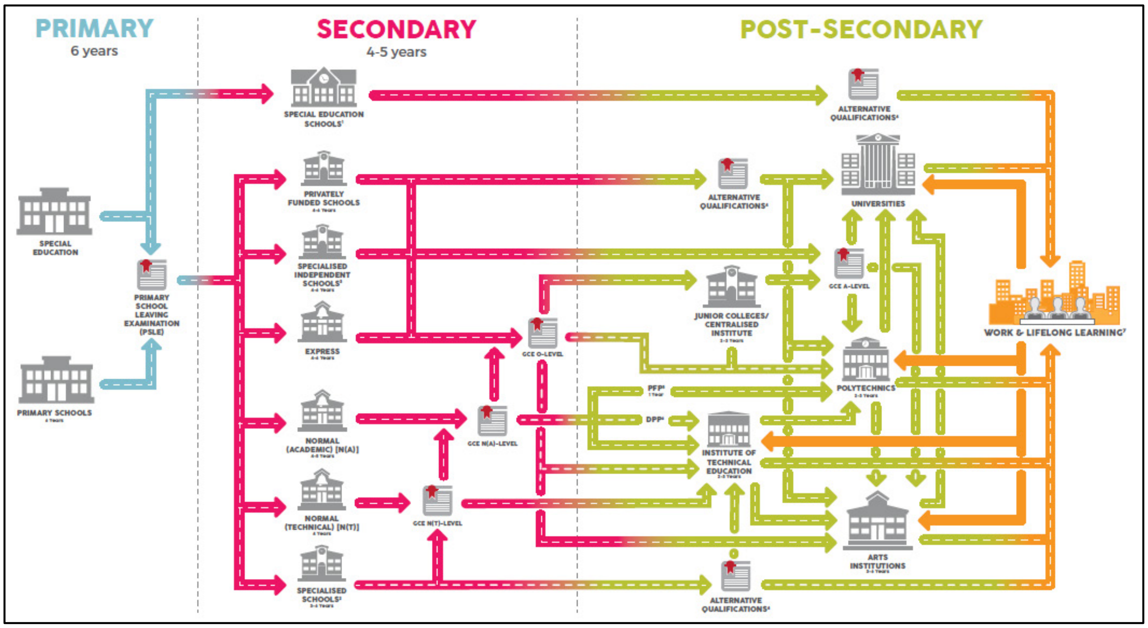

Figure 5-1: The Singaporean education system (MOE 2016)

This system is based on pervasive ideals and discourses of merit in Singapore. While "meritocracy" was originally termed as a critique of the confluence of merit with social class and the resulting reproduction of privilege and power, the underlying critique has largely been lost in the term's usage in popular and government discourse (Young 1994, 2008). In his 2013 National Day Rally speech, Singaporean Prime Minister Lee Hsien Loong, for example, expressed that "meritocracy has to remain the most

[52] CAP in Singapore is equivalent in meaning to "Grade Point Average" or GPA, which will be more familiar to readers in the US and Canada.

fundamental organising principle in our society. We have to recognise people for their contributions and their effort, not for their backgrounds, not for their status or wealth or connections." (Lee 2013). Again in his 2014 speech PM Lee also commented on how the "enduring values" of "meritocracy, multiculturalism, and modernisation" have been and continue to be key to Singapore's success as a nation (Lee 2014). Michael Hill and Lian Kwen Fee outline how meritocracy became a guiding principle for Singapore in the 1970s due to its compatibility with "multiracialism," which was another ideal also espoused by the PAP following independence (Hill and Lian 1995, 101–2).

Singapore's official policy of multiracialism is based on the four "races" that are seen to constitute the city-state: Chinese, Malaysian, Indian, and Other, conveniently captured in the initialism CMIO. These racial categories were then taken up by the Singaporean government following independence to become a common feature of Singaporean life. The four categories are represented daily in multiple areas: they are listed on personal ID cards, depicted on currencies, embedded in censuses, in government reports, the school systems, and displayed through "cultural shows" (Benjamin 1997, 72–73). Differences between the categories have become reified, where racial difference is also often equated with distinct languages (normalized through bilingual education policies), as well as religion, and laws (Kong 1997, 87).[53] In conjunction with meritocracy, multiracialism purports that all races are different but equal, and so

[53] Students of different races are separated into different second-language classes: Mandarin for Chinese students, Malay for Malay students, and Tamil for Indian students. The racial categories are also associated with distinct religions, Buddhism or Christianity for Chinese persons, Islam for Malay persons, and Hinduism for Indian persons. Additionally, Muslim followers in Singapore are subject to the Administration of Muslim Law Act (AMLA) enacting distinct laws and administrative structures in relation facets of life such as marriage, divorce, inheritance, property, religious schools.

individuals should have the same opportunities and life-chances regardless of race. Differences in success are thereby a matter of personal effort or failure.

Students are thereby sorted into various educational streams, and ultimately various differentiated occupations, based on what is judged and evaluated to be their talent and effort as measured by highly standardized and competitive examinations (Ho 2012). For the Singaporean state this represents a "pragmatic" approach – another guiding ideology of the government – of managing labour demands for various sectors of the economy (Hill and Lian 1995, 189–93; Tan 2012). However, the system has also received a great deal of criticism in recent years for promoting elitism (Koh 2014). Susan, whose parents come from a working-class background, for example, explained her impressions of meritocracy and streaming in Singapore:

> I feel that Singapore has to work on improving its education system, because I feel that there's actually two different classes… what I know is that there's two different kind of groups. There's one group, they are rich and able to afford tutors, able to hire tutors to help them with their study and stuff like that. And there's another group whose parents are uneducated, they are staying in an environment that's very volatile, or maybe they are staying in an environment that's not [good] for their studying, or maybe and their parents most often than not have no money to hire tutors for them. So, they often like run into difficulties in their homework and stuff like that (Susan 2014).

As suggested by Susan, the system of streaming has been criticized for reproducing class and educational elites, which also tend to correspond with Singaporean Chinese racial dominance and privilege (Huat 1998; Li 1989, 1998a; Tan 2008).[54]

[54] During my research I met students from a wide variety of educational backgrounds. Many certainly had followed a relatively straight academic path directly to university, progressing from their PSLE to secondary school and then JC, succeeding in their A-levels to attend Temasek University. Several students I spoke with, however, completed Polytechnic studies first and then opted to continue on to University and study computer science. I also met one student who had taken the lengthy but clearly dedicated path first to study at an Institute for Technical Education (ITE), normally the path for a student on the NT stream, and

Scholars have also shown how the conjunction of meritocracy and multiracialism works ideologically to blame or credit particular communities for their economic positions (Huat 1998, 34–38; Li 1989, 179). This is reinforced as racial categories are associated with particular cultural characteristics and particular languages. Tania Li, for example, discusses how Malay economic and educational performance – referred to as the "Malay problem" – is seen as tied to cultural characteristics and the responsibility for change placed on the community itself (Li 1989, 1998a). On the other hand, cultural characteristics associated with Chinese, such as hard-work, diligence, and entrepreneurialism are seen to explain the economically dominant position of Chinese people over Malays and Indians (Li 1989, 134–s35).

Government policies beginning in the 1980s have increasingly sought to make the system more flexible as Singaporean government has also been actively working to change the "national habitus" of citizens (Koh 2004). National education has been a major facet of national development in Singapore and of the cultivation and management of its citizens as economic resources (Comunian and Ooi 2015; Gopinathan 2001; Hoofd 2010; Koh 2004). In 1997 the government launched the "Thinking Schools, Learning Nation" (TSLN) initiative. In his speech launching the initiative, then Prime Minister Goh Chok Tong expressed:

> A nation's wealth in the 21st Century will depend on the capacity of its people to learn. Their imagination, their ability to seek out new technologies and ideas, and

---

then to polytechnic, and finally reaching his goal of studying at university. This student's experiences are exemplary of the mobility ideally envisioned as part of meritocracy, but they are also not the norm in Singapore. As suggested by sociology and anthropology of education scholars, these occasional success stories work to reinforce the idea that the system is open and flexible, while reproducing systems of inequality for those who do not "make it" by treating their social immobility as individual deficiency (MacLeod 2009; Willis 1981).

> to apply them in everything they do will be the key source of economic growth. Their collective capacity to learn will determine the well-being of a nation (Goh 1997).

The type of education promoted by TSLN is contrasted with what is seen as typical Singaporean education, which is portrayed as relying on memorization and rote learning and related to the Singapore of the past as a clean, rule-abiding, hard-working, and hierarchical society. Citizens were taught to be obedient, compliant, and effective workers for manufacturing and other export-oriented industries prominent during the 1960s and 1970s and as part of Singapore's development as an independent nation under the pragmatic but paternalist guidance of the People's Action Party (PAP), the ruling governmental party since the state achieved self-governance in 1959 (Carver 2010; Tan 2003). These skills and characteristics, however, are no longer seen as adequate for Singapore's new and growing neoliberal and cosmopolitan knowledge economy (Carver 2010, 386; Kuo and Chen 1987).

Policies in the 2000s have continued to work to make the system more flexible by giving secondary schools, junior colleges, and universities greater discretion in admitting students based not just on their examination results (Gopinathan 2007, 62–63). Beginning in 2014, several Singaporean universities also instituted a "gradeless first year," or the option to exclude several modules in their first year from the calculation of students' CAP. This new initiative was meant to relieve some of the stress on first-year students while they are adjusting to university, to reconfigure the relationship between learning and evaluation, and to encourage students to take more risks and try new and different subjects they might not otherwise pursue for fear of failure.

The government has also developed policies across a wide range of sectors beyond education, including cultural (Tan 2003; Wong, Millar, and Choi 2006), economic (Kuo and Low 2001; Sim et al. 2003), and technological sectors (Arun and Yap 2000; Wong 1992) towards this effort to change the national habitus. In November 2014, for example, the Singaporean government launched the Smart Nation vision and initiative, the latest in a series of technological and economic development projects that began in the early 1980s outlined in Table 1, although the list is not exhaustive.[55] The particular goals of the many projects vary, but many seek to make Singapore a world leader in terms of technological use and expertise by cultivating Singaporean citizens as technopreneurial subjects: technologically skilled and entrepreneurial (Ong 2005). They also centre on embedding technology throughout the city-state and the lives of Singaporeans by developing infrastructure, software and technology industries, and technology education and training.

Date published	**Report Name**
1981	National Computerization Plan (NCP)
1986	National IT Plan (NITP)
1992	IT2000: A Vision of an Intelligent Island
2000	Infocomm 21: Singapore Where the Digital Future Is
2003	Connected Singapore
2006	Intelligent Nation 2015 (iN2015)
2015	Infocomm Media 2025

Table 1: List of key technology policy and planning reports/masterplans

[55]Singapore's National Research Foundation, for example, has published a similar series of planning reports beginning in 1991. The most recent of these reports is the *Research, Innovation and Enterprise 2020 Plan: Winning the Future through Science and Technology* (RIE2020 Plan) that considers science and technology research and development. While the scope and goals of both series of reports are significantly overlapping, in the interest of space, I focus on those related to information technology rather than science and technology, as they are more relevant to most undergraduate students in computer science.

A key facet of the National IT Plan, for example, was the development of IT education to foster local computing expertise. As several reports point out, prior to the 1981 National Computerization Plan there were only 850 computer professionals in Singapore. However, this number had increased to 150,200 by 2014 (IMMSC 2015b, 48), with a projected additional 30,000 jobs by 2020 (IDA 2016). The increase in the number of information workers, including technology professionals, over time was part of a broader shift in the structure of Singapore's economy towards knowledge and information industries (Kuo and Low 2001). Moreover, the government sought to make computer use a "literacy" of all citizens and a part of national education, ensuring that all public schools from secondary school up had computers in the classroom (Austin 1983).

Through educational programs, software development was meant to shift from being an ad-hoc skill to one focused on the needs of businesses, emphasizing large-scale, integrated, and team-based projects rather than being "subject to the whims and flairs of computer professionals" (Wong 1992, 1820). Wong Seng Hon outlines how, originally, the primary path for IT professionals was to learn through on-the-job training as there were no formal options outside of pursuing computer science as part of studies in science at university (Wong 1992, 1819). Through the National IT Plan the government instituted a variety of new options for formal training, setting the foundation for university programs like the undergraduate computer science degree at Temasek University. This chapter and the next consider the intra-actions through which both the "new" Singaporean technopreneurial national habitus and the specific disciplinary norms, behaviours, and values tied to computer science education at Temasek University are cultivated.

5.2 Independent Networking Selves

In his study of education in Papua New Guinea, Wayne Fife (1991, 1992, 1994) explores coincidental performances of "competitive individualism" and "cooperative individualism," tying these practices to a "new moral order" associated with a growing urban-centred cash economy, and "traditional" community concerns, respectively. Fife argues that occurrences of cooperative individualism in Papua New Guinea – where individual "performance and desire must be tempered with the needs of the group" – sometimes disrupt the dominant messages of "order, authority and hierarchy," but were generally secondary to cultivation of competitive individualism (1991, 333–38, 1992, 219–20). Fife explains, for example, how teachers would often discipline students for sharing their answers with others, thereby reinforcing competitive over cooperative individualism, except when this sharing occurred quietly and discretely or when situations such as shortage of supplies made it impossible for students to learn and display their knowledge individually (Fife 1991, 331–32). Cooperative individualism, however, could also coincide with new and dominant forms of social relations among workers in the emerging cash economy where, upon becoming a member of a group, a person should work to fit in (Fife 1991, 338).

I explore the ways both competition and cooperation were promoted and cultivated in at Temasek University and how independent learning was similarly emphasized over cooperation. In particular, cooperative learning was a significant *component of* students' individual learning practices, where, for computer science students, the two approaches worked as two sides of the same coin. Students learned to independently seek out knowledge in cooperation with others, but that cooperation often entailed the individual and instrumental mobilization of various resources.

5.2.1 Independent Learning

From the very first of my classroom observations, I noticed that independent learning was frequently emphasized by professors as a key component to learning and doing computer science. I repeatedly heard professors emphasize to students that they needed to learn course material on their own, to practise on their own, to learn how to find solutions on their own. Part of this emphasis was to help first-year students understand their responsibilities as new university students. During a first-year course early in the first semester, for example, a student asked whether they needed to buy the textbook. The professor answered that it was not compulsory and, indeed, nothing in university is compulsory. He further elaborated that students were adults and so no one would chase after them if they did no work or exams.

Beyond this general self-reliance and responsibility as part of university life, students were told to "have a very inquisitive mind" and were constantly prompted to figure out how a theorem or problem worked at home and on their own: "You go work it out," "go back home to think why," "you can go back home to read this proof again," "I would like you to go back home and try," "you can go back home to chase the algorithm." These are just a few of the hundred-plus quotes and examples by professors that fill my fieldnotes from classroom observations across courses from first- to fourth-year, emphasizing that students were expected to "chase" down problems and solutions related to course material, and other relevant or interesting topics, on their own and outside of class time.

This focus on independent learning and doing extended beyond repeated admonishments and was embedded in the pedagogy and design of courses and curricula. As one professor explained in class to students, “in any education system we can’t teach you everything under the sun,” and so students were expected to figure out how things work for themselves. In some cases the proofs of particular theorems or parts of a program were left out of course notes, or skipped in lecture, and students were responsible for learning the material independently and outside of class. In a first-year course, students were told by the professor that an example they were being shown had intentional mistakes to encourage them so that students would type and compile the program in order to figure out what was wrong. In this way, cultivating independent learning was a teaching technique, particularly in first-year courses, directing students to tinker, try, and work with the new computing technologies they were learning. Yet, it was also an ethos about *how* a person should be a computer scientist.

The development of this ethos was a process of self-formation by which students reflect on and shape their values and behaviours. For example, second-year student Dinesh was TAing for a second-year course he had taken the semester prior. When I asked him if he tries to emphasize anything to students beyond the general content of the course, he commented on how the course had taught him how to learn and how, now, it is something he emphasizes to the more junior students he teaches:

> Prof Sheldon does sort of encourage us… one of the things is learning to learn. I thought that was quite important I realized for the course. Initially even when I

> took the course I thought that the help wasn't enough, as in they were expecting us to set up stuff for ourselves. I see the value only later on when the whole trouble-shooting learning curve actually helped me apply to anything I was learning by myself. So, I do try to not spoon feed, I think that's an important thing to keep in mind. Because students tend to be happier if the tutor just tells you the answer straight away, [if] they say this is exactly what you need to do, exactly how you need to set it up. But I think that if I just point out some resources, mention some ideas and they do it by themselves then they feel more, they have a sense of achievement. I hope that they can appreciate that. I think that's important (Dinesh 2014).

A focus on independent learning was clearly adopted and performed by this student and TA. Although Dinesh explains how he initially did not see the value in pursuing independent learning, he clearly shows his belief in the approach now. By "learning to learn" Dinesh now pursues a recursive process of self-making, which he applies "to anything" he needs to learn, reiterating and demonstrating its value through his position as a TA. This process of self-making is dual-faceted: Dinesh and others are shaping their learning processes, as well as shaping what kind of knowledgeable persons they are by repeatedly seeking out and learning new things; it is an ongoing process of shaping *how* they learn and know, and *what* they know.

Most undergraduate courses had undergraduate TAs, like Dinesh, as well as graduate student TAs, who would teach tutorials or labs and consult with junior students about a course. These were paid positions. According to student TAs, they were hired based on their grades in that course previously or being recognized by the professor for the course. Senior students said they found working as a TA to be a useful experience since, in addition to earning some pocket money, students were able to review and better learn material from the course. Many also said they found it rewarding, albeit challenging, to teach others. TAs embody future-computer-scientists-in-the-making for the junior students they are teaching. As seen with Dinesh, they perform and emphasize for their students the values and behaviours, such as independent learning, of being a "good" computer scientist, particularly given their formally recognized role as TAs within the department.

Students are summoned to make themselves as independent learners who are continuously seeking knowledge that they need (or want) and lack, for their classes and exams, for personal projects, and for their future careers. That this process is seen to rely on personal and autonomous choice and discipline is clear in professors' observations that "If you don't help yourself, I cannot help you," as one professor commented in class. Another professor similarly explained to me how he very much wanted to help students who were struggling, but if they did not want to "help themselves" – if they did not come to class, hand in assignments, or seek out aid from TAs or professors – there was very little he could do.

Students also insisted on their autonomy in making choices, seen particularly when I would ask them about their family's influence or opinion on studying computer science. "When I chose to come to Singapore, when I even passed my interview and got my first like briefing meeting, they didn't know about it. It's all my choice" one student originally from China explained. A Singaporean student similarly commented "all the decisions are based on my own decision lah. As in nobody actually influenced me into wanting to study this. It's actually my own choice, my own freedom." Another Singaporean student explained "Actually my parents… they just encourage me to pursue any field that I'm interested in, yeah. But I told them that a technical field might be something which I'm looking into, so they just like give me encouragement and support all the way." It is clear here that students often see themselves as autonomous and independent persons who choose and shape their selves and their futures.

The ability to learn independently was an approach and skill that was taken for granted in later courses. One fourth-year student explained to me how, at the beginning of term, professors would simply say, for example, "I'm using Python for this course."[56] Professors thus proceed with the expectation that students either have prior knowledge of a given programming language, or they will learn it on their own. Learning where and how to find particular resources in order to learn was therefore as important as learning a given topic. This type of learning about resources included where to find program language documentation, understanding the indispensability of using Stack Overflow (an extensive online forum where people post and answer questions about programming), and asking friends and TAs for help. In this way students were supposed to learn course material through independent time and effort, but also that to be a good computer scientist they should mobilize others to seek out answers, and how to go about doing so.

In addition to being directly emphasized by professors, the kind of independent learning suggested above was associated with broader discourses in computing and tech about what it means to be a good computer scientist, a good entrepreneur, or even a good person. Figure 5-2, for example, shows one of a series of infographics comparing successful and unsuccessful people. These infographics were posted by a community member on *Tech in Asia*. The infographic suggests that successful people seek out new

[56] As mentioned previously, Python is a particular programming language.

knowledge and, at the same time, those who do not, are thereby unsuccessful. The infographic additionally implies that the latter are arrogant and lazy. Although the infographic's posting on a tech news site suggests it is directed largely at entrepreneurs and those involved in the tech industry, the language used explicitly encompasses all people.

Figure 5-2: Infograph relating success and independent learning (@kaushik reposted in Lee 2015)

More directly related to computer science, various blog posts and news articles circulating on the Internet and social media discuss ways to "be a good developer," which includes advice such as "constantly learning, practising and improving yourself is an investment in yourself and it's your responsibility, not your employer's" (Fekeke 2014). Similarly, on Quora – a popular question and answer site – in response to a question about “What can I do to be a good computer scientist?” answers included: “There are lots of tutorials out there, so you can check them out to learn further” and “Be open minded, curious and never stop learning” (Akkuş 2015; Sharma 2015). Another news article by Matt Weisfeld (2013), a professor and software developer, about “Becoming a Programming Rock Star: 5 Traits that Make a Great Programmer” suggests that programmers “show that you are good at independent learning… Basically, employers want programmers who are self-sufficient.” These articles largely centre on computer programming as a career in industry, which is one of the common paths for students in computer science. Nevertheless, they share a common message about the value of independent learning and of seeking out new knowledge beyond the time and scope of school or employment.

Many students I spoke with said that they often read tech news, and particularly Hacker News. It is therefore likely that some students would have seen these or similar news or blog articles encouraging them to be a “great programmer” or “good developer” through independent learning. Additionally, the computer science department at Temasek University runs a public Facebook page, posting similar articles and sharing them with the 37,189 people who have liked their page (as of August 2017), many of whom are students. One such post, for example, includes a comment encouraging students and other followers to “learn new skills” to show potential employers that they can “adapt and learn” along with a list of tech skills that are no longer in demand. This post thereby also suggests that students should focus on learning the *right* things, as required by classes or employers, as opposed to just continually learning anything.

This focus on independence aligns with the Singaporean government's goals to cultivate a new *habitus* among its citizens, discussed above. It is also closely related to liberal political subjectivities explored by Gabriella Coleman and Alex Golub in relation to hackers and open source developers in the US and elsewhere (Coleman 2013; Coleman and Golub 2008). Coleman describes liberalism as both a moral and political commitment entailing a variety of qualities. Most significant for my discussion here is "promoting individual autonomy and tolerance… and preserving a commitment to equal opportunity and meritocracy" (Coleman 2013, 2). These morals and politics are embodied among hackers in a selfhood characterized as "an autonomous being guided by and committed to rational thought, critical reflection, skills, and capacity" (Coleman 2013, 11). While this autonomous being shares much with a form of possessive individualism (Macpherson 1962), Coleman argues that they more accurately constitute an "expressive individualism" that "places tremendous weight on originality, sentiments, creativity, and at times, even disengagement" (Coleman 2013, 14).

Carla Freeman (2014) explores a similar kind of expressive individualism in relation to new neoliberal subjectivities among entrepreneurs in Barbados. She argues that flexibility is the focal point of the new practices of self-making occurring among entrepreneurs, entailing significant affective changes in the way relationships are cultivated equally with clients, partners, and children. In particular, the entrepreneurial self that Freeman elucidates entails "an independence in which an individual is defined as a self-propelled, autonomous economic actor, ever responsive to a dynamic marketplace, and simultaneously encouraged to seek introspection, self-mastery, and personal fulfillment" (Freeman 2014, 20). As with Coleman's hackers and Freeman's entrepreneurs, the independent learning cultivated among computer scientists is premised largely on a selfhood that is inherently autonomous, capable of making reasoned and critical choices, and focused on "progressing" or "growing" as a person through self-discipline and accumulating new knowledge and skills.

5.2.2 Cooperative "Networking"

While independent learning as a particular way of learning and doing connects to (trans)national neoliberal practices, discussed further in Chapter 6, it was seen as specific to computing in how it entailed seeking knowledge and resources via others. In this regard, values and behaviours related to autonomy, independence, and self-reliance frequently emphasized by students and professors operate alongside simultaneous practices of cooperation and solidarity. Even though the structure of grading such as the use of the bell curve, among other factors, encourages competition among students, I frequently saw students helping one another with various problems, in classes, in labs, and among friends. A second-year student from China explained how cooperation in learning at Temasek University was different from his previous experiences:

> **Sam**: Do you find it really competitive here?
> **Bao Jia**: Competitive, but more group, more cooperation.
> **Sam**: So, it's not like the other students are trying to get you?
> **Bao Jia**: Yes, not like that. We try, I think, we try to learn from each other. That's good, quite nice, and I didn't experience that in China. (Baojia 2013)

Many other students discussed how they found others helpful and willing to answer their questions, among both peers and seniors. In the first-year lab, I often saw students leaning over to talk to their neighbour or look at their neighbour's screen. The computers were arranged so that there were several rows of workstations facing the front, with three or four terminals beside each other. Sometimes students would get up and walk over to one of their friends sitting elsewhere in the room to ask a question or discuss an answer. I also saw a similar practice where students sitting beside each other in lectures would help one another understand the material, although this was less common than in labs given that there was rarely a time set aside for working on problems or for students to talk with one another during lecture classes. As a result, they would have to talk quietly and ignore what was happening in the lecture in order to discuss. In other words, the lecture format – and lecture theatres as classroom spaces – were much less conducive to this kind of cooperation.

Students also told me about how they would ask their friends for help. A third-year student explained how she would learn something new by asking a friend, including learning a topic such as web design which they had not yet learned in class:

> So, in most of the cases I'll get my friends to teach me. I'll get my friends, like let's say if they already know that. So I'll get them to like "can you teach me," "do you know how to do this?" I won't get them to tell me the answer, but I'll get them to show me how you got to the answer (Aindri 2013).

Aindri's answer displays the same focus on self-determined autonomous learning as discussed above in that, while she is asking for help she is also seeking knowledge by learning *how* to get the answer. Students, however, also said their friends were often busy, so asking them for help was not always an option.

Students' willingness to help others is tied, at least in part, to a sense of shared struggle. As discussed in Chapter 3, learning to program difficult for many students, in terms of learning the modes of thought and practice entailed in doing computer science and coping with the time commitments. As a first-year student commented about his first couple of courses: “I put it as one whole collective hell experience. But it’s fun going through that hell lah. Yeah everyone goes through it together, suffers together. It’s just fun lah” (Ariff 2014). Shared struggles help create solidarity among students, and programmers more generally. Ariff also explained how senior students were willing to help based on the recognition of shared suffering: “also fortunately for us we have a lot of seniors who are willing to help, lah. In a sense, because actually they went through the hell before, so they know” (Ariff 2014). A fourth-year student similarly recalled one of the first-year math courses that is widely considered to be challenging: “Yeah that was horrifying… and it still is for every freshmen. Because like it’s so weird, like nothing we’ve ever seen before” (Andrew 2013). Bao Jia, who discussed above how he found the environment more cooperative than expected, commented how he thinks every student found the course content new and difficult: “I think, you know, at first I don’t know what the professor talking about. […] You can, I think you can interview the [other] students, I think they have the same feeling” (Bao Jia 2013). Students therefore identify with one another’s experiences, creating a sense of cooperative “groupness” around the difficulties of learning computer science (Brubaker 2004).

Mutual cooperation and help operated through a system of generalized reciprocity (Sahlins 1972). Generalized reciprocity is the act of giving without expecting an immediate return of equivalent value but with a general expectation that they will get help at some point (Sahlins 1972, 193–94). It was often groups of friends who would help each other, from giving aid for a specific problem to forming study groups and project groups. Yet, students were also willing to help others who are not in their regular grouping of friends. For example, when I thanked one student for taking the time to participate in an interview, he commented that it was no problem and students try to help each other out, seeing me as a fellow student with similar struggles of trying to complete requirements for a degree. As Marcel Mauss (1966) discusses, gift exchange also works as a process of building social relations, and so helping fellow students can be seen as a means of group formation, creating bonds of mutual good will that become friendships, the basis for project groups, or a means of identification with others based on shared struggles and reciprocal exchange of aid. In this way, mutual help and cooperation worked as a mild form of resistance to the overriding theme of independent autonomous learning.

This kind of cooperation was also often encouraged by professors and instituted as part of computer science pedagogy. A first-year professor commented "I really encourage students to study in a group because it helps [them learn]." Similarly, in talking to a professor about the university's programs on entrepreneurship, he explained that one of the key things they try to impress on aspiring entrepreneurs "is we emphasize the value of sharing, you know, you can't keep ideas to yourself, you can't do things without sharing either through founding teams, partnerships, whatever. That's a critical element of what you do." In addition, most courses had online forums where students could post questions or problems and share resources. This was done either through the university's internal online learning system, or sometimes using Facebook groups. Professors would also answer questions and post model solutions in these forums, along with providing news and updates about the course and assignments.

As discussed above, most courses also had TAs who were either undergraduate or graduate students. Junior students often found having officially employed and assigned peer tutors very helpful. Fourth-year student, Shirley, recalled how being able to ask a tutor questions was instrumental to her success in her first programming course and to continuing in computer science.

> You know every computing course, CS course, they have like introduction to programming… the tutors were really really helpful. I think I was really lucky because I have tons of questions always, so I know who to ask and a lot of people were helping me. And I was really really lost in the beginning, but in the end I managed to get an A and I feel maybe I can do it, so I continued in CS (Shirley 2014).

I heard some contention from first-year students over the teaching abilities of some TAs. TAs were given the teaching materials by the course professors for each week, generally in the form of practice problems that students were supposed to try ahead of time, with solutions being given and discussed in the tutorial. However, it was up to TAs to try to construct useful ways to help clarify students' problems and questions, which different TAs were seen to do with varying degrees of effort and effectiveness. In general, however, for students who attended and participated in tutorials or sought help from tutors, they found the system of smaller class-sized tutorials and access to tutors to whom they could pose questions to be helpful.[57] In this sense, cooperative learning was an essential part of learning computer science for many students. It helped them mutually clarify course concepts and build connections with others in computer science.

[57] Tutorials usually had 10 – 20 students whereas first and second-year lectures often had 100 – 200 students.

In Chapter 3 I discussed how, as students are initiated into the *sacra* of computer science, they intra-act with code, programming languages, development environments, and compilers, among other actors, in order to understand and build computing worlds. Along with the many technologies that students learn to use, they also learn to mobilize these programming language documentation, course notes, and lecture material, as well as friends, TAs, other students, and internet resources such as Google and Stack Overflow. These intra-actions connect them to a large network of other human and nonhuman actors. In most cases, professors encouraged students to use whatever resources they had on hand to answer questions or problems. Most midterms and exams (although not all) were open-book. Students could bring text-books, course notes, and a variety of other resources with them to the exam to use and to help them, although their direct contact with other humans was restricted for midterms and exams. As such, solving the problem, whether it is a test question or program design, was paramount.

During one class, the professor had been walking around the lecture hall asking individual students different questions related to the material. I recounted in my fieldnotes the situation that ensued:

> He [the professor] says there are many resources all over the place and he wants students to make use of them, asking them a question and then telling people to go look, and waiting for an answer. He says this will be replicated at home assignments, where students will have to figure things out on their own and then give an answer on the forum. "Do your own research" he says, but now that there are so many resources the problem is to choose a good one.

Students' success in computer science is premised on learning how to "network" – to use a term computer scientists and most readers are likely familiar with – with others (human and nonhuman), a practice that is both about being an independent knowledge-seeking subject and about how to connect with or draw on the expertise of other actors, persons, or resources. Networking as enacted by students and professors in computer science both posit and create a separation between entities. As discussed in the introduction, I do not see such distinctions as predetermined, but as part of the mutual intra-actions among students, technologies, textbooks, etc. However, such boundaries between entities are produced through intra-actions by students and professors with text-books, search engines, fora such as Stack Overflow, and other human and non-human actors, which become constituted as "networking".

Learning to mobilize resources was thus an essential component of independent learning. Shirley, for example, discusses how she does not independently consider herself a good programmer, but rather relies on other sources and resources to learn and develop as a knowledgeable person:

> I've learned from all kinds of sources, mostly stack overflow, so I feel I'm a not very solid programmer. So like everything on stack overflow everybody can go look at it and you know apply to their code and they can find all kinds of sources online. But it's like really really good programmers they might be really old, like before stack overflow ever appeared so read books about how to program, how to do this… I feel like what I know is something on the surface but maybe good software engineers they know how this API even works down one level. Yeah (Shirley 2014).

Shirley talks about how she has learned from various sources online, helping build her knowledge of programming. Yet, compared to "really really good programmers" she feels she has much to learn. Shirley also points to how these programmers "who might be really old" likely mobilized different resources of the time, like books. Nonetheless, Shirley's mobilization of networks of actors – Stack Overflow, for example – connects her to numerous other programmers in different places around the world while simultaneously embedding her in and (re)creating computer science disciplinary networks.

Shirley further explains how her approach to learning and mobilizing resources differs from those of a friend from another faculty:

> Well, I have a couple of friends, like as I told you the PRC friends who I hung out [with] before the school started. Back then we, I feel like we think really same, so whenever we want to do something or we are solving some problems, yeah the thinking process is really similar. But, after these four years we meet again and talk about stuff, it's kind of different. It's, yeah, sometimes it surprised me why they think this way (Shirley 2014).[58]

In particular, Shirley commented on how she and other people studying computing would just Google something – a line of code, or particular command – if they did not know what it was or how it worked. Shirley relayed how their friend in another faculty, however, responded "but it's not written in the lecture notes, can we use them?" Shirley's discussion emphasizes how a focus on independent learning and networking has changed her way of thinking and her sense of self, at least in relation to friends she knew prior to studying computing and who are studying other disciplines. Her assertions may also work performatively to distinguish herself and demonstrate her adoption of the independent networking selfhood – drawing on tools and computing-related resources – compared to her friends who approached problems in a different way. Shirley's "surprise" also normalizes this approach to learning.

As I discuss throughout this chapter and the next, students work to create themselves as "good" computer scientists, key qualities of which are to become

---

[58] PRC refers to the People's Republic of China.

autonomous and independent networkers. Shirley may have forgotten her experiences of "learning to learn," as this approach to learning has become "common sense" and part of her *habitus* (Bourdieu 1977). However, comments from early-year students such as Dinesh, above, show how learning this approach to learning begins as a self-reflexive process, introduced and taught to students through (very explicit) hidden curricula and normalized discourses of computer science, then repeatedly adopted, adapted, and performed.

As students learn to mobilize or intra-act with others, they are also becoming integrated into disciplinary networks constituted by those very same actors (Nespor 1994). As explored in Chapter 4, intra-actions with these actors are spread temporally and geographically, and often include text books, web pages, and course slides, among other media. The way these networks are cut and are solidified as part of Shirley's self and embodied knowledge differs from the networks of the "good software engineers" that "know how this API works down one level" in terms of the depth and breadth of the networks, and the specific intra-actions entailed. That is, learning different programming languages and environments, tools, libraries, in different places, and with different people all trace different networks of intra-actions. These differences also contribute to (re)producing the directional frictions discussed in the preceding chapter. At the same time, taken together, the multiple networks of intra-actions trace and constitute the discipline(s) of computer science.[59] As students extend and deepen their intra-actions (or

[59] As suggested in the introduction, there are debates as to how to define and bound computer science as a discipline. Similarly, following Annemarie Mol's (2003) analysis of the multiplicity of atherosclerosis, there are multiple definitions and experiences of computer science as a discipline, but which are made

extend and deepen their networks), they become more integrated into the discipline of computer science and more knowledgeable about computer science as a discipline. Thus, while students are treated as autonomous and independent persons and learners, that learning is premised upon their connection with others.

At the same time, persons need to retain their *individuality*. That is, it was still seen as necessary that students could be assessed and judged as individuals. And so, practices such as cheating, relying too extensively on other people for work, or receiving direct help from others in exams were all actively discouraged. One professor explained to me, for example, that assignments were given low weightage in early year courses because they could not prevent students from copying from others or getting friends to do the work for them. At the start an exam, direct cooperation was blocked and cuts made in terms of students' extended networks of peers, online fora, and other forms of intra-action through which they learned, practiced, and became good at computer science. Students then were made to display their individual knowledge and ability, insofar as it could be measured by the exam. The effort an *individual* put into the course was thus assessed in practical exams, which were given much higher weight in course grades. Thus, while the focus on cooperation was an integral part of independent learning and networking, the individual autonomy of student selfhoods is a necessary component of both practices. Cooperation operates as a form of instrumentalized interaction constituted as "networking," which entails a mobilization of resources as students intra-act with

---

coherent and singular through disciplinary and educational norms such as text books, transnational traditions, ACM curricular guidelines, accreditation practices, and so on.

computing knowledges and technologies, educational resources such as text-books, and other students and professors.

5.3 Critical Judgment and Rendering Comparable

Independent learning and networking in computer science is additionally tied to an emphasis on and cultivation of a form of critical judgment. Students were told to "take nothing for granted." In one case during class when there was an error on the course notes, the professor simply responded "don't always believe people are right." Students learned to assess, evaluate, and judge both knowledge and persons according to a wide variety of criteria, such as their accuracy, veracity, and efficiency. Professors sometimes explicitly drew a contrast between memorization and the type of learning they wanted to encourage. In a third-year course, for example, one professor explained he was trying to teach students "better learning": not to learn things by heart but to think about what they are doing. Another professor contrasted this cultivation of independence and critical thinking to past hierarchies in academia. He said that teachers in the past would impose their authority on the students, but said that they (the students) are smart and so he wants them find out for themselves. Another professor for a third-year course simply suggested to students in several instances "if you don't trust me, go back home and try." Critical judgment here relies strongly on independent networking, where if students question the accuracy of something, they should also seek out the "correct" answers or solutions.

This emphasis on critical and independent learning provides an apparent contrast and conflict with the importance placed on learning and following rules – of program syntax and logic and assignment specifications, for example – discussed in Chapter 3. On the surface, it seems contradictory to suggest that students need to learn how to seek out new knowledge and question the accuracy and intentions of their professors while also following the strict and naturalized rules dictated by compilers and course auto-markers. There are, however, a couple of ways this tension is managed and used. First, it can be a productive tension for computer scientists. I have discussed elsewhere how learning to work creatively within sets of rules can provide an arena of play for programmers, simultaneously producing feelings of fun, excitement, and challenge (Breslin 2013). This is particularly true for hackers, who often focus on playing with and stretching the boundary between following necessary rules and inventing creative ways to surpass or extend them. Second, I discuss in the next chapter how student hackers pursue and display their passion for computing by learning in-depth technical knowledge, performing independent learning and critical judgement through learning and exploration within computing worlds and not necessarily (although sometimes) at their boundaries. This tension, however, also reveals a great deal about just what is meant by "critical judgment" as it is cultivated among students in computer science at Temasek University.

Critical judgment as it is figured in computer science relies strongly on constructing a world wherein truth, accuracy, and efficiency, among other forms of measurement, can be reliably known, assessed, and judged. Practices of rendering technical explored in the preceding chapters are at the heart of making reality knowable to computer scientists. This purified technical reality provides a realm of possibility, and even hope, which can be manipulated to help solve social problems, and "change the world," as Xiaowen suggested was his goal for the future. It also provides a reality wherein they can measure and assess the value of code and programs. Critical judgment in computer science is therefore less about questioning the rules and foundation of computer science knowledge, the relationship between technologies and computer science and society, politics, culture, and gender, or the historical and geographical power relationships that make up computer science as a discipline. Rather, critical judgment is about assessing and creating the best solution for a problem.

Despite efforts like TSLN, Aaron Koh (2004) suggests that the Singaporean education system's approach to critical judgment similarly centres largely on "the 'how tos' of solving problems framed by procedural skills such as analysing, synthesising, extrapolating, evaluating, and so on" as opposed to "the ability to critique the kind of symbolic systems and semiotic meanings that are inherent in the new semiotic economy… questioning the given, not just solving problems" (Koh 2004, 339). Given the way that computer science relies on following rules, this particular problem solving approach to critical thinking and judgment thus dovetails closely with computer science's implicit curriculum. By rendering technical, computer scientists render a world that they can work within – a world wherein reality can be reliably known, assessed, and worked on. Judgments are made, then, about that technical reality, which is purified from social context. I consider here how students learn, in detail, ways of assessing the value of code and the value of persons. Students are thus rendered technical as they are judged and assessed on their ability to write "good" code, further rendered into a numerical assessment in the form of grades.

5.3.1 Judging Code

Students were taught numerous ways of judging code, proofs, or problem solutions. In several of the first- and second-year classes and tutorials I observed, students were asked to share their solutions to exercises with the rest of class, either by displaying them on the projected screen or writing their (hopefully short) solutions on the white board. The rest of the class was then generally asked to comment on what was displayed. Through this practice, students learned from others' mistakes or differences, but also how to judge the way code was written according to various criteria, which changed and expanded as students' progressed through their programs.

In first-year, displayed programs were meant to show *how* to solve a problem, when students were struggling just to learn the basic syntax and construction of a program. Course notes were also replete with examples, which demonstrated to students how a program should be written, how it should look, as well as the programming techniques needed to achieve particular functionality. This practice is seen in Figure 5-3, for example, which was taken from first-year course slides for the second week of class. It shows the basic elements and structure of a program. Most of the program elements pointed to in Figure 5-3 are necessary syntactically, but the example also includes comments, proper indentation, and particular variable naming conventions, which all contribute to making "good" program. In later years, students learned how to judge algorithms based on other criteria, namely their efficiency in terms of speed and space in various cases (average, best, and worst). Other algorithms that did not give simply right or wrong answers were also judged on their accuracy. These judgements relied on measurements that provide a means of "objective" comparison. At the same time, students needed to learn to evaluate trade-offs, sometimes making an algorithm more efficient in terms of speed at the cost of using up a great deal of space in computer memory or sacrificing the accuracy of results. Students were thereby meant to judge in a reasoned way.

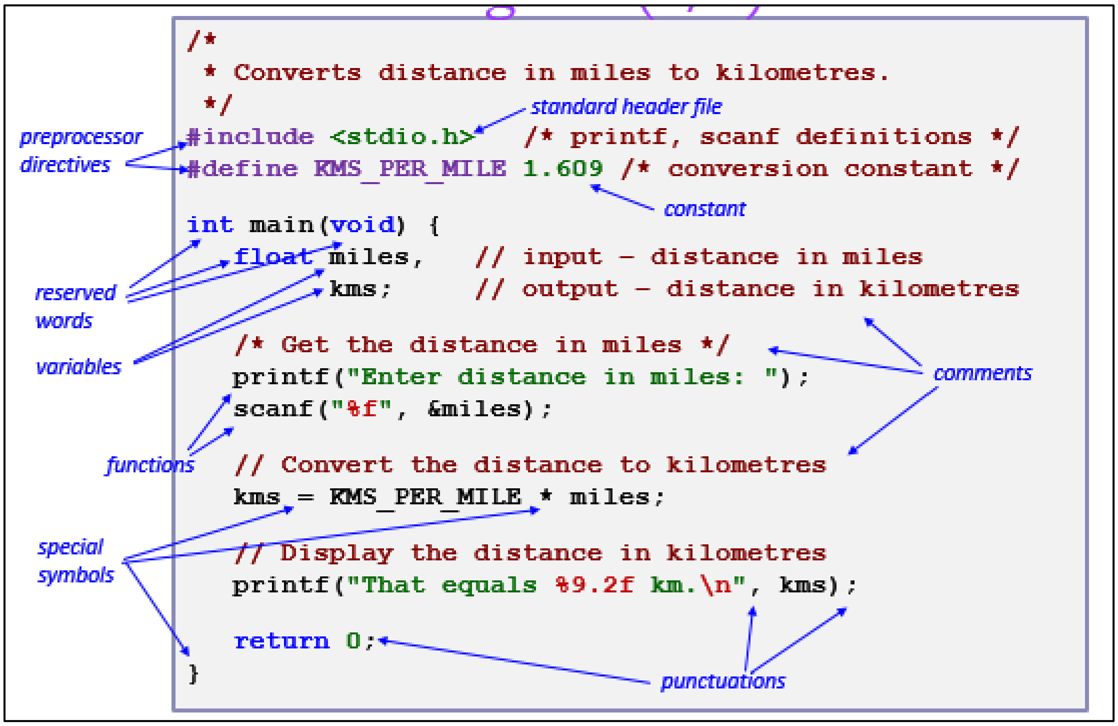

Figure 5-3: Example of a "good" program

Students were also taught to judge the aesthetics of a program, even from their first-year. By week 4 of their first-year students were being told about “acceptable” and “not acceptable” indentation styles, exemplified in Figure 5-3 and later discussed explicitly, as seen in Figure 5-4. The latter was reproduced from first-year course notes and shows proper formatting for *if* statements – a very basic programming structure that tells a computer to do “statements” “if” a certain “cond” (condition) is met and “else” (otherwise) do the other “statements.” The differences between the various styles are subtle, showing the level of detail students must attend to when learning programming.

There are functional reasons related to human readability for why certain styles are not acceptable. The first example is unacceptable because, without indentation, it becomes difficult to follow the structure of a program. Additionally, as mentioned earlier, in some languages indentation is meaningful and so improper indentation will cause a compilation error. The second is unacceptable because with the closing bracket ("}") on the same line as the statements, it becomes difficult to see if all the conditions have been properly closed. Missing brackets is a very common error in programming that will also cause a compilation error. Yet, to the computer, in the language C, all of these statements are functionally equivalent.

Acceptable Indentation Styles

```
if (cond) {
        statements;
}
else {
        statements;
}
```

```
if (cond)
{
        statements;
}
else
{
        statements;
}
```

```
if (cond) {
        statements;
} else {
        statements;
}
```

Not Acceptable Indentation Styles

```
if (cond)
{
statements;
}
else
{
statements;
}
```

```
if (cond) {
        statements; }
else {
        statements; }
```

Figure 5-4: Examples of acceptable and not acceptable indentation styles

Students thereby learn to judge code on its readability and aesthetics as much as whether it successfully achieves its desired function. Other facets of programming aesthetic that I heard discussed in various classes include: how to name different variables, such as using various forms of capitalization or appending “ptr” at the end of a variable to indicate it was a specific type of variable (a pointer); different syntax for declaring variables, which essentially equates to creating a variable of a particular type (an integer, a character, a string of characters etc.); naming and appropriate use of functions; ways of structuring a program; appropriate length of programs, functions, statements, and so on; and including comments to explain the functionality of code for other programmers. This is far from an exhaustive list, but provides some indication in the variability and scope of decisions that need to be made when writing and judging code. The situation becomes even more complex when a small snippet of code becomes a large program with hundreds or thousands of files and lines. Students must then also learn “good” ways of designing a large program.

In most, if not all cases, there are multiple “correct” ways to structure, write, or design a program, entailing an element of personal preference, along with assessment of the advantages and disadvantages of different approaches. After teaching a design concept to students and asking them whether they would apply it to their course projects, for example, one professor asked them to question whether there was value in doing so. He commented, “all of them come with baggage, cost… When you have a hammer in your hand, everything is like a nail… always think about: is the benefit worth the cost?” In this case, he also suggested that there was value in terms of the learning that students could gain if they did decide to use and apply this design. With these various forms of etiquette around coding, the intractable rules discussed in the previous chapters become much murkier as students are writing not just for the computer (the compiler and the auto-marker), but for measurements of efficiency in relation to speed and space, for example, as well as for other persons (friends, colleagues, professors, TAs). And so students must learn to evaluate and judge the correctness, validity, and efficiency, as well as the aesthetics, style, and design of code. At the same time, the realm of judgment never escapes the bounds of assessment and evaluation based on technical measurements.

A small number of computer science courses at Temasek University focus on human computer interaction (HCI) and explore the interface between humans and computers and practices for making technologies more usable and creating better user experiences, although there are many possible ways that "better" can be constituted. By comparison, the vast number and diversity of courses on topics such as programming languages, computer networks, computational biology, parallel and distributed computing, testing, and computational theory, demonstrate the dominant focus on technical education situated squarely within computing worlds. Scholars working at the intersection of STS and HCI have suggested alternative approaches to assessing technologies. Lucy Suchman, for example, argues that HCI work is not about the creation of objectively "good" products, but rather "located accountability" whereby "design success rests on the extent and efficacy of one's analysis of specific environments of devices and working practices, finding a place for one's technology within them" (Suchman 1994, 99). This alternative perspective, focusing on evaluating technology in context, highlights the ways that dominant forms of assessment in computing rely on measurement and technical renderings – evaluating trade-offs among technical measurements related to functionality, such as speed, space, or accuracy – even if the rules and etiquette are sometimes murky.

The murkiness of judgment versus the absolute correctness (or incorrectness) of following rules presents a tension, however. It is in the space between absolute correctness and aesthetic and design that computer-scientists learn to display their creativity, individuality, and skill. For example, in discussing with Xiaowen what makes a good computer scientist, he explained:

> **Xiaowen**: Just spend a lot of time thinking, but also you cannot just think, you must actually execute it and try it. Yeah.
> **Sam**: So it's not enough to design an algorithm, you have to make sure it works.
> **X**: Yeah. I guess mathematically you could prove that it works, but computing is beyond making it work. Working is like just the first step. You make it work, then you make it right, then you make it fast. (Xiaowen 2014)

Functionality is the first measure of good code; if it does not compile, does not run, does not do what is expected or desired, then it is not good code. But as Xiaowen points out, it goes far beyond this simple measure of correctness.

Particularly elegant solutions were also admired and praised. Elegance could have a variety of meanings, including code being written with very few commands or lines of code, code that runs very efficiently, or a clever solution to a problem. One professor, for example, in teaching a new algorithm to students emphasized its beauty, commenting "Come on computer science has a lot of beauty, you cannot see the beauty?" Another professor suggested that code started out like rough diamonds, and so students want it to be polished and beautiful at the end. Yet another observed to students, referring to a particular algorithm, “there’s beauty in the algorithm, every day I look at the algorithm I’m amazed by its beauty.” Professors in upper year courses seemed more expressive in this regard, as they could show students new realms of possibility and beauty as the algorithms and structures became more complex and intricate, yet still often simple and elegant. These professors were trying to instill a sense of wonder in students at the power, possibility, as well as beauty, of computer science and of programming. Yet, they are also teaching students to recognize what constitutes “good” code when it is already correct – what makes some code, some algorithms, some programs better than others. For example, in showing students how to shorten and combine multiple if statements, the professor commented that the shorter version was “prettier.”

Discussions over beautiful and elegant code are pervasive in computer science and programming discourse, again seen in blogs, news articles, and academic papers. They are also associated with debates about whether computer science is an “art” or a “science.” In his lecture for the Turing Award, for example, David Knuth declared that “when we prepare a program, it can be like poetry or music . . . programming can give us both intellectual and emotional satisfaction, because it is a real achievement to master complexity and to establish a system of consistent rules” (Knuth 1974, 670). He goes on to claim that programs can be “elegant,” “sparkling,” “noble,” and “truly magnificent” (Knuth 1974, 670). Programmers have even expanded the use of programming languages beyond creating function-oriented programs to write poetry in code and pseudocode, showing the breadth of creativity and expressiveness some computer scientists and programmers can find in code (Berry 2011, 46–51; Breslin 2013; Coleman 2013, 93–122;

Hopkins 2001). At the same time, measurements are made within a particular self-contained ethic and aesthetic. The context of an algorithm or program, and the kind of located accountability suggested by Suchman as a measure of goodness, is erased in Xiaowen's analysis of making good code, and in assessments of code's elegance and beauty.

5.3.2 Judging Persons

The expressivity of code is further tied to the expressivity of (independent networking) persons. As Coleman suggests of hackers: "The logic among hackers goes that if one can create beauty, originality, or solve a problem within the shackles of constraints, this must prove a *superior* form of creativity, intelligence, and individuality…" (Coleman 2013, 118). Coleman's discussion relates to ideas of authorship and freedom among hackers, yet it also shows the intra-actions that produce a clear relationship between creating code and creating/demonstrating personhood. Rebecca Bryant elaborates that "learning to become the type of person who can do X... implies that (1) practice and personhood are inextricable; (2) practice requires becoming a person embedded in a hierarchy of values and capable of judgments; and (3) acquiring that hierarchy of values means becoming capable of making judgments that link being good with being good at" (Bryant 2005, 233). In learning computer science, students learn a "hierarchy of values" for judging code, which then entails making judgements about "good" code, programs, or algorithms in relation to judgments about persons who are "good at" computer science.

During one of their first classes, first-year students (and I) were given a handout entitled “My Progress Chart.” This was a single page back and front, which was also part of a handbook for first-year students. It asked students to evaluate their understanding of the course each week, as well as their success in various forms of assessment and class exercises. It also had space for them to comment on what actions they would take in response to these results and, more generally, whether they were meeting their own expectations and targets. Pedagogically such a chart is understandable: by asking students to reflect on their progress throughout the course, first-year instructors hope students will recognize when they are having difficulty and seek help. One first-year professor I spoke with was effusive about how he very much hoped and sought to help first-year students who were struggling, but added with much regret that it was difficult because students do not always want the extra aid that he offered. This self-evaluation is thereby meant to help promote an active engagement by students in their learning process and, in the spirit of independent learning and networking discussed above, take action if they are not meeting expectations (their own or others’). Yet, this chart is also represents a form of self-discipline, encouraging students to constantly monitor, judge, and remake themselves based on their success at programming, their progress in the course, their grades, as well as by their weaknesses or gaps in knowledge and skill. Being a critical and judging subject therefore also means learning how to judge what and who is “good.”

Similarly students learn to judge others in relation to the code they create. As suggested above, code comes to stand in for and be exemplary of a person’s skill and ability. Dawn Nafus found a similar practice among Free/Libre and Open Source Software (FLOSS) developers, who argue that code ought to “speak for itself” in that “technological skill ought to be evident in the work itself” (Nafus 2011, 677). Nafus argues, however, that this ideal is not borne out in practice, since the value of code was often significantly determined by how strongly people promoted and defended its “goodness,” “through highly masculinized, aggressive online talking” (Nafus 2011, 679). Aggressive masculine performance was not nearly so evident among students in Singapore, although I discuss in Chapter 7 the ways judgements of skill relate to gender. Yet, in many ways, students were continually judged based on their code.

For one of the courses I observed, all of the labs were based around students displaying and explaining their programs to the rest of the class, where students in the audience were also given marks for providing them with feedback. The professor suggested the goal was for students to have their solution vetted by multiple people. Yet, as students practiced judging code, they also displayed their knowledge and skill as programmers, both in their solutions and in their comments. Students thereby learned to practise and display critical judgment as part of the embodied practice of becoming the type of person who does computer science. Those who wrote good code – correct, efficient, elegant, short, or clever, among other measurements – were thereby seen as good at programming. Additionally, the “best” solution for the week’s lab was posted in the forum, with the student’s name included. In this way, students are summoned to work on themselves and their code.

These practices of critical judgment also rely on and foster reality – and persons – in ways that *can* be evaluated and assessed for their “goodness,” and thus closely intertwine with practices of measurement, assessment, and evaluation. Grading practices, in particular, work to render technical students' personhoods so that they can be judged, assessed, and compared. In discussing the classification of engineering students in relation to their "success" and "progress" through a program, O'Connor, Peck, and Cafarella discuss how this classification is achieved through testing and grading, and "taken together, these practices serve to translate students to marks on paper" (O'Connor, Peck, and Cafarella 2015, 176). Following Latour, this process translates students and their ever-changing lives and identities as "persons-in-motion" – persons continually mobilizing (and being mobilized into) networks of human and nonhuman actors – into immutable mobiles (Latour 1987; O'Connor, Peck, and Cafarella 2015, 177). Students' lives, selves, effort, and perseverance are rendered technical, represented in numbers (or letters), although this representation is not necessarily faithful to students' efforts, affective experiences, and future aspirations.

Grades held great significance for students, both because of the influence they have for students' current and future academic and career paths, and because they become intertwined with students' self-understandings. First-year student Paul, for example, explained to me how grades were important for him and others:

> We are very concerned about how do we score for this module… I think students are very concerned about their grades. If they want to explore they will definitely do it on their own. I think that's the Asian mindset, because we've been, like I say the education system is very rigid. Yeah, we've just always been doing, even in primary school science, maths. Like things we might not be interested in but we are forced to score. So yeah, that's been the mindset throughout. And then even the university system it goes that way, if you're not very interested in whatever he's [the professor's] talking about, you should just focus on how to score (Paul 2014).

Paul explains how he and others have been focused throughout their education on grades, because of the "rigid" Singaporean education system that filters them through necessary courses whereby they are evaluated in order to direct their future opportunities and paths.

Paul further explained how he came to study computer science: "Actually, to be honest my results wasn't that great, so it ruled out all the faculties like medicine and history, all this kind of stuff. So I look at my next A-level choice and it was a decision between science and computer science." After exploring the possibilities within science and computer science, Paul decided that the latter was more interesting to him because of the possibility to specialize in security and networks. Other students and professionals in Singapore I spoke with explained that, similarly, while they chose to study computer science, this choice was also conditioned by their grades. Alternatively, some students used their scores on their A-levels as a guide to what they should study based on where they were most successful at the secondary level and in exams. Students' sense of self is then intimately tied to the measures they are given in the form of grades.

Additionally, as mentioned above, students' grades for most courses are modified to fit a bell-curve (also known as a Normal distribution, or a Gaussian distribution). The bell-curve is the epitome of practices of rendering technical. In exploring the development and spread of statistics and statistical governance, Ian Hacking discusses how, through the use of the Normal distribution to model and assess populations and persons, "normal" came to describe not just how things are but also how they should be: both fact and value (Hacking 1990, 163). Regardless of efforts and abilities of professors in teaching and students' own efforts and agency in learning, students' collective grades are meant to follow a "normal" distribution. This also means that, in the end, a person's grade – meaning their success or failure – is dependent not only on their own efforts and inputs, but also those of every other student in their class.

Timothy Mitchell (2002) further explores how various forms of mapping and modelling were used by colonial governments in Egypt to make remote places and people knowable and legible. Hacking also discusses how statistics emerged with the growth of state bureaucracies as they tried to control and improve their populations (1990). Grading and bell-curves similarly work to render students knowable, legible, and controllable. One of the justifications given by the Provost at Temasek University, for example, was "to achieve consistency" in grading across courses, professors, classes, and years; students are thus rendered into numbers that can be compared, judged, and hierarchized.

The centrality of testing and grading again highlights how *individual* independent learning and critical judgment is prioritized and promoted. This holds true even if its significance is diminished for students with a gradeless first-year, and even when exams are open-book and allow students to mobilize their networks of aides. Students' ability to progress through their program and earn their degrees is ultimately premised on earning high enough grades to pass their courses and aggregate to a high enough CAP. These grades create comparable rankings, which are attributed to and taken as representative of the degree to which an individual person is good at computer science. Some courses also maintained leaderboards or scoresheets that ranked students across their assignments and other class activities, shared publicly and including students' names. Students' degrees were similarly ranked, although according to much broader classifications, between Honours, divided into Highest Distinction, Distinction, Merit, or no extra designation, and then Pass or Fail. These calculations, and the comparison of students more broadly, also rely on the exclusion of context (as with the judgement of code), and summon students to compete with one another for grades and opportunities, such as exchanges abroad, scholarships, and jobs and careers.

5.4 (Trans)national Competition

I consider here how, despite the ethos of cooperativity and solidarity discussed above, whether they wanted to or not, many students also keenly felt they were necessarily competing with others. Moreover, technological development in Singapore starting in the 1980s, if not before, has drawn extensively foreign expertise. I show how government policies relating to these “foreign talent,” along with the transnational movements of computer science knowledge, education, and expertise, summons students to compete with one another and with (sometimes imagined) others around the world.

As discussed in Chapter 1, Singapore’s highly successful development since its independence has focused on openness to multinational corporations (Gopinathan 2001, 23). Favourable taxes, special economic zones, a stable labour environment, “modern” and reliable infrastructure, and standard usage of English in education and business have all been coordinated by the Singaporean government to foster a welcoming environment for international business, finance, and expertise. In relation to technological development, foreign experts were seen as necessary to compensate for the early lack of local expertise in sufficient numbers, in relation to both professionals and educators. As part of the national computerization program in the 1980s, for example, the National Computer Board suggested that “overseas expertise should be brought in, either through recruitment of foreign talent or by engaging overseas companies to develop it jointly with local companies" to help with establishing and upholding professional standards (The Straits Times 1981d). An early news commentary similarly suggested, "there is no magic wand to wave around; expertise has to be acquired the hard way through training and education. In the meantime, Singapore would have to rely on foreign expertise as a stop-gap measure" (The Straits Times 1981b).

While the government has sought to produce a thriving local industry, foreign experts and foreign investment are also seen as a necessary component to achieving the government's policy goals, such as becoming "#1 in the world in harnessing infocomm to add value to the economy and society" (iN2015 Steering Committee 2006, 15) and then to become the "world's first Smart Nation" (IMMSC 2015a, 2). In this regard, the Intelligent Nation 2015 report, for example, explains:

> Enterprises will need a special breed of infocomm professionals – "techno-strategists" who have a strong grasp and understanding of infocomm technologies as well as business knowledge of various industries. Singapore will work to develop such expertise, continue to attract them, as well as build an environment that facilitates the country's access to global infocomm talent. Singapore will also attract aspiring foreign technopreneurs to use the island as a development and engineering centre for their business ventures, and as an operations hub for penetrating international markets (iN2015 Steering Committee 2006, 31).

The report later explains the necessity of these foreign "technopreneurs" as "crosspollination of knowledge and ideas between local and foreign technopreneurs will promote diversity in technology capabilities and spur innovation by the local industry" (iN2015 Steering Committee 2006, 60). As Aihwa Ong suggests of expatriates in "mega-cities" like Singapore, "the expatriate community keeps the city in the global game. One may say that a megacity is in part defined by the number of mobile professionals and entrepreneurs who make it a long-term stop in their international itineraries" (Ong 2007, 86). Foreign experts are thought to bring new and creative ideas, adding essential ingredients to the local tech ecosystem.

These policies have led to a perceived favoritism towards foreigners by the government, which became a contentious political issue in 2013 when the Singaporean government published the report "A Sustainable Population for a Dynamic Singapore: Population White Paper." The White Paper was meant to set the direction for the government's population policies as the city state grappled with a declining birth rate, aging population, and continued need for a viable workforce. While the report first lays out the importance of Singaporeans as the "core of our society and the *heart* of our nation" (National Population and Talent Division 2013, 14), much of the report also attends to the need to attract more and new immigrants, residents, and citizens to build and maintain a "a globally competitive and vibrant economy, one which sustains inclusive growth, creates opportunities and good jobs for our people, and helps Singaporeans achieve their aspirations" (National Population and Talent Division 2013, 31). Overall population growth and especially the number and proportion of new foreigners in Singapore that the government was suggesting as part of that growth was the primary point of contention for Singaporeans.

The report stated that the current population of Singapore as of June 2012 was 5.31 million, comprised of 3.82 million residents (citizens and permanent residents) and 1.49 million non-resident (students, domestic workers, foreign workers, and foreign talents, among others). The government projected a total increase in population to 5.8 to

6 million by 2020 comprised of 4 to 4.1 million residents and 1.8 to 1.9 million non-residents. By 2030, the government projected a population of 6.5 to 6.9 million made up of 4.2 to 4.4 million residents and 2.3 to 2.5 million non-residents (National Population and Talent Division 2013, 46–48). These "possible population trajectories" were being set to "meet the present and future needs of Singapore and Singaporeans" (National Population and Talent Division 2013, 49) and "address our demographic challenge" (National Population and Talent Division 2013, 66). The projection or trajectory means an increase of 1.2 to 1.6 million people over twenty years on an island that is only 720 km^2 (projected to increase to 766km^2 by 2030) (MND 2013, 4).[60] More significantly, 0.81 to 1.1 million (approximately 67%) of that increase is to be non-residents or foreigners, which does not include the number of foreign-born migrants to be given permanent-resident status.

The White Paper was met with resistance and protest by citizens, a rare occurrence in Singapore. Protests at Hong Lim Park – the only place in Singapore where citizens can legally publicly protest – in February 2013 after the paper was released included several thousand people, along with speakers from current and former opposition party members (Goh and Mokhtar 2013). There were also debates in the media about the issue (e.g. Leonard and Ong 2013). In response, in September 2013 the government announced changes, which came into effect in August 2014, that require employers to consider Singaporeans before hiring those who would require an

[60] Through land reclamation projects, the government has been working to increase the land area of the city-state. The land area of Singapore has increased from 581.6 km^2 in 1960 to 719.7 km^2 in 2016 (Singapore Land Authority 2017).

employment pass. The government also developed a job bank where new positions must be advertised for at least 14 days, and increased the minimum qualifying salary for new Employment Pass applications from $3000 to $3,300 per month (Teh 2013). At the same time, Acting Minister for Manpower Tan Chuan-Jin argued that "What the government is doing is to help them get a fair opportunity. Singaporeans must still prove themselves able and competitive to take on the higher jobs that they aspire to" (Ministry of Manpower 2013).

The policies discussed throughout this chapter act as "instruments of power for shaping individuals" (Shore and Wright 1997, 4). Students felt they needed to cultivate themselves to compete with these "foreign talents" in Singapore, as well as other programmers and computer scientists around the world. As second-year student Qiaohui explained:

> **Sam**: Do you find it very competitive?
> **Qiaohui**: Oh yeah, definitely it's very competitive. I don't know about other faculties but here in computer science… it's very competitive.
> **S**: Too competitive, or?
> **Q**: Um, I shouldn't say too competitive. Hm, I think it all depends on your own ambitions.
> **S**: So for you, anyways, not too competitive?
> **Q**: Not that competitive. But still, at least you need to keep up with the rest of the class because of the bell curve. If you just go at your own pace, you're going to fall behind for sure. (Qiaohui 2013)

This competition was also keenly felt in relation to students and professionals from other places. Aindri, a third-year Singaporean student, for example, explained:

> It's really competitive… For high marks, for everything. Because, the way that we discuss it, me and my friends, is that, you are a local student, right, so maybe in your college or in your class you were average or maybe you were in high average. But when you come in to [Temasek] you're actually competing right, at your level, with students who are at the best in their levels from other countries,

> which makes it even more worse. Yeah. And, this race for marks or the bell curve, you know it makes it harder, yeah definitely (Aindri 2013).

When I later asked specifically about government policies on foreign talent, she elaborated:

> I think it's, like I have nothing against it, I think it's like really good. But, of course it places much more pressure on those who are already here, which makes it even more stressful, right? Like everyone, it's in human nature to go to a place which gives you more opportunities or which will give you a chance to make your life better, economically or in whatever aspect. Right. So, I think it's definitely good, and they should. But from my perspective, it places more pressure on me, right so it makes it more stressful. So, in that way I have to up myself even more (Aindri 2013).

Singaporean students like Aindri argued that students from overseas started in computer science with more programming experience, putting them at a disadvantage in classes, particularly with the emphasis on competition and comparison through the bell-curve. In this regard, Aihwa Ong (2005) argues that these policies and practices have privileged foreign talent at the expense of local expertise: "Expatriates bearing intellectual capital – entrepreneurs, scientists, computer programmers – come to be inscribed with new values of citizenship, while citizens are found ethically (and ethnically?) deficient" (Ong 2005, 349). In other words, these policies have created an implicit, but undefined, standard against which students in Singapore are judged, although students from elsewhere did not always see themselves as the already virtuosic programmers that they were sometimes perceived to be.

In addition to this competition in Singapore, there was a sense among students that computer science education, the basic overt curriculum and courses, provides a foundation, but one that is largely equivalent to and interchangeable with that of every other reputable computer science program around the world. This is likely a desirable goal for educators and industry. For example, when the NCB organized a joint certification program with the British Computer Society in 1982, the goal was to create a national standard for computer professionals and ensure that "professional competence in Singapore will be relevant to the rest of the world" (The Straits Times 1983). For students, however, it was a cause for concern. The curriculum alone then provided no way for students to distinguish themselves from the multitudes of other programmers and computer scientists globally.

For example, when I asked second-year student, Xiaowen, if he thought the curriculum prepared him to work anywhere in the world, he responded:

> Yeah, I think, by itself the curriculum doesn't prepare you much, but there are other opportunities that you can take that will prepare you to work. Like exchange programs, internships, and whatever. Things that expose you. Yeah, so by itself the curriculum is standard. I think any university would provide you the curriculum, it's just technical, especially for computer science (Xiaowen 2014).

Another student explained to me one of the reasons she switched from computer science to a program more oriented towards business or management:

> I wanted to go over to project management, yep. Which I think is also not a bad thing lah, in Singapore, because for technical cert[ification] I'm thinking that there are a lot of talent out [there], but then project manager is something else. It's not like, you know like programmers you can always like go to India and outsource it to others (Liz 2013).

Most, if not all, students were aware of government practices and policies on funding and/or hiring foreign students and professionals. Similarly, the US tech industry is well-known for outsourcing projects to companies elsewhere and for hiring expendable labour from elsewhere (see, for example, Biao 2006; O'Donnell 2014). Thus, as students compare themselves, and are compared to one another, within the university, nation, and ultimately across the world – based on their coding, their grades, and their degrees – they are summoned to render themselves in ways that improve these measurements and become competitive individuals.

5.5 Conclusion

The hidden curriculum of computer science at Temasek University cultivates students as independent and networking learners. They learn to (re)make themselves as autonomous and creative individuals who are able to mobilize a variety of human and nonhuman actors. Students also learn to critically judge, measure, evaluate, and assess code, algorithms, programs, and persons according to a wide variety of criteria. These criteria, however, are situated within the context of reality rendered technical: the messiness of culture, embodiment, and politics are purified into a technical form that *can* then be measured and evaluated. For computer scientists this purification renders a world of power and possibility wherein reality can be known and manipulated. Many computer science students thereby hope to change the world for the better.

Yet, just as reality is rendered into technical form, so are students' persons as their academic lives and selves are inscribed into grades and aggregated into their CAP, to provide a single numerical measurement of their personhoods. This measurement makes students comparable and summons students to compete with one another. Policies that privilege foreign talent and the efforts to create standardized curricula also foster a sense of transnational competition with other students, programmers, and computer scientists (trans)nationally. According to Singapore's discourses of meritocracy, students' success and grades should correlate to their efforts and talent. While the repeated measuring and standardization of students runs contrary in many ways to current policy efforts to change the national habitus and create flexible, entrepreneurial, and risk-taking citizens, they also ultimately promote an individual, autonomous, and competitive selfhood that simultaneously feeds into neoliberal discourses, which I explore further in the next chapter.

Chapter 6 : Neoliberal Passion and Entrepreneurialism

My first introduction to the promotion of hacking and entrepreneurship in Singapore and at Temasek University was through the student hacker group's "Welcome Tea." Most co-curricular groups held such welcome teas at the beginning of term to introduce the group, their goals and purpose, and invite new members to join or participate in the future. For this welcome tea, the organizers gave a presentation about "hacker culture," explaining what it is and how and why they wanted to develop and grow it among students at the university. The group has a broad but particular vision of the culture they aim to promote. They explained how they explicitly distinguish hackers from "crackers," which refer to the more nefarious and illegal connotations and activities of the former term related to breaking various forms digital security. Instead, they drew inspiration from a variety of figures involved in free and open source software development and entrepreneurship, largely from the US. They referred directly to figures such as Richard Stallman (generally seen as the "father" of Free Software), Dennis Ritchie and Ken Thompson (creators of the C programming language, discussed in Chapter 3), and Mark Zuckerberg (the founder of Facebook), among others.

The students promoted a hacker culture based around values such as passion, "playful cleverness" (a term used by members of the student hacker group), and technological skill. This involves exploring the limits of what is possible – a trait they attribute to building the Internet and UNIX, among other innovative achievements – as well as being part of a community with like-minded people. The student group organized workshops, weekly talks, and a hackathon, dedicated to teaching particular skills not part of the academic curriculum, focusing on Open Source and Web development in particular, and promoting passion, interest, dedication to technology, technical skill, and innovation in general.

This chapter explores the promotion of this hacker culture – particularly as it pertains to the promotion of passion – by students, professors, and the Singaporean tech community, as well as by Temasek University and the Singaporean government. This discussion provides a case study of one of the particular ways that processes of rendering students comparable and competitive plays out in computer science. I discuss, in particular, the meaning and significance of passion and how the members of the student hacker group come to exemplify what it means to be a passionate – and thereby good – computer scientist. As such, these students become models for comparison and competition for others. I also discuss how this promotion of passion relates to national policies entrepreneurship, which reproduces a neoliberal form of affect and of work. In this regard, the cultivation of competition among students, discussed in the preceding chapter, connects to (trans)national processes of neoliberalization.

6.1 Building a "Hacker Culture"

The term "hacker" has many meanings. Two of these were discussed in the welcome tea, namely the distinction between "crackers," on the one hand, and the mix of entrepreneurship, tinkering, and technical skill that the hacker group enacts and promotes on the other. The role of "hackers," and the meaning of the term, has been contested since at least the 1970s when they were alternatingly derided as "computer bums" or hailed as "revolutionaries" (Ensmenger 2015, 38–40). For the former, Sherry Turkle explores, for example, how computer scientists and hackers at MIT adopted and even celebrated their identity as "archetypical nerds, loners, and losers," embodying a masculine avoidance of emotion and social interaction, and seeking instead intimate relations with machines (Turkle 2005, 183–203). Alternately, those same hackers were celebrated as innovative "wizards" and "heroes" who were positively reconfiguring and revolutionizing social relations through their use and creation of technology (Levy 2010; Turner 2006, 116–17).

As discussed in the preceding chapter, Gabriella Coleman and Alex Golub have explored how hackers (largely based in the US) adhere to and promote a particular form of liberalism based on freedom and expressivity (Coleman 2013; Coleman and Golub 2008). They further discuss three different versions of "freedom" practiced by varying groups of hackers: (1) "cryptofreedom," working either on creating new forms of cyber-security or on ensuring individual privacy; (2) the free and open source software movement focusing on access, openness, and sharing of source code; and (3) radical individualist freedom focusing on breaking or "cracking" rules and barriers for the sake of fun and pleasure (Coleman and Golub 2008, 259–66). The student hacker group explicitly distances itself from the last category and its illegality, but varyingly enacts the first two forms of hacker practice, in addition to adopting entrepreneurial values associated with Silicon Valley startups and open source development as it has been embraced by many large multinational tech companies in the past decade.[61]

[61] The Free and Open Source Software movements began in opposition to corporate practices of maintaining the secrecy of proprietary software and code. Free Software and Open Source are separate terms part of distinct but related movements, the first established in the 1980s by Richard Stallman who founded the Free Software Foundation, and the second was closely tied to the dot com boom and an economic argument of increased value and cost savings that came by Eric Raymond (Kelty 2008, 99). Corporate involvement with Free and Open Source began in 1998 when Netscape gave away the source

However, in Singapore, where computerization and software development was largely first a government instituted project beginning in the 1980s, the form of entrepreneurship and hacking currently being promoted by the student hacker group and others has a more recent history. For example, although the government has been promoting technological development and entrepreneurship for the past three and a half decades, Singapore's "hackerspace" was started only in 2009 after its founder returned to Singapore from the US where he had founded and worked at start-up companies (hackerspace.sg n.d.). Many in Singapore feel that the current local entrepreneurship and hacking scenes are deficient in a variety of ways, including the overall quantity and quality of "talent," a devaluation of risk related to "Asian" and Singaporean cultural values, and generally low pay for programmers and engineers (see, for example, Do 2013; Quek 2015). In speaking with various members of the student hacker group at different times, they similarly told me that they self-organized as a group to promote a hacker culture, partially because they felt it was significantly lacking among students at Temasek University.

---

code to its main product, Netscape Navigator, to today where websites like Github provide a platform for over 3 million developers to share the source code for thousands of different projects and programs. Entrepreneurial start-up companies as well as large corporations make use of and contribute to these and other freely shared libraries and tools for their products.

Stallman and Raymond, as well as other figures like Mark Zuckerberg cited by the student hacker group have diverging politics and moralities in relation to the freedom and ownership of software and the role of companies and profit. Yet, they are all treated as examples of hacking because of their shared involvement and embeddedness in discourses and practices relating to hacker culture, entrepreneurship, and innovation. Free and Open Source Software are thus no longer seen as antithetical to corporate development, growth, or profit, even if figures like Stallman insist on moral rather than economic arguments for the freedom of software (Stallman 2014).

The student hacker group adopts and works to promote a particular way of being a computer scientist that centres on interest in and passion for computing. As one of the members of the group explained:

> Computer science, as in the faculty, was filled with students who weren't very passionate about computing. And it was very strange to me and a bunch of my friends, because like at that time Facebook was really hot and the social networking craze was still on, so it seems like Silicon Valley and technology in general is on the ascendency and here you have classmates who just code for school and classes and they go back and they do other things - they're not passionate about computing… And so [the student hacker group's] goal is to just grow the attitude that programming should be fun, can be fun. So this is hacker culture that you can plug into, that building things and learning things out of interest and just because you can should be the norm. I think to some degree we succeeded (Andrew 2013).

Andrew argues that passion should be the primary motivator for those studying and doing computer science. The hacker group argues on their website that hacker culture is *the* most important element to have in a tech community, seeing it as a source of innovation, producing economic value, and producing a community of like-minded people and companies for creating new projects and ideas.

The student hacker group centred on approximately fifteen "core-team" members who were responsible for management of the group and running group activities, all of except one of whom performed as men. These students were also from several different nationalities, including Singaporean, Chinese, and Indian. They held a meetup every Friday on campus that was open to anyone and featured a variety of computing-related content including personal projects or interests of members and others, and speakers from various companies who gave recruiting and technical talks. These gatherings were casual, with pizza provided, and attendees would sometimes bring their laptops to do computing work during or after the talks. The group also ran a series of workshops, held on the weekend once a month on different technologies or programming languages students might find useful but were not taught in class, centring primarily on web development. These introductory workshops provided the basics for participants to continue learning on their own afterwards, and were attended by students from several different faculties. One participant commented to me that she was there to learn skills that were sought after by employers and prevalent in "today's technology world."

The core team also ran a hackathon during the second semester of the academic year. This 24-hour team competition that took place in the main computing building at Temasek University was attended by students in computer science and related fields from the university along with a couple of teams from local polytechnics and secondary schools. The hackathon was open to students from any educational institution in Singapore. Following the pattern of many other hackathons around the world, the teams were tasked with building a technology (generally a program, but sometimes also integrating hardware). At the end of the 24-hour period the teams' projects were presented and judged, with prizes awarded for first-, second-, and third-place, as well as popular choice and best freshman or non-tertiary project. Swag (free promotional items) was provided for participants from local companies and Silicon Valley company, Palantir, which has a recurring presence at Temasek University.

The early evening was also attended by computer science students who were not competing but wanted to experience the atmosphere or meet with friends. I attended as a participant but non-competitor and spent the evening developing a program to help with coding field notes, which – like the products of many hackathons – I never finished (Irani 2015, 15–16). I also spoke with organizers and experienced the atmosphere of my first hackathon. The rest of the 24-hour overnight period was filled with quiet consultation among team members, intensive planning and programming, and eating large amounts of the junk-food provided by the organizers. As the night wore on many participants could be seen asleep on the desks in front of them or across several chairs, which was my strategy for taking a nap in an empty classroom. The morning presenters were dreary-eyed but eager and the large common area filled with new energy as judges and observers, including professors and members from the wider Singaporean tech scene filled the area for the presentations and awards.

The practices of the student hacker group are connected to those of the wider Singaporean tech community, which similarly runs a variety of meetups and workshops usually centring on specific programming languages or topics. For example, I attended meetups or workshops for specific programming languages including Ruby, PHP, and Python, as well as for Agile development practices, and maker and other hacker meetups. These meetups largely followed a similar format to the Friday student hacker group gathering, with casual and informal presentations on a particular personal project and/or the technical details of a particular technology that the presenter wanted to share. Presenters were local or international professionals, students, or self-taught developers (none of which are mutually exclusive categories).

Thus, despite Andrew and others' perceptions that a hacker culture was lacking among students and in the Singapore tech scene more generally, there were a great many tech related events and gatherings happening in Singapore, although largely through the efforts of a small group of people, including the student hacker group. Even at the student hacker group welcome tea it was clear that many more people participated than the organizers anticipated given that not enough pizza had been ordered and the meeting room was so full that the organizers had to remove several tables to make more space. I also frequently observed and keenly felt the ways in which the values of the student hacker group led to the cultivation of particular personhoods, among hackers as well as among computer science students more generally. Since leaving Singapore, the news and other media I have read suggests that the prominence of the Singaporean entrepreneurial scene has continued to grow and expand (e.g. Tegos 2016).

In the coming pages, I explore in greater depth the values that the hacker group seeks to promote, looking specifically at the role and meaning of passion. I also consider how passion is related to talent and other "proxy" qualities and how the values of the student hacker group worked to create a benchmark against which students came to measure themselves.

6.1.1 The Meaning of Passion

It was almost a truism among students and faculty in computer science at Temasek University that a person needs to have interest or passion for computing to be/come a "good" computer scientist. One graduate student and TA explained this relationship when I asked if he thought anyone could learn computer science: "Yeah, if he has interest, yeah. It's all about interest… The thing is if he has no knowledge he needs more time, he needs to devote more time as compared to other, but it's more about interest" (Arjun 2013). When I asked what made someone a good computer scientist, fourth year student Shirley similarly explained how learning new things – and *wanting* to learn new things – is necessarily ongoing for computer scientists throughout their studies and careers:

> Strong incentive to learn stuff. Like always keep himself or herself updated with all the technologies. Like back then if you can do Java like you're considered good already, but after iOS you have to learn this, and after Android came out you have to learn this… So when all kinds of cool stuff came out and they open the API, you got to go read them and try to do something about it. Yeah, like passionate (Shirley 2014).

Shirley continued to explain how passion was a necessity for being a *good* computer scientist.

> But passion, I mean, because I really see someone and they only want to graduate they only want to finish this course and get a job and so they don't really like computer science, they don't really like programming. I don't think they can be a good software engineer, yeah, so at least you're interested in what you do (Shirley 2014).

Passion or interest was similarly mentioned in the answers of many students I interviewed and asked about what makes someone a "good" computer scientist, including those from earlier years.

For one first-year student, passion was the first quality he suggested as necessary, among several others: "A passion for coding I guess, and that is one lah, a passion for coding. I guess to be able to solve problems quickly. Maybe a good grasp of maths and coding… maybe some creativity like depending on… what kind of work is in it" (Ariff 2014). Many of these students did not participate in the student hacker group. When I asked, Ariff simply answered "I'm not skilled enough," in addition to not wanting to do extra coding, which he assumed was entailed in participation. Nevertheless he shared hacker ideals about the necessity of passion for coders and computer scientists and programmers to do a good job.

At the same time, when I asked in interviews for students to describe how someone showed passion or what makes a person passionate, they had a difficult time explaining it. In general, “passion” seems to represent a dedication and desire to learn more about computing and to do computing, particularly outside of class time and through self-motivation, showing the quality’s close relationship with independent learning. Performing passion could thus take on many forms, such as pursuing personal projects relating to computing; learning about various technical details in depth relating to particular programming languages, computer hardware, or algorithms; participating in tech communities like the student hacker group; reading computer science research; participating in programming competitions; or just programming for fun. Passion could therefore be demonstrated in multifarious ways, but was also singularly and narrowly focused on technological knowledge and technological engagement.

There were also proxy qualities or activities that were taken as indicative of passion that students and others could perform. As has been discussed in literature on hackers, and computer scientists more generally, in the US, they are commonly perceived as embodying “geekiness,” which include playing console and computer, strategy-based or role-playing games; enjoying science fiction and fantasy television shows, movies, and books; and showing an (often singular) interest in all things technology related, and with distinctly gendered implications and styles (Ashcraft, Eger, and Friend 2012; Cheryan et al. 2013; Turkle 1988). While many in computer science contested these stereotypes, displays of “geekiness” could work as a proxy for passion, or at least interest. A woman who I met at one of the student hacker group workshops, for example, commented to me how she usually dropped a well-timed reference to science fiction media such as Star Wars or Battlestar Galactica at tech events to pre-empt any questions as to whether she was in the right place.

The prominence of “geekiness,” however, was much less marked in Singapore than found in the literature on computer scientists in the US and I consider below how members of the student hacker group performed and presented themselves differently than many hackers in the US and elsewhere. Practices like playing board games, enjoying scifi and fantasy media such as Lord of the Rings and superhero movies, and playing computer and video games also seemed quite common among students in general, not just those in computer science. Many scifi and fantasy titles/franchises, including Marvel Comics, Lord of the Rings, and Game of Thrones have become part of popular and mainstream consumption and culture in Singapore as in the US and Canada. These markers were therefore much more dispersed and so much less distinctive in Singapore than suggested in literature on the US.

The centrality and meaning of passion, however, is nonetheless closely tied to discourses about entrepreneurship and labour in the tech industry in places like Silicon Valley and related to movements like those for Free and Open Source Software. On their website, the student hacker group uses an essay by Eric S. Raymond, a well-known American software developer who co-founded the Open Source Initiative, to explain their motivations. In the first among five points, Raymond argues that:

> To be a hacker you have to get a basic thrill from solving problems, sharpening your skills, and exercising your intelligence. If you aren't the kind of person that feels this way naturally, you'll need to become one in order to make it as a hacker. Otherwise you'll find your hacking energy is sapped by distractions like sex, money, and social approval (Raymond 2001).

Members of the student hacker group draw on multiple (largely US-based) figures like Raymond for their inspiration. They also participate in and run activities like hackathons that are a key part of entrepreneurial and hacker practice in the US and many other places around the world (Irani 2015). Raymond's statements also point to the exclusive and singular interest that is seen as an important part of hacking, where other activities "like sex, money, and social approval" are seen as mere "distractions" rather than normal facets of life. In this way, while "geekiness" may be less prevalent in Singapore, becoming a computer scientist can nevertheless become an exclusive identity where, to demonstrate one's passion, a person must make all facets of their life about computing.

This idea is reproduced in numerous small ways on social media platforms and across the Internet. The computer science department at Temasek University, for example, frequently posts on Facebook and other social media sites with news and events from the department, articles about programming, the tech industry, and technological developments, and internally-made memes, pictures, and infographs. Many of these pictures are expressions of celebration wishing students "Selamat Hari Raya Haji" and "Happy Friendship Day," or offering inspirational or reflective quotes by figures from Gandalf to Bill Gates.

I found the series of depictions mentioned in Chapter 4, which provide repeated stereotyped contrasts between "programmers" and "non programmers," particularly intriguing for the way they perform this idea that for those in computer science everything is or should be about computing. In addition to the contrasting representations of a heterosexual couple versus the SQL command “commit” under the heading “Commit,” depictions include: for “Ruby,” a simple representation of a gemstone under the non-programmers heading, but for programmers a snippet of code from the programming language named Ruby; for “Tree” a representation of a leafy tree (the plant) for non-programmers, but for programmers a binary tree – one of the quintessential computer science data structures; and for “String” a piece of yarn for non-programmers is paired with a series of boxes illustrating another data structure known as a string is shown.

As seen in these examples, the topics of comparison are mundane, but they both suggest that programmers are different from normal people and that those differences are intimately tied to a focus on technology across (and to the exclusion of) multiple facets of life and reality. I discuss in Chapter 8 how the promotion of a singular dedication to computing conflicts with other obligations and connections, such as those to family and place, as well as with gender roles and norms. These depictions are also part of a transnational flow of performatives of the embeddedness of computing worlds in everyday life, with examples of job ads, birth announcements, and marriage proposals, among other practices, being done in code.

In discussing the relationship between entrepreneurship and neoliberal discourses in Barbados as they are enacted locally and internationally, Freeman (2014) argues for the importance of attending to cultural specificities. She shows how entrepreneurship in

Barbados intersects with Barbadian cultural forms such as matrifocality, distinctions between reputation and respectability, and local religious practices. In the case of computer science in Singapore, however, it is difficult to decide which cultural specificities to consider. Students, professors, and administrators come from many different countries, computer science knowledge and practice are localized but heavily infused with their largely American historical influences, and students are encouraged to travel to other places and experience other "cultures" through academic exchanges, internships, tourism, and interaction with international students.

There is some media discussion of the Singaporean concept of *kiasu* – a fear of losing out, often oriented towards self-interest – which often translates to heavily focusing on grades and fearing to take risks. Those who are *kiasu* are often denigrated, as being *kiasu* runs contrary to the entrepreneurial spirit that many current government policies seek to promote. Recently Nominated Member of Parliament Kuik Shiao-Yin, for example, commented:

> I don't think kiasu culture should be celebrated. In fact, I think we should kill it. Because all these behaviours that we are telling Singaporeans are necessary to take us into the future - innovation, productivity, collaboration, generosity to the needy - they are wholly dependent on a person's desire and drive to generate greater worth and real value to share with the world. And kiasu culture doesn't give a damn about generating or sharing worth and value (Kuik quoted in Ong 2016).

The website for Singapore's Hackerspace also features the tagline "Singapore's first kiasu-free zone" ("About Hackerspace.SG" n.d.). While entrepreneurial risk-taking is premised on self-interested individuality, it differs from *kiasu* in its expression. Self-interested entrepreneurialism promotes calculated risk, whereas *kiasu* tends towards selfprotectionism and risk-aversion. Moreover, in my experience, the concept was rarely used by students.

However, the focus on passion among students at Temasek University takes on a local specificity in the heightened awareness that students seemed to have of the continual movement of persons and opportunities – or limits to opportunities – as part of the Singaporean tech scene and labour market, discussed in the preceding chapter. I also discuss below how performing and demonstrating passion works to set some students apart, to make themselves desirable to employers. First, however, I consider how passion is cultivated and developed among students; that is, how someone becomes a passionate computer scientist or hacker.

6.1.2 Cultivating Passion and Talent

The promotion of passion, and hacker culture more generally, represents the cultivation of a particular kind of affective personhood. Carla Freeman (2014) discusses how entrepreneurs in Barbados treat their "selves" as projects that are constantly being remade – projects intimately interlaced with neoliberalism. That is, they are entrepreneurs *of* themselves, following Foucault (2008). Their selfhoods are thus always in the process of becoming, always a "work-in-progress," similar to how Judith Butler has described the project of acting and doing gender (Butler 1988, 1999). Freeman shows how many facets of Barbadian entrepreneurs' lives, including partner and parental relationships, work, self-care, and religious practice, all become part of these projects of self-making, cultivation, and care. In this way, separation of work and non-work no longer has meaning. I did not investigate the non-academic lives of hackers and computer scientists at Temasek University in enough depth to consider whether their lives and selves are following similar paths to Freeman's entrepreneurs. Yet, it was clear that students' lives in school, including their personal and co-curricular endeavours, their goals and plans, were also significantly shaped by projects of self-making that were often oriented towards becoming passionate – and thereby marketable and employable – persons. In this regard, while passion refers to "strong or overpowering feeling or emotion," it also takes on a sense of self-interested calculation among students ("passion, n." 2005, II-6-b).

There were conflicting views and uncertainty among members of the student hacker group, and among students and professors more generally, as to whether it was actually possible for a person to cultivate or develop passion if they did not already "possess" it. This ambiguity was tied to similarly shifting and multiple views among the students and professors I spoke with about the nature of selfhood in relation to passion, talent, and skill. These perspectives generally varied between seeing the self as constituted by nature where talents and abilities are essential and inherent to an individual, or nurture where persons are "entrepreneurial selves" whose skills and values could be intentionally cultivated and could grow and change (Bryant 2005; Foucault 2008; Freeman 2014; Kondo 1990; Mahmood 2005). Both perspectives rely on a selfhood that is independent, autonomous, and reasoning, yet they differ on the role of agency in self-development.

The question of the nature of selfhood and of essential traits also closely relates to the ongoing question in computer science and computer science education research as to whether certain students possess a "talent" for programming. Over the past decade, for example, there has been growing attention to the significance of students and teachers having a "growth mindset" as opposed to a "fixed mindset" for students' success in learning programming and continuing in computer science (Aronson, Cohen, and McColskey 2009; Cutts et al. 2010; Dweck 2006; Dweck 2008; Murphy and Thomas 2008). A fixed mindset entails a belief that certain qualities are fixed, that some groups have them and others do not, often related to gendered and racial stereotypes. A growth mindset, in contrast, entails the belief that qualities can be learned and acquired (Dweck 2008, 2).

There is also a long history in the US of conducting aptitude tests, for assessing eligible computing students or job candidates. Nathan Ensmenger writes about early hiring practices in the US:

> One of the perennial problems facing the computer industry, in the 1950s and 1960s as well as the present, was defining precisely what characteristics or training made for a good computer programmer. As was mentioned earlier, programming was frequently seen as a black art whose success or failure was dependent on the idiosyncratic abilities of individual programmers. This notion was reinforced by a series of aptitude tests and personality profiles that suggested that computer programmers, like chess masters or virtuoso musicians, were endowed with a uniquely creative ability (Ensmenger 2010b, 19).

More recently, one unfortunately popular working paper with the catchy name "The Camel Has Two Humps," which was later retracted (Bornat 2014), suggested the authors had developed an aptitude test for incoming students to assess and distinguish those who could succeed at programming. The implication of the paper, and of aptitude tests more broadly, was that some students inherently have the capacity – or talent – for programming whereas others do not. The paper was mentioned to me independently by at least two senior computer science students, one of whom told me he was unsure of the results and implications, but which also shows its circulation in computer science discourse.[62]

One professor I spoke with similarly recalled how he was required to take an aptitude test when he was a student in Singapore in the mid-1980s. The desire to be able to assess who will become a successful programmer continues to the present. This professor mentioned that implementing an aptitude test is a repeated topic, discussed every now and then with a view to assessing which students would become good computer scientists, but also added that the department did not want to discourage students from applying and that he was unsure what qualities to include or how to conduct such a test.

With passion and talent taken as an innate quality, for many in the student hacker group, as well as others who self-describe as "interested in" or "passionate for" computing, pursuing computer science was then seen and explained as a process of self-realization. I asked all students early in interviews how or why they came to study computer science. Many answers were matter of fact: "Well, I like computers so it's quite natural" (Guoliang 2014) one student who I interviewed at the student hackerspace explained. Another student who is involved with the student hacker group, along with many other campus activities, explained:

> It goes way back to secondary school… it was an independent school, so the school decided for us what our curriculum would be like, which means that they expose all secondary 1 and 2 students to computing… So right at the start it was HTML CSS and then it slowly progressed to Flash, and then it went to PHP a bit of JavaScript and C++ so that was where I was first exposed to it, you know. And from then on, I was, you could say I was addicted ah. So on my own I would you

[62] The paper was retracted after my fieldwork.

> know do some little experiments, read up more, you know technology, and flash, all the devices and stuff, that caught my attention as well so it was, I always had an interest in this kind of things. And then the break came when I had to choose my university major. Yeah so, long story short… I decided to come to [Temasek University], yeah, so that's how it started. (Xiaowen 2014)

Xiaowen is referring to the fact that most Singaporean schools do not offer computing related courses at Secondary School or Junior College.[63] However, Xiaowen was fortunate enough to attend an independent school that did offer courses in computing, a distinctive opportunity that he elides in his discussion, where he was "exposed" to computing and which sparked the interest that he "always had." While their passion and interest for computing sparked at an early age may indeed be a significant reason for these students to pursue computer science, these responses and narratives also act performatively. Guoliang and Xiaowen are producing and enacting their innate and deep interest in computing through their self-descriptions and stories of their paths to computer science, even as these interests are seen as an essential, or at least a deep and meaningful, component of their personhoods.

The contrast of the answers above to that of students who are more ambivalent about their participation in computer science is illustrative: "I sort of fell into this area because I don't have much choice. Our only choice was between Science, Engineering and Computing, these three areas. I don't have much interest in Science or Engineering, so ok, so Computing," Qiaohui explained to me when I asked her why she chose computer science (Qiaohui 2013). Qiaohui is a student from China studying in Singapore

[63] Beginning in 2016 several schools will be newly offering computing or programming related subject in response to the government's "Smart Nation" initiative.

on a scholarship that restricts her choice of subjects to study. I will discuss my interview and interactions with Qiaohui further in Chapter 8, but while she sometimes described her strong interest in computer science, she also was much more ambivalent about her future in the discipline. Her answer provides no declaration of pursuing a passion or interest. Instead she made a calculated choice among field of possible options, about which subject was the least worst choice for her.

At the same time, while students and professors often treated passion and talent as essential and inherent, students also clearly discussed the ways they were choosing their activities, projects, and courses to make themselves distinct and passionate persons. A second year student, for example, explained why he pursues independent projects despite the impact it sometimes has on his grades:

> I think I should be spending time on that because there are a lot of people who do well in academics, there's nothing really that sets you apart. And frankly, as I said before, everything I study, I don't really think it will help me later, where these things do help me for sure. Because I can really see, it's really tangible oh this is helping me right now. So, sometimes I wish I had more time to spend on these things (Dinesh 2014).

Many students had specific skills, languages, or abilities they were interested in developing. After discussing with a senior student his plans to work at a company to develop new skills before founding his own startup, I asked what skills he felt he was missing:

> More recently, business development, the ability to cut deals when you're a tiny company…. This is the long run, and if I want to do a company in Asia, in Singapore, I have to learn local advantages. So, step one would be to figure out what those advantages are. I have an idea what they are, but how to use them, how do they look like, what are the pitfalls (Andrew 2013).

Similarly, when I asked another student whether she knew about or paid attention to government policies, she responded: “I pay attention to Singapore IDA’s [Infocomm Development Authority of Singapore] policies into different areas of computing. I will be able to see the trend and try to equip myself with the kind of knowledge indicated in IDA news to cope with the instability of IT career” (Xuan 2014). Students focused their learning around the persons they wanted to become and the careers that they wanted to pursue.

This practice was clearly evident among members of the student hacker group, as seen in Andrew and Dinesh’s discussions: both students were involved with the group. At the same time, Xuan was not involved with the group yet clearly shows how she seeks to “equip” herself with knowledge and skills oriented towards her future career. It is difficult to separate affect and instrumental action here, as each is taken as demonstrative and productive of the other. Dinesh pursues independent projects because he is interested and passionate, and also because he sees it as a good career move, but these projects simultaneously demonstrate that he possesses those very qualities and feelings. Passion became a means and end for students to make themselves as a particular type of computer scientist, embodying particular affective qualities and particular practices and behaviours, while also performatively creating those sentiments and reproducing their value and significance.

This cultivation of passion, alongside independent learning and critical judgment discussed in the preceding chapter, can be seen as the cultivation of a particular computer science *habitus:* particular dispositions, ways of behaving, dressing, acting, moving, and thinking that are directly related to the social structure in which they are generated and practiced (Bourdieu 1967, 1977). As discussed previously, a great deal of social science research on education has explored the ways class, gender, and racial positions, identities, and practices are reproduced in strategic ways through education (Bourdieu and Passeron 1977; Holland and Eisenhart 1990; Khan 2011; MacLeod 2009; Weis 1985; Willis 1981). Paul Willis, for example, showed how and why “the lads” chose working class lives based on the interaction of their “penetrations” of the ideas and material reality underlying the schooling system and ultimately the limited opportunities it offers, and “limitations” imposed on these realizations that led them to valorize manual labour as an expression of masculinity (Willis 1981, 128–130,148). He showed how they made changes in bodily style and dress to embody an opposition and freedom from the structures of the school, even as that oppositional style served to cement their class position (Willis 1981, 17–22).

Shamus Khan, conversely, shows how students at the elite private school of St. Paul’s embody privilege rather than resistance. He argues that modern youth that attend this school “develop privilege: a sense of self and a mode of interaction that advantage them” (Khan 2011, 14). Yet, rather than cultivating particular tastes that set them apart, they “display a kind of radical egalitarianism in their tastes. Privilege is not an attempt to construct boundaries around knowledge” and as a result “inequality is explained not by the practices of the elite but instead by the character of the disadvantaged” (Khan 2011, 16). Khan shows how through repeated practice during their time at St. Paul’s, students learn to embody a sense of ease in a wide variety different social and cultural settings and activities, including while eating, dressing, and dancing. Students have thus embodied the sense that they have the right to be wherever they are and do whatever they are doing. This transposable *habitus* thereby displays and reaffirms their privilege.

Bourdieu's approach to *habitus* treats these dispositions and their transmission as performed and practiced largely unconsciously or sub-consciously – as "second nature" (Bourdieu and Passeron 1977; Bourdieu 1967). He considers how students at universities in France are able to exchange their "cultural capital" – knowledge, dispositions, values acquired through their early life and previous education – for privilege, acceptance, and success at the university. In their lengthy analysis of the relationship between education and thought processes, for example, Pierre Bourdieu and Jean-Claude Passeron consider the influence of such cultural capital in oral examinations:

> Class bias is strongest in those tests which throw the examiner onto the implicit, diffuse criteria of the traditional art of grading, such as the dissertation or the oral, an occasion for passing total judgements, armed with the unconscious criteria of social perception on total persons, whose moral and intellectual qualities are grasped through the infinitesimals of style or manners, accent or elocution, posture or mimicry, even clothing and cosmetics; not to mention orals like those of the École Nationale d'Administration or the literature agrégation, where the examiners almost explicitly insist on the right to implicit criteria, whether bourgeois ease and distinction or university tone and breeding (Bourdieu and Passeron 1977, 162).

Students such as Xiaowen and Guoliang who discussed above their "natural" disposition and interest in computer science both had previous experience programming from secondary school. They are able to exchange these knowledges and skills, their ease with programming structures and modes of thought, as well as self-narratives about their early learning experiences, for recognition and easy acceptance as a member of the student hacker group.

Most members of the student hacker group that I spoke with had similar prior experience in programming, robotics, or another related subject. Research on computer science education and gender in the US has shown how experience in computing prior to starting university is often seen a significant factor in students' success throughout their studies, and this difference is significantly gendered (Margolis and Fisher 2002). In Jane Margolis and Allan Fisher's well-known book *Unlocking the Clubhouse: Women in Computing* (2002), which explores the experiences of men and women studying computer science at Carnegie Mellon University, they consider a variety of factors that privilege men's success and continued enrollment over women's including early childhood and secondary school computing and math experience, as well as enacting stereotypes of "geek" computer scientists, and disparities in confidence and interest. This disparity in success and enrollment can be understood at least partially in terms of cultural capital, although Margolis and Fisher do not use a structural analysis in this way.

Yet, students clearly also *work* deliberately to make themselves as particular knowledgeable, critical, and passionate persons. Willis and Khan both show how *habitus* can be acquired or trained with some level of deliberateness, even if not in a fully self-conscious manner. As with Khan's (2011) discussion of students' repetitive practice at school of learning to be at ease in different situations, the repetitive practices discussed in Chapter 3 are meant to train students in modes of practice and modes of thought through repetitive activities such as listening to lectures, doing programming exercises, assignments, and tests, and various personal interactions where students learn to judge and assess what makes a good computer scientist, and what is good computer science.

Starting a computer science degree at Temasek University with the cultural or technical capital of previous knowledge and experience was then not the only pathway to becoming a good – independent, networking, and passionate – computer scientist.

Take fourth-year student Shirley, who I have discussed in previous chapters and above in relation to the independent learning practices she developed, using a variety of resources like Stack Overflow. Shirley did not generally participate in the student hacker group, only took a short introductory course in programming before starting computer science at Temasek University, and like Qiaohui she discussed how as a student on scholarship from China she did not have many choices in which discipline to pursue. Yet, Shirley had participated in an overseas exchange and an internship and entrepreneurship program at a startup, both in Europe. When I spoke with her she had an embodied confidence and ease that I had come to recognize mostly in senior students, and particularly those who had studied or worked abroad. She also took all of the advanced computer science courses, as well as the senior level courses that were considered "hard core" and a lot of work, and worked as a TA for one of the introductory programming courses. Her group of friends is largely situated in computer science and when I asked her if she felt like she belonged in the discipline she responded:

> Yeah I feel like computer science is kind of my passion and all my friends around me they kind of feel yeah they are really passionate about programming and building software and building their own like stuff like cool stuff, some geeky projects. Really cool, yeah (Shirley 2014).

While Shirley is not involved directly in the student hacker group, her participation in the internship meant she was partially involved in the same field of discourse and practice, or community of practice (Lave and Wenger 1991). Although Shirley does not claim the same self-narrative as Xiaowen and Guoliang above, she clearly presents herself as a person who is passionate and who has shaped her self and her life around doing computer science. Shirley's case shows how students can *develop* a computer science habitus*;* Shirley has chosen her path through computer science as a way to create and display a passionate personhood.

6.2 The Hegemony of Entrepreneurial Passion

While I discuss in Chapter 8 different ideas among students about what it means to be a good computer scientist, I consider here how students' options are limited. The emphasis on entrepreneurship, hacking, and passion within the university, the tech industry, and as part of national policy constrains students' choices when they need also to make themselves attractive to employers after graduation. It may not be seen as such by students, but these projects of self-making are thereby also processes of subjectivation. Students are caught up and summoned to make themselves in ways that fit with government discourses and an ethos of neoliberal governance (Foucault 1990, 1997a; Mahmood 2005).

6.2.1 Summoning Entrepreneurial Citizens and Subjects

Singaporean policies continually call upon Singaporean citizens and workers to become more – more creative, more risk-taking, more innovative, and more entrepreneurial. The National IT Plan in 1986 first sought to encourage "creativity and entrepreneurship" in Singapore (The Straits Times 1986). In 2006 (twenty years later) the "Intelligent Nation 2015" report continues to discuss the need to "develop globally competitive infocomm professionals" projecting that "to thrive in the world in 2015, Singaporeans… have to be more risk-taking, entrepreneurial" (iN2015 Steering Committee 2006, 70), and then again eight years later the "InfoComm Media 2025" report suggests that the workforce training system should do more in several areas, including "to cultivate in workers the willingness to take risks and innovate" (IMMSC 2015a, 38). In order to continually envision future goals and developments for Singapore to become a globally leading nation, Singaporean policy visions thus continually emphasize what local expertise lacks – what needs to be changed and improved.

To this end, the government has created special spaces where an entrepreneurial ecosystem is meant to grow and thrive. The Biopolis, Fusionopolis, and Block 71, all located within the area known as "one-north," featuring a Mass Rapid Transit (MRT) stop of the same name, are sites that were developed in close proximity to educational institutions like the National University of Singapore, and research centre A*Star, as well as the National University Hospital, to promote a contained ecosystem where innovation and creativity could be fostered in relation to biotechnology, technology, and entrepreneurship. I frequently visited Block 71, featuring an industrial building that has been converted to space for technology and other start-ups and related organizations and where there were frequently held technology talks and meetups. Several scholars have considered the formation of these regions or zones and how they work to produce spaces where alternative cultural and creative talents grow, seemingly separate and freer from the moral and cultural constraints governing Singapore's "heartland" population (Carver 2010; Clancey 2012; Fischer 2013; Krishna and Sha 2015; Tan 2003; Waldby 2009; Wong and Bunnell 2006).

Efforts to continually cultivate more creativity and innovation in Singapore are also spurred on by academic articles, research reports, and global indices that similarly judge the creativity, innovativeness, and entrepreneurialism of Singaporean workers and the Singaporean tech ecosystem. Some of this research touts the success of technological development in Singapore (Arun and Yap 2000; Wong 1992). Yet, others point to various ways Singapore and Singaporean workers continue to be deficient (Comunian and Ooi 2015; Wong, Millar, and Choi 2006). Caroline YL Wong, Carla CJM Millar, and Chong Ju Choi, for example, suggest that:

> Although Singapore displays many characteristics typical of a knowledge-based economy, Singapore's new economy seemingly lacks a stimulating climate conducive to imagination, innovation and adventure that will attract and retain globally mobile talent (Tan, 2003, p. 403)… There is a need to work further on capabilities such as knowledge creation and application especially when they are applied to the entrepreneurship scene in Singapore. This is where social and cultural mindsets would have to embrace individual creativity, diversity, and community-led initiatives. (Wong, Millar, and Choi 2006, 80, 86).

As Kai Wen Wong and Tim Bunnell point out, government and academic discourses about innovation and new economies often echo one another and "academics may be complicit with the dynamics of the new economy that they are studying" (Wong and Bunnell 2006, 81). That is, academics often themselves contribute judgements about Singaporeans citizens and workers as lacking creativity and other entrepreneurial qualities.

In this regard, I want to emphasize that the story I am telling here – my analysis of Singaporean polices and of computer science students – is not meant to assess or judge the level of creativity or passion possessed or demonstrated by my research participants, or Singaporeans more generally. Studies such as that of Wong, Millar, and Choi (2006) and a variety of global indices that do make such assessments are part of the field of discourse that I am exploring in this dissertation. In this regard, the numbers, the rhetoric of deficit, and the valuation of creativity and innovation all need to be problematized. For example, Singapore was rated the 9th most Innovative Country in the world in 2014 by the World Economic Forum and 9th in the world on the Global Creativity Index (Florida, Mellander, and King 2015, 22; Schwab and Sala-i-Martin 2014, 20). Despite these competitive rankings, the government continues to suggest that more creativity and more innovation are needed. During a press conference at the Research, Innovation, and Enterprise Council (RIEC) in 2016, for example, Prime Minister Lee Hsien Loong commented:

> We will need to train our young people to encourage them and interest them in science and technology and give them the sense that, indeed, it is possible to do exciting things in Singapore and change the world… We have to make this a creative, a fertile, an innovative and enterprising environment (Lee 2016).

These contradictions clearly show that assessing levels of creativity, innovation, entrepreneurship, and related qualities are contextually dependent and politically laden. Multiple actors around the world are invested in the question of, and power to determine, exactly what counts as "creativity."

In particular, speeches and news reports by policy makers over the past two decades often refer to Silicon Valley as the epitome and model for technological and economic success – what Singapore could and should be, but is not (yet). In his 1999 National Day Rally Speech, for example, then Prime Minister Goh Chok Tong expressed: "We must foster an entrepreneurial culture in Singapore. We have to create a 'Silicon Valley' state of mind in Singapore – creative and willing to take risks, setting up start-up companies and getting venture capital" (Goh 1999). These calls for Singapore and Singaporeans to be more like Silicon Valley continue to the present day. Most recently, current Prime Minister Lee Hsien Loong commented that Singapore needs to value engineers like they do in Silicon Valley in order for Singapore to achieve similar success, after visiting the area and meeting with entrepreneurs such as Facebook founder Mark Zuckerberg (Spykerman 2016). Policy makers in Singapore thereby look beyond the city-state for ideas and models of technological development and practice that can be applied in Singapore.

These policies treat Silicon Valley and places like it as the embodiments of creativity, innovation, and entrepreneurialism. Yet, as long as Silicon Valley is revered as the vanguard of future-making (Suchman 2011), it also holds the power to define the meaning of these terms. Planning and policy makers' efforts to call upon Singaporean citizens and workers to become more entrepreneurial, more risk-taking, more creative, more innovative, more like experts in Silicon Valley, thus continually positions them – and Singapore more broadly – as not yet there, as lacking in these qualities in sufficient depth or quantity, and as always in need of improvement or greater investment. As the "Infocomm Media 2025" report points out, Singapore needs to continuously move forward to "remain at the forefront and be ready to be among the early adopters of next-generation communications technology" using the "5G standard for mobile wireless communications that is expected to be formalised in 2020" as an example of the need to progress as part of ever-changing and ever-expanding technological futures (IMMSC 2015a, 13).

In this regard, the competition to be a world-leader both in technological usage and expertise is similarly always ongoing. The national computerization plan in the 1980s had relatively modest goals that were national in scope as the government sought "to turn Singapore into a computer city" (The Straits Times 1983), although the plan relied in many ways on involvement by companies and organizations from the US, Japan, and Europe. At the same time the project sought to create regional relevance and pre-eminence for Singapore, aiming to become "the brain centre of the region" (Zhou 1982). However, the government's visions quickly expanded, seeking to be a global hub with global expertise. The introduction of the "IT2000: A Vision of an Intelligent Island" report in 1992 envisions: "With Singapore positioned as a vast information 'gateway,' the country can also become a global hub attractive to companies with global operations and to experts in numerous fields who can apply their expertise world-wide" (NCB 1992, viii). Eight years later, the "Infocomm 21: Singapore where the Digital Future Is" report notes a further "paradigm shift" brought about by the spread of the internet, necessitating Singaporeans to "think global, act local" (IDA 2000, 5). This push for global connectedness continues and intensifies towards the future and looking into 2025, with the "Infocomm Media 2025" report envisioning "modern and state-of-the-art" infrastructure in the form of a Heterogeneous Network (HetNet) to "connect everything" "connect everyone" and "connect everywhere" "all the time" to become the "world's first Smart Nation" (IMMSC 2015a, 2, 22).

Literature on the growth and success of knowledge economies have positioned cities as zones of creativity and the centres of creative industry (e.g. Gordon 2013;

Hospers 2003; Yencken 1988). Desire is a significant facet of both contemporary capitalism and the constitution of such megacities, which work “as a space for the actualizing of national desire for global values and prestige” (Ong 2007, 86). In this regard, Ong suggests that “megacities” like Singapore are “mega” not so much in terms of the size of population but rather in terms of “the scale of political ambition invested for the urban accumulation of foreign talent and creative knowhow” (Ong 2007, 83). Singaporean policies on foreign talent, education, and technology development set those desires into motion. Universities and institutions in Singapore, along with students, professors, and others, are caught up in these policy visions as the goal to become creative, innovative, and entrepreneurial becomes a normal part of the (neoliberal and Silicon Valley-centric) social and moral order.[64] Among students in computer science, the diffraction of these policies and their norms of conduct, in intersection with the desires and demands of employers, is to (re)produce passion as part of a hegemonic personhood in computer science.

[64] These national policies, for example, are implicated in policies and practices at Temasek and other universities. At Temasek University there is a university-wide centre focused specifically on entrepreneurship, running internship programs for students to work at start-ups overseas, building industry partnerships for developing and commercializing intellectual property, providing a platform for networking among entrepreneurs, and conducting interdisciplinary and collaborative research projects. The computer science department also has a branch focused specifically on entrepreneurship that has developed partnerships with large multinational tech companies to help students and professors found successful start-up companies. A professor and administrator explained, “it’s not so much about what regulations there are but really, how do you exploit the ecosystem, what money you can get, what talent you can get, what help you can get.” The university and department are therefore contributing to the governments’ entrepreneurial goals, even if it is not done specifically to follow government policies. Moreover, the history of the National University of Singapore shows that the government has frequently had direct involvement in shaping the direction and content of higher education in Singapore (Gopinathan 1989).

6.2.2 The Hegemonic Computer Science Personhood

Passion is not quantified or measured in the same way as course learning, which is given a numerical assessment through course grades. Nevertheless, students come to judge themselves and others based on their performance of this passion, just as they learn to judge algorithms and code as discussed in the previous chapter. When I asked one student if he thought he had the kind of passion he had been discussing as essential to learning and doing computer science, particularly in the Singaporean context where the pay was not nearly as good as in finance or management, he responded:

> I guess, I would say that I'm passionate, but I can see people who are more passionate than me. Sometimes I feel inadequate when actually I have done a lot more than many of my peers in the school. But, yeah, sometimes it's just difficult to find the time to do what you want to do (Christien 2014).

Christien judges his passion and achievements relative to other students, simultaneously feeling like he is lacking while also recognizing the many things he has done. When I spoke with Christien he was in the middle of an internship with a large tech company in Singapore. He also discussed how he had many ideas for tech projects, but was also dedicating a great deal of time to a co-curricular activity in martial arts. Having varied interests thus limited his ability to dedicate time to these projects, particularly in relation to other students he had met in advanced classes and associated with the student hacker group, who spent the majority of their time on computing related endeavours.

In this way the members of the student hacker group are positioned as hierarchically superior to or better than others based on their passion, interest, and dedication to computing. When I told people about my research, I was constantly advised to speak with or hang out with the student hacker group, as well as other hacker, maker, and programming language communities around Singapore. These suggestions partially reflect the groupness and self-promotion of these "groups" (Brubaker 2004). They are certainly the most visible, having particular spaces (co-working spaces and hackerspaces) dedicated to their computing practices, as well as specific and generally public events and meetups to share and discuss their projects, activities, and interests. Hackers and open source developers have also received the greatest academic attention from social scientists, who focus mainly on the new social forms that are engendered through distributed and open forms of collaboration, or on the large gender disparity among hackers and open source developers (Coleman 2013; Coleman and Golub 2008; Håpnes and Rasmussen 1991; Irani 2015; Kelty 2008; Nafus 2011; Takhteyev 2012; Turkle 1988, 2005).[65] Yet, I also felt that these various groups, events, and spaces were repeatedly suggested because they are seen to exemplify "real" or "good" programmers and computer scientists.

The way students are compared (and compare themselves) through grades and the bell-curve, closely relates to the exceptionalism that is perceived of and promoted by hackers. In this regard, as Ian Hacking argues, while normal can describe an average – an ideal or perfection, or something that is "right" – it also came to describe something that is *only* average and that can and should be improved (Hacking 1990, 168). A student's performance of passion, along with technical skill and course grades, are taken as something that can and should always be improved upon. Students learn that what they

[65] According to a survey by GitHub (one of the primary platforms for code sharing and open source development) of its members in 2017, only 3% of its members are women (GitHub 2017), whereas the proportion of women in tech positions at private companies such as Google is 20% (Google 2017)

should strive for and embody is the right tail end of the bell-curve where a person is exceptional – both in terms of being different and better than the "average" or "normal" student. Certainly it is difficult to contest the value of spending a great deal of time, and wanting to spend time, on computing related activities for developing a person's knowledge and skill. Yet, having and showing passion are becoming necessary for being perceived as a "good" computer scientist; passion has become hegemonic.

Following Antonio Gramsci, the dominance of a social group manifests in two ways: through "the 'spontaneous' consent given by the great masses of the population to the general direction imposed on social life by the dominant fundamental group" and through the operation of "state coercive power which 'legally' enforces discipline on those groups who do not 'consent' either actively or passively" (Gramsci 1988, 306–7). Within the realm of computing, while members of the student hacker group would certainly contest this statement, hackers and entrepreneurs are becoming, if they are not already, "the dominant fundamental group" within computer science at Temasek University; the student hacker group was the default point of reference for what constitutes a good computer scientist, for research on computer scientists, and in the search for tech employees.

In relation to the student hacker group at the Norwegian Institute of Technology, Håpnes and Rasmussen (1991) argue that their processes of identity construction and self-promotion worked as a form of claim-making to a privileged position, with greater visibility and contact with and support from faculty members. The identity embodied and promoted among Norwegian hackers was significantly different from that I found among the student hacker group in Singapore. The Norwegian students' performances of self compare closely to those of early MIT hackers discussed by Turkle, including a disavowal of bodily concerns (dressing, washing, grooming, etc.), an avoidance the complexity associated with intimate personal relationships with other people (romantic or otherwise), and a singular focus on technology (Håpnes and Rasmussen 1991; Turkle 1988, 2005).

In contrast, I found the majority of those involved with the student hacker group at Temasek University to be clean, well-spoken, and generally casually dressed but with care taken in their appearance. As a group they were focused on outreach as much as internal collaboration and identity formation, and so group members were involved with a great deal of social interaction as they ran workshops, gave presentations on their projects, and organized events, for example. Some group members were particularly interested in entrepreneurship, partially due to the ways the tech industry has changed since the 1980s, where hacking (as an identity and practice) now intersects with free and open source development and entrepreneurial startup endeavours.

Yet, despite the many differences between the Singaporean and Norwegian hacker groups, both of their efforts at identity formation and self-promotion have produced for them positions of power and influence, as well as an identity and benchmark against which others are compared and assessed within their respective schools. The Temasek University student hacker group had been given their own hackerspace on campus, which was fairly significant in size, and which, while technically public, was used primarily by group members. The group's events were supported by a university division dedicated to fostering and promoting entrepreneurship, as well as given general encouragement by the department. While faculty members were generally not directly involved in the group, I did see one professor attend a student hacker group event. The group had also become one of the primary points of contact for local and international companies, organizing special recruitment talks, product demonstrations, and technology talks by organizations including Microsoft, Google, Palantir, and Viki (a Silicon Valley video streaming site, with offices in Singapore).

Beyond just benchmarking passion as a means of assessment, the cultivation of passion and hacker culture represents the promotion and normalization of a particular way of becoming and being a computer scientist – a hegemonic form of personhood. This personhood is exemplified by members of the student hacker group and given dominance as they perform and become representative of passion. They also become representative of the values that the government has promoted for two and a half decades. Their dominance is also reinforced through the coercive power of the tech industry to hire those who display these qualities and to discount those who do not. The student hacker group then works as an embodied example against which others compare and judge themselves and their own performance of entrepreneurship, risk-taking, innovation, and creativity.

Historians of computing have shown how, in relation to the tech industry in the US and the UK, a seeming continual under supply of computing professionals has rarely been just about the numbers (Abbate 2012; Ensmenger 2010; Hicks 2017). Measures such as "character," "personality," "dedication," "motivation," and "aggressiveness" were used historically in the UK as highly subjective judgements of skill for programming jobs in ways that were distinctly gendered and otherwise biased (Abbate 2012, 64). Employers I spoke with similarly treated passion as a proxy for skill and talent. One recruiter for a large multi-national tech company who was interviewing students from Singapore, for example, explained that her company wanted a "great technical person" – the "techiest of the techies." She said she looked at graduating students' GPAs and such qualities as leadership skills. However, she also commented that involvement in CS or technology related clubs shows a "passion for technology," which suggests that they can "crank out good code because they're so passionate about it." A hiring manager at another multinational tech company similarly explained:

> I want someone who's passionate about the job… Somebody that comes in at 9 leaves at 5:30, whatever it is, does their 8 hours, generally not, yeah. I'm not sure if I'm saying something right now, but generally not that passionate. You know, I want someone who is passionate for software development as well as having the skill and ability to do that (David 2014).

As an ambiguously defined quality, employers have the control to determine what performances of passion are valued, how much passion one needs to display, and what kind of person is capable of displaying these qualities.

Legends and debates also circulate through tech and business news and blogs about the 10X programmer or engineer, the idea that a good programmer is ten times more productive than just an average programmer.[66] Tech companies therefore try to sift through a sea of merely "average" programmers to find the "top talents who are passionate and committed to creating the best products and services that are a step

[66] This idea originated with a 1968 paper that compared programmer performance, which was then popularized by computer scientist Frederick P. Brooks Jr. in his book "The Mythical Man Month" (Brooks Jr. 1995; Sackman, Erikson, and Grant 1968).

ahead," as a recruitment brochure from Sony that I picked up at the computing career fair at Temasek University asserted. The idea is that if such a 10X programmer exists and employers can successfully find and hire them, they will produce much greater "value" for the company. With the hiring practices of many major future employers tied to the idea that students should have and demonstrate passion, students' choices of *how* to be/come a computer scientist are limited. Students are rendered into narrow personhoods that can be assessed and measured (qualitatively and quantitatively) to see if they are the "techiest of the techies" and will produce value for tech companies.

The hegemony of passion is further reinforced as the continual sense of deficiency in terms of passion and entrepreneurialism being (re)produced by the student hacker group, government policies, and employers and the tech industry also works to manufacture labour shortages. Nandita Sharma considers how qualitative labour shortages are created in Canada in relation to jobs that are not "attractive" to Canadians and because Canadian citizens legally and economically have more and better options, thereby justifying the recruitment of migrant workers (Sharma 2006, 98; see also Knott 2016). Education and jobs in computing are relatively and increasingly attractive in Singapore, despite statements that it is a less prestigious field than business, law, or medicine. It was also expected that over time Singaporean experts would be produced in sufficient numbers to meet local industry and public-sector demands. The National IT Plan, for example, had as one of its goals: "upgrading the skills of existing IT workers and providing *enough* R&D software engineers" (The Straits Times 1986, emphasis added). However, changing areas of expertise needed within the sector mean that the number of workers available – and the quality of expertise – is always judged as deficient such that more professionals and more experts are always needed.

For example, according to a 2016 study by Singapore Management University and J.P. Morgan stated that 15,000 additional ICT workers would be needed by 2017, particularly in the areas of cyber security, data analytics, and network infrastructure (J.P. Morgan-SMU 2016). Yet, at the same time, unemployment in this sector was 5.5% for residents, well above the national average in 2016 of 2.1% (MOM 2017; TodayOnline 2017). Thus, while professionals were and are being produced by expanding local educational programs, qualitative shortages by type of expertise are being manufactured by, and to justify, the continual derogation of Singaporean citizens and workers and the privileging of foreign talent, including students, educators, and professionals.[67] The (re)production of deficiency simultaneously summons the need for students and workers in Singapore – and for Singapore as a place – to continually shape themselves in ways that are desirable and competitive as part of the (trans)national tech industry. As such, in computer science, hegemonic personhood is intimately intertwined with neoliberal forms of affect and exploitation.

6.3 Reproducing Neoliberal Affect and Work

Paul Willis's study of "the lads" showed how their culture of resistance worked to reproduce their working class positions, in part through their positive affirmation of

[67] Singapore also relies heavily on foreign workers to fill jobs that are "unattractive" to Singaporeans, such as construction and domestic work. Such foreign workers are afforded much fewer rights, privileges, and opportunities for integration compared to foreign "talent" in Singapore (see, for example, Huang and Yeoh 2003; Yeoh 2006; Yeoh, Huang and Gonzalez 1999).

working class culture (Willis 1981, 101–4). The cultural developments of the student hacker group were not a form of resistance in the same sense as the lads', despite the roots of hacker and open source communities in the US in counter-cultural movements (e.g. Turner 2006). The white-collar work that students in the hacker group, and in computer science more generally, aspire to is also relatively privileged financially and socially. Nevertheless, like the lads, the members of the hacker group experience the reproduction of their exploitation as a form of positive self-affirmation.

As members of the hacker group advocate for the value and importance of passion, dedication, and entrepreneurialism for computer science work, they work to embody those values in their everyday practices and performances. As Andrew argued of hacker culture: "building things and learning things out of interest and just because you can should be the norm." Andrew Ross (2003), however, has explored how "no-collar" work offered by companies in places like Silicon Alley (New York) and Silicon Valley (San Francisco) – where employees are given freedom in terms of self-management, access to stock options, flat work hierarchies, and casual dress, among other "perks" – easily turns into over-work and exploitation. As Carla Freeman further elucidates:

> At the heart of the entrepreneurial ethos is a vigorous entanglement of selfhood and labor for envisioning and making one's self entails particular forms (and a particular intensity) of work. Not only do entrepreneurial labors increasingly exceed the formal boundaries of productive enterprise to include every facet of social reproduction (i.e. work 'at home' and work 'at work' bleed into one another), they seem to permeate every crevice of conscious (and even unconscious) life. (Freeman 2014, 3).

The pursuit of passion is thus equated with over-labouring, as students and employees dedicate (all of) their time and selves in the name of creativity, dedication, and passionate work. Daniel Cockayne additionally found that entrepreneurs in Silicon Valley "celebrate themselves as 'scrappy' risk takers, and though they are in positions of 'precarity,' they make carefully calculated risks that are made 'acceptable' through other forms of personal security, as well as through their passionate attachments to their work" (Cockayne 2015, 468). In other words, the insecurity and over-work – the precarity – entailed in entrepreneurship are seen both as necessary and positive aspects of work and existence.

Neoliberal affect in the form of entrepreneurial passion thus works to instrumentalize students' lives and identities such that these are funneled towards the reproduction of neoliberal values, norms, and structural inequalities, and students' successes or failures towards individual responsibility. Through a hidden curriculum of promoting independent learning and networking, discussed in the preceding chapter, students are rendered technical and they are incited to compete for academic opportunities and employment. As Freeman suggests, the emphasis on affect and "affective exchange" in the current period, also requires "that people not only be emotional but that they show their emotions in identifiable and commodifiable ways" (Freeman 2014, 3). Students are compared and judged on their grades, but also on their performances of affective qualities such as entrepreneurial passion, to demonstrate that they are the "techiest" and dedicated to independently seek out knowledge "out of interest and just because you can."

This combination of independent learning and networking with entrepreneurial passion operates as a kind of "cruel optimism" – an optimistic investment and attachment in a kind of work, selfhood, and desire or vision for the future that is central to both reproducing and surviving forms of neoliberal inequality (Berlant 2006). As Freeman suggests, drawing on Raymond Williams (1965), a "neoliberal 'structure of feeling'… is not simply a present awash in emotions through which the subject then makes sense of herself and her world, but a present and imagined future that are increasingly entangled in and through an emotional register" (Freeman 2014, 3). As discussed in the Introduction, students, entrepreneurs, and others involved in computing often seek to "change the world." "We code the future" was also the slogan used by the computer science student social club, and was displayed prominently on t-shirts designed and sold by the club, which were frequently worn in the standard fare of shorts and t-shirts among computer science students. This optimism is a way of reaching, seeking, and becoming, since *being* entrepreneurial is never fully achieved: there are always more technologies or forms of knowledge to be learned, more projects to undertake, and further dedication and passion to be demonstrated.

While Cockayne suggests that entrepreneurs are privileged precarious workers in that they are able to provide for themselves various forms of security, he also argues that their advocacy of entrepreneurial passion normalizes precarious work more broadly such that "only enterprising and self-satisfied working lives are rendered fully recognizable by systems of governance" (Cockayne 2015, 468). Moreover, this normalization reproduces the desire for such a system and its positive affirmation (Cockayne 2015, 468). Some students in Singapore, but certainly not all, may be able to rely on similar forms of security, yet students' dedication to and promotion of hacker culture similarly normalizes and renders natural particular forms of precarious (over)work as the hegemonic way of doing computer science and being a computer scientist.

As discussed in Chapter 3, through their intra-actions with code and computing technologies students and others in computer science work to render 'reality' technical and then to render natural that technical reality, creating a world for computer scientists full of possibilities for magic and play. The connection of the hacker groups' goals with national projects promoting and cultivating such entrepreneurial values and personhoods also exemplifies the ways forms of neoliberal governance are being put into operation by the Singaporean state. As these processes diffract onto students' lives, they are summoned to demonstrate their passion for operating in and creating these technical worlds, in order to become future employable persons.

6.4 Conclusion

Students in the student hacker group at Temasek University have worked to meet the calls made by Singaporean government policy and by the tech industry and become the "techiest of the techies." They have also worked to cultivate a hacker culture, themselves promoting the importance of being passionate about and interested in technology and computing to become a "good" computer scientist. I have looked particularly at the role and meaning of passion, showing how this quality has become normalized among students and professors, with those who display passion exemplifying a hegemonic subjectivity. This hegemony is maintained through mutual consent about the superior passion, dedication, and skill of those in the student hacker group, as well as through the power of employers to choose what kinds of students they want to hire.

The possible ways of being/becoming a “good” computer scientist are thus narrowed. Computing knowledge and practice are structured such that they centre on places like Silicon Valley, and in order to gain prestige, recognition, legitimation, and mobility, actors must orient towards the practices and values of such centres or meccas (Takhteyev 2012). As long as Silicon Valley and places like it are seen as the embodiments of creativity, innovation, and entrepreneurialism, they also hold the power to define the meaning of these terms. A continual sense of deficiency is thus being (re)produced by government policy, as well as transnational academic articles, reports, global indices, and discourses, justifying the recurring privileging of foreign talent and (re)producing neoliberal forms of affect and value that both demand and normalize students’ and others’ over-work and exploitation.

These conclusions (re)produce a sense that capitalism or neoliberalism possess “the predictability of a well-oiled – if somewhat complicated reproduction machine,” a critique by Dorinne Kondo of Willis’s approach (Kondo 1990, 222). This sense of mechanistic entrapment continues in next chapter, which explores how the centrality of “the woman problem” in computing works as an anti-politics machine that reproduces and masks gendered governance and inequality, but is then complicated in Chapter 8, which explores the conflicting intra-actions of neoliberal values, citizenship, gender, and student aspirations.

Chapter 7 : Anti-Politics and "Women in" Computer Science

When I spoke with students and professors about gender in computer science, most already knew that there were fewer women studying and working in the field relative to men in the US, if not in Singapore. As one professor graciously wrote in his email to the students in the course about my research: "Personally, I think this research topic is an important one, not just in North America and in Singapore, but also for the whole world. We in Singapore are slightly more 'blessed' in that our gender gap in Computing is not as skewed as that in other countries, but that also makes us an interesting case study too." The issue of numerical disparity between women and men in computing, and indeed in many Science, Technology, Engineering, and Mathematics (STEM) fields is prominently featured in the media, with almost daily articles, reports, and posts from international news media and tech blogs, and circulating on social media, about the so-called "gender gap." Near the end of my fieldwork period, for the first time several large multi-national tech companies – Google, Facebook, and Twitter, among others – released statistics enumerating the (lack of) diversity in their workforces, further propelling the issue to the forefront of media discussion and debate (e.g. "Making Google a Workplace for Everyone" 2014; Williams 2014; van Huysse 2014).

At the same time, throughout my research, many of those I spoke with argued that there was no relationship between gender and computer science – that gender was not significant or relevant to the discipline and, actually, irrelevant or secondary to skill, passion, and practice in terms of becoming a "good" computer scientist. A first-year student, for example, explained: "Anyone should be capable lah. I had no background.

My friends have no background. But, hey, we survived the first sem[ester], we are still going through it… like I said, there's just a learning curve, I think anyone can learn coding" (Ariff 2014). A second -year student similarly suggested: "computing is one of those better areas where gender actually doesn't matter. Because when people see your work and it's good, it's good. You can't really tell them female or male – no one cares" (Christien 2014). The sense that gender does not matter and that "there's just a learning curve" aligns well with values of meritocracy discussed in Chapter 5.

Professors took similar positions; in asking a professor about whether gender plays a role in entrepreneurship, he responded "Not particularly, but now that you've asked me, I never thought of it before." Another professor compared students' gender to their eye-colour, stating he did not pay attention to either trait in particular students. Gender is thus broadly seen as an issue or topic distinct from the development of technological cultures and technological knowledge. At the extreme, gender becomes completely invisible; another professor I spoke with expressed surprise when I suggested that there were fewer women than men in computer science in the US and Canada, and in Singapore. "Really?" he expressed, "there are fewer women in computer science? I don't know, nowadays maybe, but in my time it's not like this. Why is there fewer?" he questioned when I asked what he thought about the situation.

When gender was recognized as an issue or matter of concern, it was often seen as secondary in significance relative to other issues relating to technological cultures, in Singapore and elsewhere. As one organizer of the student hacker group explained, for example, when I asked whether they considered gender when organizing the group:

> Not actively because we are still grappling with the problem of how to get – I think it will be something that the club will think about more and more often as there are, as the bandwidth improves. Like, I think we are making so much progress in terms of getting people involved in to things like this, which is not an issue in the US, such that eventually you reach a level where there's already engagement, then it frees you up to think about the other secondary issues like gender representation. In fact, as a first pass, I would say that gender isn't even on the table right now because there are all these issues that haven't been addressed – there's no engagement, like people are not interested, they're not good enough, such that gender isn't even on the table properly in the way that it is now in the US because engagement's already there and then now they have to deal with the second order problems (Stanley 2013).

In this case, gender is seen as something that can be addressed separately, later – and is a thing in and of itself to be considered – in contrast to more basic and primary problems such as achieving increased participation.

This chapter explores how assumptions of what gender "is" – namely, that gender is exclusively about the dichotomy of men and women, taken as equivalent to males and females, and often specifically about women – shapes discussions and understandings of gender and computing, among students, researchers, and, often, the media-consuming public.[68] While I show that there are indeed inequalities in Singapore and in computer science relating to gender, I also consider how technical renderings of gender as "the woman problem" naturalize gender differences and (re)produce ways of approaching gender in terms of problems and solutions. I draw on Sandra Harding's (1998) tripartite theorization of gender as constituted through structural, symbolic, and individual facets. Harding's attention to gender is grounded in feminist standpoint theory with its focus on

[68] The media debates around the memo released by an ex-google employee (Damore 2017), arguing that the fewer number of women in tech compared to men can be justified, at least in part, by biological differences, are a prime example. I also published a response to these debates, which includes a brief summary of the various issues and positions (Breslin 2017).

the distinctiveness of women's subject positions in constructing knowledge and ways of knowing (Harding 1991; Hartsock 1983; Smith 1974), rather than the processes of making and doing gender – the intra-actions that work to constitute gender. Nevertheless, Harding's differentiations of the various ways gender is constituted usefully points to the different layers of intra-action in the production of gendered selfhoods and norms. This chapter focuses on structural and symbolic gendered intra-actions.

I follow Karen Barad in treating material and semiotic – or the structural and symbolic – as "inextricably fused together" (Barad 2007, 3). I see structural-symbolic intra-actions as entailing encounters among students and others with government, tech industry, media, disciplinary and academic, and societal policy, law, practice, and discourse. Students' behaviours, actions, and values relating to technology, gender, sex, sexuality, class, and citizenship and nationalism are thus governed and (re)produced through these intra-actions. I see individual gender as being about a person's sense and performance of self, produced and enacted through these structural-symbolic intra-actions, which I explore in Chapter 8. The distinction between structural-symbolic and individual gender can be seen to reproduce problematic distinctions between structure and agency, which I have sought to complicate following Barad (2003, 2007), Foucault (1982, 1990, 1997a), and Butler (1993, 1999), as discussed in Chapter 1. However, these distinctions are conceptually useful for examining the operation of power as part of gendered intra-actions and, particularly, the reproduction of hegemonic gender norms (Barad 2007, 213,219-220).

I first consider how research on gender and computing and STEM – or, more often, women in STEM – has focused on comparing and contrasting men and women within a heteronormative framework. I also consider how gender differences and inequalities are (re)produced in Singapore in relation to national policy, and as part of students' experiences at in computer science at Temasek University. In this regard, distinctions between men and women tend to associate logical thinking, analytical skill, and passion – values, behaviours, and skills discussed in Chapters 5 and 6 seen as necessary to become a "good" computer scientist – with men and masculinity. At the same time, I show how structural-symbolic gender is (re)produced as part of an anti-politics machine, continually replicating technical renderings of heteronormative gender and measurements and assessments of difference and (in)equality, while masking the symbolic-structural reproduction of gender norms and inequalities (Ferguson 1990). In turn, the next chapter will explore individual enactments of gender. As I will show, the framework of heteronormativity, dichotomized comparison of genders, and focus on "women in" computing, hides and torques the multiple, fluid, and ambiguous gender performances and identities of some students.

7.1 "Women in" Computer Science

Research on gender in relation to computing and other STEM fields has been dominated by an attention to women and, specifically, the numerical lack of women in these fields compared to men, particularly in "Western" countries. As Wendy Faulkner suggests, this work can be characterized as "women *in"* computing and STEM, compared with "women and" or "gender and" STEM (Faulkner 2000, 87–88). This research has been replicated in various forms over multiple decades, with numerous explanations offered for women's numerical underrepresentation: masculine associations with the field, women's lack of interest or ability, and women's feelings of "gender inauthenticity" caused either by masculine associations or by biological differences, among others.[69] This work has also done little to change the numbers in computer science, and in the US the number of women in the field has declined over the past decade (Ashcraft and Blithe 2010; Hayes 2010). Additionally, although numerous scholars have argued for different or more nuanced approaches to understanding the relation of gender to computing or technology, explanations and efforts continue to centre on the educational and career "pipeline" and the lack of women.[70]

Anna Vitores and Adrianna Gil- Juárez provide a detailed overview of the research on women in the computing pipeline and particularly the metaphor of the "leaky pipeline," which points to the "problem" of women leaving or "leaking" from each stage of the educational and career path (Vitores and Gil-Juárez 2015, 2–4). Critiques of this model and metaphor include: normative assumptions about linear career and life progression; the focus on supply and therefore on the "problem" of girls' or women's lack of participation; and how the act of comparison itself makes men the standard of measurement (Vitores and Gil-Juárez 2015, 5–6). I return to some of these issues in my discussion below. Despite these critiques, the metaphor continues to be used by

[69] See, for example, (Ashcraft and Blithe 2010; Camp 1997; Dambrot et al. 1985; Durndell 1991; Hill, Corbett, and St. Rose 2010; Margolis and Fisher 2002; Montanelli Jr and Mamrak 1976; Trauth, Nielsen, and Hellens 2003; Turkle 1988).
[70] See, for example, (Bray 2007; Faulkner 2001, 2000; Henwood 1998, 2000; Lagesen 2005; Landström 2007; Sturman 2009; Taylor 2002; Vitores and Gil-Juárez 2015; Wajcman 1991) for scholarship arguing for or working towards multi-faceted, intersectional, and non-binary approaches to gender and technology.

researchers, policy makers, tech companies, and media. Vitores and Gil-Juárez point out that "the persistence of the same question for years reinforces a feeling of 'stability'… or even inexorability related to the topic of women in computing" (2015, 2). This stability provides further incentive to measure and compare over time, but evidently achieves very little change.

I interrogate here the assumptions and practices that underlie these ongoing technical framings of the "problem" of women in computing. As discussed in Chapters 2 and 3, when I began my research I was determined not to make it about the numbers. Yet, I struggled while doing my fieldwork not to frame my research in that manner. Heteronormativity, and the equation of gender with women, shapes the frame of discourse – it governs the "conduct of conduct" – for understanding and researching gender and computing (Foucault 1991). In particular, the binary comparison between men and women that is a central part of this "problem" shapes the types of questions that can be asked and the types of solutions or interventions that are possible.

Feminist scholarship has long explored how masculinity and men are taken as the normative standard. Simone de Beauvoir, for example, suggests:

> The terms masculine and feminine are used symmetrically only as a matter of form, as on legal papers. In actuality the relation of the two sexes is not quite like that of two electrical poles, for man represents both the positive and the neutral, as is indicated by the common use of man to designate human beings in general; whereas woman represents only the negative, defined by limiting criteria, without reciprocity (de Beauvoir 1953).

In this way, women were made the gendered "other" in opposition to men, who are the "unmarked marker" of women's difference (Frankenberg 1993, 203). In the context of the United States, for example, "to be white, or straight, or male, or middle class is to be simultaneously ubiquitous and invisible. You're everywhere you look, you're the standard against which everyone else is measured" (Kimmel 2002, 3). In Singapore, while whiteness continues to be privileged in relation to the state's British colonial history, policies and discourses of multi-racialism that define official and equal Chinese, Malaysian, Indian, and Other "races" mask social and economic inequalities that tend to privilege Chinese persons and cultural characteristics (Huat 1998, 34–38; Li 1989, 1998a, 174–75; PuruShotam 1998, 91; Velayutham 2016).

Many scholars have also explored how men and masculinity are privileged in computer science and technological fields (Ensmenger 2015; Faulkner 2000; Haigh 2010; Håpnes and Rasmussen 1991; Lewis 2006; Mellström 2002, 2003). In 2012, news media and academic scholarship began using the term "brogrammer" to refer to a recent narrow and misogynistic version of masculinity that has become prominent, particularly in Silicon Valley, involving frat-house-like sociality and performances of technical heroism, generally among men, to the exclusion of other identities and persons (Hicks 2013; MacMillan 2012). Yet, this term is just the latest illustration of a historical predominance of men in computer science that began in the 1980s in the US (see, for example, Ensmenger 2010b).

In her detailed discussion about the intersection of feminism and technology, Judy Wajcman goes so far to refer to "technology as masculine culture" to highlight the ways that technology – and what is seen as technology – is intimately intertwined with masculine performances and social relations (Wajcman 1991, 137–61). The association between masculinity and technology has been seen as quite "durable" across time, and space (Faulkner 2000). At the same time, many scholars have also worked to deconstruct this relationship, offering counter-examples and showing the significant contributions of women to technology throughout history and in different places (Abbate 2012; Faulkner 2000; Lagesen 2008; Light 1999; Mellström 2009; Misa 2010). As such, these masculine associations are, in many ways, broad generalizations that often do not fully fit particular contexts across time or across cultures. Nevertheless, in terms of symbolic associations among students in computer science in Singapore, which I discuss below, as well as in research about gender in computer science, men and "masculine culture" are often taken as the default and invisible norm.

As discussed in Chapter 4, professors at Temasek University tended towards using masculine pronouns more often than feminine pronouns, although gender-neutral choices were most prevalent and this tendency was sometimes recognized by professors, as prompted by my presence. Beyond pronouns, particular renderings of masculinity and femininity are also seen in the algorithmic stories and framing of problems. As Wagner outlines relative to Gale-Shapley solution to the Stable Marriage algorithms, it rests on a treatment of men as the initiators and proposers of marriage, and women as passive recipients who do not even ultimately consent affirmatively to their pairings (Wagner 2009, 294).[71] A wide variety of research has also considered the ways technologies are

[71] This pattern is also clearly seen in stories about algorithms and the presentation of algorithmic problems in computer science texts, discussed in Chapter 4, which reproduce women's exclusion from technological knowledge and practice, as well as particular renderings of masculinity and femininity. Education scholars have shown how gendered representations in textbooks operate as a hidden curriculum reproducing gendered and class-based norms and values (Apple 1992; Clawson 2000; Kho 2013, 91–156). In this regard, in Donald Knuth's four volumes of *The Art of Computer Programming* (1997a, 1997b, 1998, 2005), the words "she," "her," "hers," "woman," "women," "female," "females" appear 50 times, referring to: generic persons like "Alice" from algorithms discussed in Chapter 4; the inventor or author of an algorithm,

gendered according to stereotypes and assumptions about masculinity and femininity, "co-producing" gendered technologies and gender norms in the process (Berg 1999; Brahnam, Karanikas, and Weaver 2011; Faulkner 2001, 2000; Oudshoorn, Rommes, and Stienstra 2004; van Oost 2003; Wajcman 1991). Ellen van Oost (2003), for example, shows how the men's shavers developed in the 1970s and 1980s projected and embodied particular ideas about gender – that men want to tinker and women prefer simplicity. By providing options for adjustability and incorporating monitoring features on electronic displays, men's shavers both repeated and reinforced the idea that men like and are good at using technology. In comparison, the women's shavers hid the technology by hiding screws and providing no options for adjustment. They were also marketed as a cosmetic product, suggesting shavers for women are not technological things and, by extension, women should not need to use or understand technologies.[72]

Scholars have also explored how computers are feminized and treated as objects, akin to women both metaphorically and physically, as well as akin to slaves or servants,

---

theorem, or technique (three times); and as objects or categories searched for or sought after. In the latter case, one such problem suggests: "In an enrollment file that contains information about students at a university, it may be desirable to search for all sophomores from Ohio who are not majoring in mathematics or statistics; or to search for all unmarried French-speaking graduate student women," and then in a different search problem in a different volume to "find all blue-eyed blonde women of ages 21 through 23." In contrast, the words "he," "him," "his," "man," "men" appear 681 times. Men are referred to as: authors and creators, including Knuth himself who refers to himself in the third person; competitors or players in a game; and as actors in scenarios, including one where the players in a game are seeking to determine the order of other men being "brutally executed" in order to save one's own life. Men and masculinity are thereby portrayed as active and competitive, and sometimes involved in violence. Knuth's texts can be seen as exceptional and the first volume was published in 1968, prior to most research and activism around issues of women and gender in computer science. However, the first volume is also now in its third edition, most recently published in 1997 (the edition I used for my searches). Knuth's texts were not assigned in any of the courses I observed at Temasek University, although Knuth himself was referred to as a "pioneer" of mathematics in computer science by one professor, and another suggested that his books were "the" books for students to buy. The CLRS text reveals much more balanced use of pronouns and gendered references, with many of their problems based around both men and women "Professors."

[72] I discuss this example, and others, in (Breslin and Wadhwa 2014b; 2017).

thereby intersecting with issues of race and class (Brahnam, Karanikas, and Weaver 2011; Chasin 1995). At the same time, researchers have shown how, when creating programs and technologies, designers, developers, and programmers often rely on what has been termed the "I-methodology," using themselves (often men, white, and educated persons) as model users and implicitly embedding norms of masculinity in programs and technologies (Berg 1999; Forsythe 2001; Oudshoorn, Rommes, and Stienstra 2004). Thus, men and masculinity take the position of, or are portrayed as, active developers – creators of technology – whereas women are seen as the passive objects of men's desires, and creative production, or passive users of "cosmetic products" rather than "technologies."

Following a similar logic, research about gender in computer science often takes masculinity and men as the default or norm in measuring qualities such as confidence and motivation (Beyer et al. 2003; Cech et al. 2011; Shashaani 1997; Turkle 1988). In this research, women's success (or lack thereof) is assessed based on whether they perform the same level of confidence or motivation as men, without questioning the desirability, value, form, or implications of such qualities. This approach was popularized by Sheryl Sandberg, the Chief Operating Officer of Facebook at the time of writing, who wrote the book *Lean In: Women, Work, and the Will to Lead* which called upon women to improve themselves and their careers through individual perseverance, self-determination, and self-advocacy (Sandberg 2013), thereby promoting neoliberal femininity and feminism (Hooks 2013; McRobbie 2009). Conversely, Wajcman points out how some feminist approaches to technology have relied on essentialist conceptions of women's difference from men, while valorizing women's qualities such as "humanism, pacifism, nurturance and spiritual development" (Wajcman 1991, 9). These values were taken as the antidote to male and masculine science, as based on dominance and appropriation of nature, and invested in mastery and detachment (e.g. Keller 1985; Wajcman 1991, 6–7).

There are a variety of problems with these essentialist associations with masculinity and femininity, including: their failure to recognize historical and cultural specificities and differences in gendered metaphors and qualities; the social construction of nature and what is "natural"; and the extent to which the qualities attributed to women relate to their historical subordination (Wajcman 1991, 8–10). Nonetheless, such approaches to gender – built upon the assumption of heteronormative gender binaries – are additionally baked into the use of statistical comparisons. Statistical binary comparisons are themselves a process of rendering technical and countable, a way of capturing information about categories of persons, namely "men" and "women" – or, more often, "male" and "female" – and measuring and assessing them. As Ian Hacking has suggested, this process has a "looping effect" where "sometimes, our sciences create kinds of people that in a certain sense did not exist before. I call this 'making up people'" (Hacking 2006). That is, assigning people to categories such as "male" and "female" works to summon people to embody and inhabit those categories, making up people in a certain way that does not otherwise exist. I consider in the next chapter how students contend with classificatory norms relating to gender.

This rendering, however, also produces "problems" that need to be solved in relation to the measured differences between men and women. As discussed in Chapter 3, students are initiated into systems of thought and practice whereby they learn how reality can be rendered technical and framed in terms of problems with technical solutions. Such problems only need to be made explicit, broken down into smaller parts, each with particular, logical, and detailed steps to solve according to the various algorithmic, data structure, and programming language practices that students have learned. Taking literal numerical equality as a desirable goal, differences in measurements are a clear problem to which "solutions" can be found, the success of which can then be measured by those same comparisons between men and women, a process I explore further below. These renderings of gender thus shape how masculinity and femininity are measured and defined, the types of problems that are considered, and what is seen as a problem.

With an understanding of the ways a heteronormative framework and practices of rendering technical shape gendered discourse and research, it is nevertheless important to acknowledge and recognize the significant challenges and issues that women as a "group" do indeed face in computer science and in STEM fields more broadly. Researchers, and women in STEM fields themselves, have repeatedly shown the ways that those identified as women are made invisible, discriminated against, and harassed in these fields (Bort 2016; Fowler 2017; Hill, Corbett, and St. Rose 2010; Nafus 2011; Powell, Bagilhole, and Dainty 2009; Sami 2015). Discrimination in various forms has become particularly prominent in relation to Silicon Valley, both in terms of the widespread media coverage that has emerged about it and the apparent severity of harassment and exclusion that occurs. Researchers have additionally explored how practices that work to reinforce masculine social and cultural norms and the predominance of men in computer science and other STEM fields have a broad and lengthy history (see, for example, Forsythe 2001; Haraway 1997; Harding 1986; Hicks 2017; Keller 1985).

In this regard, through social practice and discourse, women are treated as a distinct "group" and the ways this "groupness" is constituted, reproduced, and treated can be and has been a significant and meaningful area of research. As such, against a backdrop of assertions by many in computer science and in Singapore that there are no barriers facing women, I now consider the construction of women's gender roles, as well as the ways femininities and masculinities are structured and produced in unequal ways by government discourse and policy in Singapore.

7.2 National Gender in Singapore

Singapore is portrayed nationally and internationally as a story of success in terms of economic and national development, including in terms of women's education and economic participation. That is, Singapore is recognized as having achieved significant measures of equality between men and women, with Singapore ranked 11th in the world on the UN Gender Inequality Index, for 2015 (United Nations Development Programme (UNDP) 2015, 224). Women's literacy increased from 34% in 1957 to 93.8% in 2010. Approximately 93.6% of women aged 25-34 had achieved at least secondary education, only 1.9% lower than the rate for men (Kho 2013, 2–3). The year of my fieldwork, women comprised 50.5% of all students enrolled at Temasek University. As discussed above, while these statistics are problematic for reproducing gender binaries and differences, they are also widely used nationally and internationally as a means of assessing equality between "opposite" genders, which is why I cite them here.

A variety of government policies and practices in early post-colonial Singapore indeed treated and cultivated men and women in equal ways. Population policies prior to the 1980s, such as "Girl or Boy, Two Is Enough," promoted the idea that girls and boys should be treated and valued equally, although these policies were themselves a response to "traditional" values that privileged sons over daughters (Kho 2013, 53). In interviewing different multiple generations of women in Chinese families, Elspeth Graham et al. (2002) explore how the parent generation largely showed mixed preferences for girl and boy children, in contrast to the grandparent generation who largely favoured boys based on "traditional" and patriarchal family values (Graham et al. 2002, 83). While the authors explore different generations of women's fertility decisions and perspectives in relation to a failure of policies beginning in the 1980s that promote pronatalism and "Asian Values," their responses about gender preference might also indicate a success of previous anti-natalist policies that promoted gender equality. National education in 1968 also sought to produce citizens who were "rugged, vigorous, intelligent and capable, endowed with a strong sense of patriotism, possessing high standards of education" with no differentiation between girls and boys (Speech by Dr Tay Eng Soon, Minister of Education, cited in Kho 2013, 54). In 1966, the national need for skilled industrial labour also led to the promotion of technical education for both girls and boys. These efforts included campaigns following the introduction of a formal curriculum in 1968 that argued that girls are suited to technical careers and provided career-guidance specifically for girls (Kho 2013, 57).

Efforts were thus being made to shift "traditional" ideals of femininity based in "passivity, docility, and submission" and to encourage girls to be more like boys, who were "active, rugged, and tough" (Kho 2013, 54, 56). These policies had the effect of promoting the idea that men and women in Singapore have achieved equality. As Singaporean education scholar Ee Moi Kho points out:

> [I]t appears that access to modern education and job opportunities has empowered many Singapore women. For many in Singapore, gender issues are not of significant concern because the ruling party's declared policy of equal opportunity has allowed women to achieve much in society. Women in Singapore seem to have achieved status equal to that of men, and the rise of women's status is often attributed to their access to education (Kho 2013, 4).

Yet, despite these perceptions, Kho points to a variety of statistics relating to women's current economic and family roles, including wage disparity; decline in women's labour participation after the age of 29; disproportionate responsibilities for childcare, housework and other forms of caregiving; and a lack of women in leadership roles, to suggest that statements of equality are not all that they seem (Kho 2013, 6–7).

Additionally, despite wide-scale national education for all citizens and the necessity (for women, families, and the state) of women's participation in the economy, the Singaporean state has also treated motherhood as women's primary role. Women were thus supposed to maintain their roles as home makers alongside their participation in the workforce. In 1983 then Prime Minister Lee Kuan Yew's now infamous National Day Rally speech addressed issues of low birth-rates, particularly among educated women:

> Our most valuable asset is in the ability of our people. Yet we are frittering away this asset through the unintended consequences of changes in our education policy and equal career opportunities for women. This has affected their traditional role as mothers. It is too late for us to reverse our policies and have our women go back to their primary role as mothers, the creators and protectors of our next generation. Our women will not stand for it. And anyway they have already become too important a factor in the economy. Therefore, we must further amend our policies, and try to reshape our demographic configuration so that our better-educated women will have more children to be adequately represented in the next generation (Lee 1983: 11).

Regulations, such as the stipulation that no more than 1/3 of students admitted to medicine at the National University of Singapore beginning in 1979 be women, illustrate the ways that these ideas were enacted (Kho 2013, 64, 80). As Lee stated in his off the cuff comments in this speech “you just can’t be doing a full-time heavy job like that of a doctor or engineer and run a home and bring up children. It is tough” (Lee 1983:11). Study of home economics (previously domestic science) was made compulsory for girls in 1985, which also meant that they could no longer enroll in technical studies courses (Kho 2013, 67).[73] Women’s educational and economic opportunities were also said by the government to depend on national economic growth (Kho 2013, 56–60). In other words, should the economy shift so that women’s participation was no longer required, policies promoting their inclusion could also shift.

Heng and Devan offer a somewhat polemic but poignant critique of Singapore’s pro-natalist policies at this time, and the implications of these policies in terms of women’s roles in reproducing state power:

> The demand that women serve the nation biologically, with their bodies – that they take on themselves, and submit themselves to, the public reproduction of

[73] Prior to 1985, domestic science was offered only to girls, but girls could opt to take technical education instead (albeit only 50% of girls, from 1968 to 1977) (Kho 2013, 57–60).

> nationalism in the most private medium possible, forcefully reveals the anxious relationship, in the fantasies obsessing state patriarchy, between reproducing power and the power to reproduce (Heng and Devan 1995, 201).

Femininity thus was promoted as associated with domestic concerns such as cooking, sewing, and caring for babies and children, contrary to earlier policies that called upon both girls and boys to be economic contributors with “rugged” and technical skills (Kho 2013, 67–68).

For men, however, one facet of life that continues to define their role in Singapore society is the requirement for National Service. Since 1967 all male Singaporeans and Permanent Residents are required to train and serve for two years, with enlistment required between the ages of 16.5 and 21. Following this period of service, recruits become Operationally Ready National Servicemen until the age of 40 or 50 and, depending on rank and special skills, must serve periodically each year. Moreover, at the time when girls and women were being (re)trained in a domestic femininity, army training through the National Cadet Corps in the 1980s was being revised to make boys and men more physically tough and fit (Kho 2013, 70).

As discussed in Chapter 6, the early to mid-1980s was also precisely the time when the government was working to grow national computing and IT industry and expertise, even as girls and women were being called upon and pushed towards more “traditionally” feminine roles and fields such as “literature, music, and dance” (Kho 2013, 68). The contradiction between the government’s need for more technical experts and their push of women away from technical subjects towards domestic roles led to conflicting policies and discourses from the 1980s into the 2000s. In 1981, an article from the Straits Times suggested that "We can also optimise our scarce manpower by encouraging more women and 'prematurely' retired persons to work." (The Straits Times 1981c). Yet, later that year, they ran an article about "technofear" that emphasized women's fear of using new household technologies and quoting Dr. Frank Osman, a British industrial psychologist, suggesting that "there are cases where even the thought of having to operate an expensive new washing machine brings housewives out in a cold sweat" (The Straits Times 1981a). Women were being cajoled to join the workforce and contribute to the nation as technical experts, while the media was repeating common tropes of women (and, to a lesser extent, men's) fears of technologies and lack of technical skill and ability.

Despite the mixed messaging, women seemed to participate enthusiastically in the tech industry and became a significant proportion of programmers and computer professionals in the 1980s. In 1982, an article suggested that "no computer firm in Singapore employed a woman engineer," yet in 1984 a total of 91 women graduated from the Department of Information Systems & Computer Science at the National University of Singapore compared to only 19 men (Kheng 1989, 311; The Straits Times 1982). While 1984 was an exceptional year for the number of women graduates, from 1983 to 1987 the number of women graduating was close to equal to, or even more than, that of men (the lowest proportion of women was in 1985, with 42% of graduates being women) (Kheng 1989, 311).

It is interesting to note that while enrollment and graduation proportions of women were relatively high in computer science, the opposite was true in engineering, where university programs experienced significant shortfalls of women in their programs (Kho 2013, 71).[74] The specific and concerted national governmental push to grow the computer industry in Singapore and train local computer professionals, as well as the good prospects for career advancement and salary, may partially explain this difference (Kheng 1990, 19). In considering this unusual trend relative to the comparative lack of women in British computer science at the time, Lorna Uden suggests that it is also possible that girls and women prefer computing relative to engineering because it is not associated with "getting their hands dirty" (Lovegrove and Segal 1991, 388). Singapore's institutes of higher education seemed to have the same impression at the time, as a Straits Times article from 1988 commented that the National University of Singapore and Nanyang Technological Institute were both "anxious to dispel the traditional stereotype of an engineer as a brawny male sweating it out under the hot sun, embroiled in dirty and heavy work" (The Straits Times 1988). Viviane A. Lagesen has made a similar argument about the relatively high numbers of women compared to men in computer science in Malaysia. Among the students she interviewed, engineering was seen as more "masculine" since it required work outdoors and contact with men labourers. In contrast, computer science was associated with "office work," which was done indoors and considered relatively secure and safe (Lagesen 2008, 14).

[74] Computer professionals are often referred to as engineers, as seems to be the case in the 1981 Straits Times article on the number of "engineers" in computer firms. Yet, as educational disciplines, computer science and engineering (particularly in the 1980s) are distinct, where engineering generally refers to civil, mechanical, industrial, and electrical engineering whereas computer science focuses on computer theory and software with some connections to computer hardware.

Yet, even with national efforts to build computer industry and expertise in Singapore, some effort by educational institutes to change perceptions of engineering, and women's high rates of involvement in computing education and industry, there was little concerted effort by the government to challenge the symbolic-structural factors that limited women's participation in engineering and technical fields. This situation persisted despite the government's active involvement in shaping gender roles and symbolism in earlier years, and the government asserted that maintaining social stability was a priority over promoting a particular gender ideology (Kho 2013, 60, 73).[75]

Policies began to change in the 1990s when both home economics and design and technology became compulsory for all students, a change that was fully implemented by 1998, and the government began promoting shared responsibilities in terms of managing the household (Kho 2013, 74–76). School textbooks were revised beginning in 1987 to reflect this new approach to gender roles and family (Kho 2013, 76). There were also ongoing efforts to attract more women to science and engineering.[76] Education Minister Lee Yock Suan stated in 1996 that the government was attempting to change perceptions about engineering in schools and make science and engineering university courses more attractive, with training for school principals and career guidance coordinators on careers

---

[75] Gender ideology here refers specifically to gender roles – along with the acceptance of homosexuality and transgender persons – that are seen as counter to "traditional" values and roles of femininity and masculinity, which the government treats as non-ideological.

[76] There were no comparative efforts to attract women to computer science covered in The Straits Times, although the relative number of women graduating from computer science at the National University of Singapore had dropped to a range between 35% and 44% between 1994 and 1999. These numbers are notably lower overall than in the 1980s but comparatively high to fields such as Science and Engineering in Singapore which consisted of approximately 30% at both the National University of Singapore and Nanyang Technological University (Nirmala 1996).

in these fields (Nirmala 1996). The minister suggested that while science and engineering might be perceived as for “weird individuals pursuing esoteric research in ivory towers” or for “traditionally in hard hats and boots, doing robust fieldwork,” respectively, this work “involves brain-power, not brawn power” and, for women, “entering these professions did not decrease their femininity” (Nirmala 1996). That is, in some cases, schools were working to train men and women in equal numbers, and treat them equally, encouraging women to join fields historically dominated by men such as science and engineering. At the same time, these efforts to encourage women’s participation relied on the idea that femininity was at odds with “esoteric research” and “robust fieldwork.”

Meanwhile, the government continued to maintain the need for a patriarchal and gender divided society, as expressed and performed in a wide variety of ways. As the first woman Minister of State Dr Seet Ai Mee explained in an interview with the Straits Times in 1992:

> Economically, the country needs women to go to work, yet we say we want to strengthen the family support for the children, too… I accept that there are conflicting messages, but I believe that there are women who can handle both career and family and be superwomen. However, there are some women who can handle only one area. These, then, should concentrate on taking care of their children well (The Straits Times 1992, 2).

Women’s roles as wives and care-givers within the context of a family was further emphasized by then Prime Minister Goh Chok Tong in his 1994 National Day Rally speech, where he explained the policy of making the medical benefits of women in the civil-service dependent on their husbands since, “changing the rule will alter the balance of responsibility between man and woman in the family. Asian society has always held the man responsible for the child he fathered. He is the primary provider, not his wife”

(Goh 1994, 43). Women and men were clearly being positioned in different roles by the government, with motherhood and family care largely the domain of women, whereas financially supporting the (heterosexual) family remained the responsibility of men (Kho 2013, 77–79). Along similar lines, foreign husbands of Singaporean women remained ineligible for permanent residency, whereas the foreign wives of Singaporean men were eligible.

Some schools enacted policies that similarly conflicted with efforts towards equality, or gender neutrality, where secondary school girls, for example, were discouraged or forbidden to wear short hair since “such close crops are unfeminine and unbecoming of schoolgirls,” while similarly banning “hair reaching the ears or collars” among boys. (Goh 1993, 25). This last issue, relating to physical performances of gender, continues to the present, although not without debate. For example, when three secondary school girls shaved their hair for charity, they were told to wear a wig by the principal, citing school policy that disallowed “unfeminine” hairstyles (Chua 2013). In 2016, the career centre at Temasek University also facilitated a workshop entitled “The Right Image: Make-up for Ladies,” run by a makeup artist for the makeup company M-A-C. The workshop was promoted as to equip “ladies” “with insider tips to look your best at interviews and networking events.” As one commentator on the university confession site argued, however, “it feels like for girls, your worth, credibility and capability of performing a job well is somehow tied to your face and beauty.”

This multiplicity in government discourse and practice across various institutions continued into the 2000s, but with new moves to eliminate structural-symbolic differences between men and women in government policy. For example, the quota on women medical students was reviewed and lifted in 2002 and women civil-servants were given the same medical benefits as men in 2004 (Kho 2013, 80–81). Yet, the government's push for Singapore to become a "global city" in the 2000s, preceded by earlier efforts to grow Singapore's global entrepreneurial and technological capacities, also operates through gendered (as well as class-based and racialized) discourses and practices (Cheng, Yeoh, and Zhang 2014; Liew 2014; Yeoh, Huang, and Willis 2000). In exploring practices of Singaporean men marrying foreign women, Yi'En Cheng et al. explore how "local men are facing increasing pressure to become highly educated, globally mobile, financially able and successful career men" (Cheng, Yeoh, and Zhang 2014, 5). At the same time, the discourses and practices surrounding Singapore as a "global city" often position women as wives of "global talent," as maintainers and reproducers of family values and care, or as providers of domestic labour, rather than mobile talents themselves (Yeoh, Huang, and Willis 2000). These distinctions are reproduced in research that has largely focused on the "masculine global city" (Yeoh, Huang, and Willis 2000, 156).

Like the confessional commentator quoted above, Singaporean students I spoke with recognized some of these gendered divisions and norms. One student, Naomi, who was particularly reflective about gendered social practices, discussed still-prevalent stereotypes and assumptions about gender roles for men and women, but also how these roles conflicted with messages taught in schools and personal goals of different women:

> **Naomi**: Not much gender issues, aside from some more traditional people, those are the people that would think that girls are not that great at computing.

> **Sam**: Do you think that's sort of more of a traditional way of looking at things, rather than modern?
> **N**: Yeah. Because once you go through school you will notice that girls and guys there really isn't that much difference in grades. And guys, and once they come out from army they know this one fact, is that they have two years of doing absolutely nothing and the girls are fresh out from poly[technic] or uni[versity] and they know that they are really disadvantaged. So yeah, they don't take girls lightly.
> **S**: Yeah, I guess that makes sense. And have you ever encountered any issues, in classes or friends or anyone?
> **N**: No, I just find a trend that guys might try harder than girls, I have no clue why… It could be the fact that in Singapore, guys are taught that they will most likely be the sole provider for the family. And girls are like, right now they know that they can have a career, but I think when I was young there was more emphasis put on like you might have a family, it's not so much about career. But, in the course of education system, they will still tell you to do well. It's a bit odd.
> **S**: Do you think there's a lot girls who still have that emphasis on family rather than pursuing a career here or?
> **N**: I've never met any. But I heard my friend say that some of her friends have more emphasis on family. I don't think in computing. The fact that you actually take up an IT career means that the girl is quite focused on a job. Oh yes, and other girls I know are usually Indians. They really really like computer engineering, and they like to try very hard. (Naomi 2014)

Naomi also talked about how government policies for maternal leave put women at a disadvantage because there were limited protections, at least in practice, that prevented discrimination in hiring when employers would assume that women were more focused on their families or would quickly leave their jobs to raise a family. These discussions reveal ongoing assumptions that women are responsible for child and family care, whereas mobile, entrepreneurial, and professional men who provide for their families but who are not largely responsible for domestic work are taken as the norm in professional business.

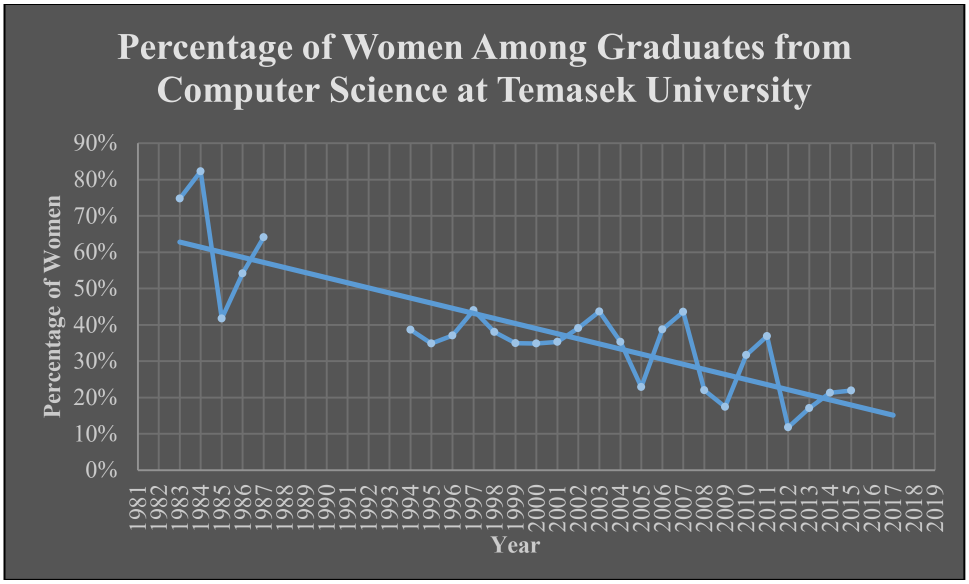

Figure 7-1: Percentage of women graduating from CS at Temasek University

The persistence of these structural-symbolic disadvantages is reflected in a slow decline in the number of women enrolled in computer science at Temasek University, shown in Figure 7-1. It seems likely that the increasing and repeated focus on computer science as associated with global entrepreneurship, rather than national development, may also have contributed to this trend. Note, however, how Naomi began her discussion by saying “not much gender issues” and, as with many others I spoke with, such gender issues were at least partially framed as a comparison between the behaviours and qualities of men and women. From the discussion above, it is nonetheless clear that there are structural-symbolic gendered differences and inequalities produced and enacted in Singapore in relation to computer science and more broadly. I explore further how renderings of masculinities and femininities are discussed by students and professors in computer science at Temasek University, where a framework of gender difference and comparison leaves little room for considering the ways structural-symbolic framings of gender themselves are a source of “gender issues.”

7.3 Gender in Computer Science at Temasek University

I briefly discuss here some of the ways that men and women were distinguished by participants and the various characteristics attributed to or performed in relation to masculinity or femininity. This is necessarily a brief overview of what was in many ways a complex and multivalent gendered environment, despite the overarching and pervasive heteronormative framework, due to the variety of persons of different nationalities, "races," religions, and classes involved. At the same time, as mentioned above, throughout my research, many of those I spoke with argued there was no relationship between gender and computer science – that gender was not significant or relevant to the discipline, and irrelevant or secondary to skill, passion, and practice in becoming and being a "good" computer scientist. I want to take these claims seriously and not simply dismiss them as a form of blind privilege or false consciousness.

It is possible that students had yet to face most forms of gender discrimination given their early career stage, although research in the US suggests barriers and discrimination relating to gender and computing occurs in that context even among primary school children (Ashcraft, Eger, and Friend 2012). Yet, the operation of power, and particularly the constitution of gendered discourse constructed on the foundation of binary equal but different genders alongside the centrality of rendering reality into technical problems and solutions in computer science, also shape frames of discourse. In particular, students' discussions in many ways reproduced structural-symbolic gender as expressed and enacted in transnational research on women in computing, and in Singaporean national discourse and policy. In this regard, it is also clear from students' discussions that gender *does matter*, but often in ways not easily captured by discourses and practices of numerical and binary comparison.

When asking about gender norms and roles in interviews, many of those I spoke with often turned to comparing and contrasting the qualities of women versus men. I began to ask about these differences in my interviews because it seemed one of the few ways many people were willing to speak about gender norms without an obvious sense of confusion or discomfort. My doing so evidently accepted and reproduced the naturalness of such distinctions, and certainly led interviewees towards that direction of discussion. At the same time, as discussed above, heteronormativity was widely taken as a given. Some students would also nevertheless respond that they saw no difference in men's and women's abilities or in the ways they do computer science. In general, students and professors would also openly contest or reframe my questions if they felt that I was asking leading questions or not addressing topics they found significant.

Several students commented that men and women have different abilities in terms of physical strength – that is, it should not be forgotten that men and women are different – but since physical strength is not required, there was no reason women cannot do computer science. Second year Singaporean student, Xiaowen, commented:

> Like biologically girls are physically not as strong as guys, so in terms of maybe manual labour they are not going to be able to be as fast as guys or running or sports… I don't know if it has anything to do with science, like their right brain analytical, there not so sure lah, not so sure. But one guy one girl same time same amount of practice should be the same (Xiaowen 2014).

In other cases, however, contrary to claims that gender is not relevant, some students would begin to speculate or reflect on the different qualities demonstrated by men and women in relation to doing computer science. A second year Chinese student, Qiaohui, for example suggested:

> I think generally males, boys, are more capable, are more natural at solving problems and applying analytical thinking, these sort of things. And apparently are more devoted, like they devote really a lot of time just to crack a problem. But for me, I feel constantly distracted sometimes, and it's quite of a headache for me. Sometimes I find it really hard to really focus… It's a female, I think it's definitely a female/girl issue. Yeah, so this is not a part which will bother a boy, so that's what I think (Qiaohui 2013).

Another student in her third year suggested that men were more "passionate" but women were more "hard working" offering a contrast, but one that seemed to support the idea of equal but differing capabilities. At the same time, some saw women as possessing certain qualities that are ideal for computer science: "Actually it's kind of weird, because I like to think that girls are generally more meticulous, so they can do well at this, just off the top of my head," Ping, a fourth-year Singaporean student expressed when I asked him why he thought there might be fewer women in computer science (Ping 2014). These comparisons of men's and women's qualities closely resemble those made in academic literature exploring masculine and feminine qualities in relation to women's abilities and interests in relation to computer science, mathematics, and software use (Beckwith et al. 2005; Beckwith and Burnett 2004; Burnett et al. 2010; Dambrot et al. 1985; Tan, Czerwinski, and Robertson 2003; Turkle 1988).[77]

Interest in computing was often perceived as based in a natural difference between men and women, and used as an explanation for the fewer numbers of women

[77] In relation to conceptions of gender/sex differences in terms of ability, it is important to note that after an extensive review of the related literature on mathematical ability and participation in science, Stephen J. Ceci et al. (2009) conclude that social and cultural factors are much more relevant in women's decisions to pursue science than biological factors such as hormonal differences or brain structure.

pursuing computer science and related disciplines. When I asked students if they had any explanations or personal theories about why there might be fewer women in computer science, difference in interest was a common answer. "Like my sister – might be a stereotype – but she has no interest in knowing how the computer works, she's just happy it works. If that's your approach to computers, it's a bit difficult to be doing computer science. I can't for the life of me make her interested," Ping also explained, saying his sister planned to study political science instead (Ping 2014). As I discuss in Chapter 2, often students and others would ask me why I left computer science, which was partially, although certainly not exclusively, related to issues of interest. My response was then taken as an (or the) answer to why women leave computer science. While interest is often taken as a natural and essential trait, interest is not a neutral category and claiming differences in interest often masks exclusion and discrimination (Abbate 2012, 39–72; Cheryan et al. 2013; Ensmenger 2010a; Foor et al. 2013; Macdonald 2014).

In this regard, students would more often than not associate qualities such as "passion" and "analytical thinking" or "logic" to men, qualities that were discussed in Chapters 3, 5, and 6 as the widely perceived qualities required to become a "good" computer scientist. Additionally, the majority of core members of the student hacker group, who come to represent what it means to become a "good" computer scientist, perform as men. These performances reinforce the associations between masculinity and qualities such as passion and logic, which form part of the basis of research that suggests technology is part of "masculine culture" and closely intertwined with masculinity (Faulkner 2000; Wajcman 1991). It also illustrates how binary comparisons and essentialist constructions of masculinity and femininity support women's subordination or discrimination in computing.

As one first-year student from Singapore loosely suggested in his discussions about learning programming and barriers:

> **Sam**: What do you think it takes to make somebody good at being a computer scientist?
> **Paul**: Um, I guess it's more of logical thinking than good memory skills, because you don't really need good memory to get the syntax right, you just need practice and to really solve the problems, you need logical thinking. Yeah, algorithms, that's why I think that's what is stressed in the courses as well, yeah, logical thinking, breaking down the steps.
> **S**: Do you think it's something anyone can learn to do?
> **P**: Definitely. Some people are gifted, but it still can be picked up lah I guess.
> **S**: Ok, would there be any particular barriers you think to somebody learning?
> **P**: If you are a very emotional person. No no no, I think, barriers – I mean, studies have shown like guys can view figures better, or something like that. Maybe that's the only barrier.
> …
> **S**: And do you think being "male," do you think that's shaped your experiences so far in any way?
> **P**: I don't know, I still think it's logical thinking, only that aspect. Yeah, it might give me an advantage I don't know. Because I have heard from a girl's point of view, like is it really the logical thing that's hurting them, or are they just being lazy, I don't know. (Paul 2014)

These associations cross lines of national citizenship, as I had a similar if more explicit discussion with a second-year student from China:

> **Sam**: Do you have any ideas, personal theories, as to why there are few women in computer science?
> **Bao Jia**: First I think it's because of the nature of human - I think girls more like arts, right? Some say male is more logic… Yeah, so I think for the nature of human, I think boys tend to be more interested in logic things, which is programming. Girls, by nature, they are not. And also, I think if you say programmer, it's usually a man's job, it's not for female. That's also common knowledge.
> **S**: So, it's seen as a men's job to program?
> **B**: Yes, it's seen as that. If you say 'oh you are a programmer,' he must think, oh he's a man. So maybe some girls don't want to challenge themselves. Also, I

> think the job for the programmer is not suitable for girls. To sit down in front of the [computer and monitor], I don't think it's good for their body, their health.
> **S**: Do you think it's good for guys' health?
> **B**: It's not, but I think worse for girls because they are going to give birth to babies, so I think it's not good for them. My perspective, my opinion. (Bao Jia 2013)

This student's responses were exceptional in that they suggest he sees computer science as essentially a field more suitable for men or males, in terms of preference or interest, gender association, and physical demands. Nevertheless, as seen above, stereotypes about sex, gender, computing ability, and desire were organized in terms of a contrast between men and women. This occurs even as both students discussed above commented that they had women colleagues who they worked with for assignments or in class and who they saw as equally capable of doing computer science as them, or as men in general.

As suggested above, this contrast focuses most significantly on women's qualities and how they compare to men. When I asked about gender, or talked about my research about gender, my questions or discussions were generally taken to be about women. As one professor asked after I gave my in-class introduction about my research: "you're interested in girls mainly?" In this case and others I would respond that I am interested in speaking with everyone, but as I discuss in Chapter 2, it rarely seemed appropriate or possible to speak about gender theory in significant depth. Similarly, when I would ask students and professors who identified as male/men whether they thought their gender influenced their studies or experiences in computer science in any way, they would often answer no, and then sometimes follow up and suggest that I should ask women students because they might have something to say. "I think you need to talk to more girls in computing, I wouldn't be able to give you anything useful because it's just not something

I really think about" one senior student told me, for example, likely indicating that gender is not something the he and other men/male students *have* to think about, as "unmarked markers" (Frankenberg 1993, 203).

7.3.1 Enacting Structural-Symbolic Gender

Several students, professors, and administrators that I spoke with suggested that, where gender assumptions and associations like those discussed above existed, they were just "perceptions" with no basis in reality or fact. Yet, the intra-actions entailed in (re)producing structural-symbolic gender have diffractive effects. When I asked about the role of gender in computer science, one TA commented: "It's a fact right, girls have problems in doing mathematics. You can see this girl is having a problem," he said pointing to an empty seat that a woman student had occupied an hour earlier during a tutorial. Yet, he continued, "but that boy is able to understand," pointing to another previously occupied seat. "So I think it's general, right. Even my cousins, I mean, the girl is not able to understand. So I think it's general, girls have problems in mathematics. I don't know, but I have seen this many times" (Arjun 2013). That is, one woman's difficulties in mathematics and computing are taken as representing the abilities of all women everywhere, an occurrence which has been termed the "stereotype threat" (Spencer, Steele, and Quinn 1999).

The risk of the "stereotype threat" has been shown to affect women's, and those belonging to other minority groups', performance (Aronson, Cohen, and McColskey 2009; Aronson, Fried, and Good 2002; Spencer, Steele, and Quinn 1999). While I have no data in relation to grades or test-taking to explore this facet of women students' experiences in Singapore, it was clear that concerns about gender and the role of gender impacted upon some students. A discussion with Meisi, a third-year student from China, shows the effects of discussions about gender and stereotype threat on her perceptions of her own ability and that of other women.

> **Sam**: Do you see programming being related to gender in any way, yours or other people's?
> **Meisi**: I think the school have separated the girls with the boys. Because in the course I just mentioned right now… the tutor says every group of four people there should be a girl student… So I think teachers really think girls are not as strong as skilled as capable as boys in programming. Or maybe girls are not passionate enough as boys.
> **S**: What do you think about that?
> **M**: Yeah, most the students are boys, and the girls are always asking the boys, how to say, so many questions. So maybe the boys are very devoted in learning things themselves and the girls maybe tend to ask seeking help.
> **S**: Does that affect you in any way?
> **M**: Yeah, maybe. So, it makes me a little disappointed in myself, because every time I sit in the programming lab, whenever I encounter a problem and I can't solve it after a long time I ask a boy and he, he's always with the right answer. I think every hardworking boy in that lab seems to know everything so it makes me a little disappointed towards myself… I also discover that really girls are not very, maybe in that course girls happen to be a little bit weaker than the boys. Yeah. Maybe the boys are very crazy about the assignment. But anyways girls and boys all finish the assignment in the end, so yeah. (Meisi 2014)

By highlighting women's distinctiveness and separating them into different groups, Meisi feels that this also reflects the tutor's and teacher's perspectives on women's abilities. Her own practices of asking questions, inevitably to "boys," (re)produces for her the sense that this reflects her own abilities as well. The comparison of men's and women's abilities and skills, built on a foundation of assumptions about heteronormative binary genders, thus significantly shaped how students interpreted and experienced computer science, themselves, and one another. These comparisons also more often and more significantly affected, or were seen as related to, women than men.

More generally, national policy and transnational research on gender/women in computing thus (re)produce gender norms that limit the possible ways of being a man and being a woman in computer science and in Singapore. Structural-symbolic renderings also summon particular interests and practices among women and men in terms of femininity and masculinity, respectively. As Judith Butler suggests, gendering "is the matrix through which all willing first becomes possible, its enabling cultural condition" (Butler 1993, 36:7). Renderings of femininity (re)present and summon women as less skilled and interested in or passionate about technology compared to men. Moreover, while women are a necessary part of the Singaporean workforce including in the government's own policies, their primary role according to government policy is as mothers. Technological passion and entrepreneurial mobility, on the other hand, are rendered as associated with masculinity (and vice versa).

As discussed in Chapter 6, students learn to enact a particular computer science *habitus* in order to "become" computer scientists. This *habitus* is also gendered as students must learn to display and perform technical skill and passion (or not) in relation to both the hegemonic personhood in computer science and masculine and feminine norms. These structural-symbolic genderings are further enacted in students' intra-actions in designing technologies. In one course, students were asked to think about the users of their intended self-selected projects and outline their "personas." Personas are a technique for detailing the characteristics of a particular user (usually invented rather than referring to a specific person) to make them more relatable while creating a design and program that meets their needs (Cooper 2004). There has been some critique of personas and their tendency to lead to stereotyping (Marsden and Haag 2016; Turner and Turner 2011). While some students confused the practice of creating a persona with outlining the different user groups of an intended program or technology, in either case there was a similar tendency among students to draw on and present gendered stereotypes in discussing men's and women's roles and skills.

For example, the personas that one student group developed for a shopping program to help users locate goods in a supermarket included a woman persona representing "housewives" who "have a lot of time on their hands" and "not usually equipped with computers" so have limited technical skills. In contrast, they used a man persona to represent professionals who rarely use the supermarket, are single, and have expert technical skills. All other personas presented by this group were also men, representing students and store staff. Similarly, another student group presented four personas, three men and one woman, for a public transportation program to help someone monitor their progress on a journey using a mobile device. The three men included a retiree tourist, a student, and a senior sales rep who needed to multi-task and help finding unfamiliar places for leisure and work. The one woman persona was presented as a person who lacked a sense of direction, could not remember directions, and could not orient a map, thus would clearly need the help of such an application. Other groups similarly presented personas that relied on gendered stereotypes and biases that presented men personas as skillful, professional, and knowledgeable, whereas women personas lacked technical and other skills and were tied to roles of care and motherhood (e.g. the "housewife"). These groups were formed of a diverse multinational group of students.

These examples demonstrate the ways that gendered intra-actions diffract, co-producing both gender and technology (Cockburn and Ormrod 1993; Wajcman 1991). Following the research by feminist technology scholars who have explored the ways that gendering becomes implicit in designs, students in computer science at Temasek University reproduced a variety of associations and stereotypes relating to men and women, their interests and abilities, and their roles as gendered persons (Berg 1999; Oudshoorn, Rommes, and Stienstra 2004; van Oost 2003). These structural-symbolic genderings are enacted in the design of technologies, which themselves enact those very genderings; they produce a feedback loop reproducing norms about masculinity and femininity, and their relationships with technology.

These perspectives on gender are tied up with national discourses officially promoted by the Singaporean government and circulating as part of Singaporean media and discourse, which promote women's roles primarily as mothers and homemakers. These perspectives additionally rest and rely on a foundation of heteronormativity and transnational approaches to understanding and researching "women in" computing, which invites comparisons between opposing and binary genders: men and women. Moreover, the sense that gender relates to women, that women may be less interested or skilled in doing computer science, along with discourses focused on problems and solutions, leads to a focus on numbers and numerical imbalance, and on finding "solutions" to the "problem" of women (and their scarcity) in computer science.

7.4 Transnational Solutions: An Anti-Politics Machine

Relative to Canada and the US, the number and scope of "solutions" focused on women in Singapore – usually in the form of activities, events, and projects – are quite few. Yet, they are also growing in number. During my fieldwork, I attended and heard about only a couple of events focused on women. The career advisor for the school of computing had organized an event for women students to meet with women graduates and professionals to network and discuss their experiences and perceptions of gender in the workplace after graduation. I was also told of one computer science course that was known for only ever having three women enrolled over the course of a decade. The solution that was currently being used was to offer women who enrolled in the course an immediate five percent bonus on their mark, provided no one in the course objected. In another course that focused on group work, groups were instructed to include at least one woman student, as discussed by Meisi above, along with separating students from the same national or ethnic groups. While this rule may not have been directly related to an issue specifically about women or gender, this is how it was interpreted by Meisi who saw it as a way to help women who have been perceived as less "strong," "skilled," and "passionate." Finally, one instructor promoted the Grace Hopper Conference in class, suggesting all students learn more about Grace Hopper herself, and that women students consider applying for a scholarship to attend the conference.

These are most of the few examples of explicit and purposeful attention to gender and to women that I observed or heard about among students, professors, and administrators in the computer science program at Temasek University. As seen in

Chapter 4, gender in the form of heteronormativity had a much more significant implicit presence in how computer science is taught and conceptualized. Yet, where there was deliberate attention to gender, it generally focused specifically on women, tying in with transnational approaches to diversity and inclusion in the multinational tech industry and transnational STEM disciplines. The Grace Hopper Celebration promoted by one professor, for example, is “the world’s largest gathering of women technologists” and is organized by the Anita Borg Institute and the ACM (ABI n.d.).[78] The Anita Borg institute was cofounded by American computer scientists Anita Borg and Telle Whitney, and the conference or “celebration” draws on the figure of Grace Hopper, an American computer scientist and US Navy Rear Admiral credited with inventing the first compiler. The celebration is held in various cities in the US and is a significant research and networking event for women in both academia and industry. It is also attended and sponsored by many of the US-based multinational tech companies such as Amazon, Google, eBay, and IBM, to name just a few.

These multinational tech companies also organized women in computing networking and recruitment events in Singapore. I attended, for example, the “Tech Femme” event organized by Microsoft during a visit by recruiters from the US, who were also giving technical and recruitment talks at Temasek University. The event was held at the Microsoft offices in Singapore, which focused solely on client relations. The theme for the event was “heroines” with organizers wearing superhero t-shirts and the cupcakes

[78] Chapter 4 also discussed the role and significance of the ACM as an international (but largely American) organization.

for attendees were all decorated with super hero figures. The event also featured several panelists, all women, one from Australia, one from the US, and two working in Singapore but originally from India and Taiwan. The panelists repeated many of the symbolic gender associations discussed above, with one suggesting, in response to a question about barriers facing women in tech, that some women have interests that are not as "techy" as men, but that they can work in different departments at Microsoft. The panelists, however, also recognized there were stereotypes that women are not as technical or are often "quieter," which they need to work against. These responses, while contradictory, are also similar in that they treat gender as inherently binary and limit their talk about it to comparison between men's and women's qualities. They reveal and reproduce the conduct of conduct that occurs when speaking about gender and computing as centring around the qualities women need or lack (in comparison to men) in order to succeed in computer science (Foucault 1991).

Many of the other events and groups centred on women in tech in Singapore were similarly tied to multinational corporations or international organizations. There have been several "Rails Girls" events in Singapore, the first held in 2011, centring on the programming language and framework Ruby and Ruby on Rails. These are part of the international Rails Girls group founded by Linda Liukas and Karri Saarinen, both Finnish computer scientists, who held their first event in Helsinki and later workshops in several European cities as well as Singapore. During my return trip to Singapore in 2016, I also attended a meetup of the Singapore PHP user group that meets to discuss projects and information relating to the programming language PHP. The meetup featured a variety of tech talks but at the end also introduced an organizing initiative to begin a local PHP Women's group. This was presented as inspired by a talk about the global online PHP Women's group that was given by Jenny Wong, a programmer from the UK, at the last PHP conference in Singapore. The organizations that focus on women in tech in Singapore have proliferated since my fieldwork. These include: PyLadies SG, the "Singapore chapter" of an "international group focused on getting more women into the Python community," founded in June 2014; TechLadies, a locally organized and run coding boot camp for women, started in January 2016; the PHP Women's group; and, most recently, Women who Code, an international non-profit organization that supports women in technology related careers, launched in Singapore in January 2017.

The growing number of groups and gatherings suggests they are significant and meaningful for some women and tech communities in Singapore. Yet, the international origins of the majority of these groups also point to the reproduction of this transnational and traditional way of "solving" pipeline and gender issues. The events, groups, and solutions repeat and reproduce local and transnational traditions of focusing on women in terms of their abilities, interests, and perceptions and casting these as the problem to be addressed. These approaches to gender additionally intersect with research that takes women's numerical minority as the key problem of gender.

As I have suggested elsewhere (Breslin 2015), however, research and solutions that centre on the quantitative comparison between men and women often works as an "anti-politics machine," following James Ferguson (1990). Ferguson considered how and why development projects in Lesotho were continually implemented despite also continually failing to achieve their goals. He outlines how political and economic issues are continually reconstructed and presented through "'development' discourse" as problems with technical solutions, similar to Tania Murray Li's process of rendering technical (Ferguson 1990, 27–29; Li 2007). Ferguson, however, additionally argues that although the projects fail according to their own measurements, they do work to increase governmental control in areas of development, hide the inherent politics of the changes that are made, and depoliticize poverty, the workings of which he calls an anti-politics machine (Ferguson 1990, 255–56).

In the case of women in technology, Vitores and Gil- Juárez suggest about the oft-used leaky pipeline metaphor:

> The metaphor, embedded into the discourse of competitiveness, offers a view of women and other (racial) minorities as statistical categories for policy makers who need to recruit and retain a techno-scientific workforce composed of people other than white males, [which] became an *apparatus* of knowledge and policy production (Vitores and Gil-Juárez 2015, 4).

The continuation of this technical framing of projects ensures that women (and men) are continuously counted, measured, and judged, while hiding the ways that such research on women in computer science (along with related diversity and inclusion initiatives) maintain the status-quo in terms of heteronormativity and structural-symbolic gender inequality in computing and tech. The intersection of heteronormativity and practices of rendering technical form the foundation of this anti-politics machine – they "structure the possible field of action of others" (Foucault 1982, 790) – which shapes the frame of discourse, circulating transnationally, for understanding and doing research on gender and computing, and for understanding and doing gender in computer science.

Moreover, the modes of thought entailed in rendering technical shape the possibilities for thinking about what are valid or real problems in computer science and beyond. The framing demands a problem to solve; in other words, a solution must be possible in order to provide an area for intervention. The focus on *women in* computer science then becomes understandable. The problem can be formalized and quantified; if the issue is the numerical lack of women, the solution is to increase the number of women in computer science. Processes can thereby be devised to solve this problem. As such, while many (myself included) may start research with the intention of avoiding questions of women in computing, the possibilities for thinking, communicating, and doing research on gender in computer science are continually redirected to questions of comparing "opposite" genders and thereby of technical problems (with implied solutions).

7.5 Conclusion

Catharina Landström (2007) has argued that much research looking at gender and technology co-production are slanted towards addressing the social construction of technology, leaving gender as an implicitly heteronormative and stable "black-box." Vitores and Gil-Juárez additionally suggest that research on gender and computing needs "new eyes" to escape the way the pipeline metaphor channels thoughts and questions to gender binaries and the construction of gender (Vitores and Gil-Juárez 2015, 9–10). I similarly show how research on "women in" computer science builds on a heteronormative framework and works as a transnational tradition that repeats and reproduces technical renderings of gender based on binary comparisons and numerical accounting and equality. These traditional frameworks of heteronormative gender binaries, the equation of gender with women, and binary comparison, all shape the frame of discourse – governing the "conduct of conduct" – for understanding and researching gender and computing.

This framing renders claims that "gender actually doesn't matter" in Singapore or in computer science comprehensible, where international measures of equality are largely favourable and few students observed or experienced any overt forms of bias or discrimination. At the same time, I show how there are indeed structural-symbolic gender inequalities in both contexts. Renderings of femininity in discourse, practice, and policy (re)present and summon women as less skilled and interested in or passionate about technology compared to men, with their primary role, according to government policy, as mothers. Masculinity, on the other hand, is rendered as associated with technological passion and entrepreneurial mobility, dovetailing with the hegemonic personhood discussed in Chapter 6. These inequalities, however, are both masked and reproduced by discourses and practices that compare men to women, often taking men as the invisible norm – the "unmarked marker" – and thus seek to "solve" the "problem" of women in computer science.

The next chapter explores the paradoxes and challenges that students experience in relation to hegemonic norms and values discussed above and in the preceding chapters, including in-depth the discussions about gender and selfhood by several students in computer science. I consider how their lives, identities, and aspirations do not easily fit within gender categories and metaphors about the computer science pipeline, and the torqueing effects of these categories as these students contend with values, behaviours, and norms relating to gender and computer science, expectations from parents and family, and possibilities for their future careers and lives. These struggles and torques are not easily “solved” by projects that focus on numerical comparison and equality.

Chapter 8 : Reconfiguring (G|R)enderings

> **Samantha**: Now that you're working on this degree, what do you hope to do after? You said you're bonded to the Singapore government, and you don't necessarily want to program, but do you have other ideas?
> **Zhang Wei**: For example, computer science students are admitted to many areas, like bank, consulting, many things, IT support. So, I think my ideal for now, maybe I'll change, for now I think I can go to IT support, something like that with more communication with clients. So for after graduation, I have two [possible] roles. One role is to work, another role is further study. But because of my program, further study in Singapore is ok if you have ability, [but] if you want to go abroad to study, like in America, in Europe, you need to give a deposit to guarantee that I will be back. So, I'm still deciding because it's quite a big amount, so I need to think about it.
> **S**: Yeah, so you would have to come back for more years again?
> **ZW**: No no, just 6 years, but still a deposit.
> […]
> **S**: Do you think [the bond] it's a good idea?
> **ZW**: Yeah, I think it's a good idea. You know, I think it's good benefit for China, for Singapore, and for me as a student. Some students complain about it, [they] think 6 is too long, but I don't think so. First, why it benefits Singapore: because more student all over Asian area they can serve Singapore after graduation. And for me, because first I'm not very satisfied with my university in China. I didn't do well with college entrance exam, and this gives me a good opportunity. You know, here, I expose myself to the multi-culture, and that's wonderful. I think it's more like Western education. The system, and many opportunities as I mentioned, it's wonderful for me. And, for China, I think because I'm the only child. You know we have only-child policy in China… So I think I will take care of my parents when they get older and older. So after I work here, I possibly go back to China, so I also can serve my motherland. So that's mutual benefit. Yeah, I think from my perspective, maybe I'm too positive.
> **S**: I guess it depends if you find a job you like in the end.
> **ZW**: Yes. But, you know what, Samantha, I learned from my experience that sometimes life is not about what you want and then you get. I try to, what I get I want. Like, I'm chosen to learn computer science, so I try to like it. Because [even if] I don't have interest in it, it's ok. (Zhang Wei 2013)

This dissertation has explored students' experiences in becoming computer scientists, in relation to intra-acting practices and discourses that work to shape and produce students in particular ways. The students and professors that I spoke with were dynamic and complex persons with varying interests, aspirations, and selfhoods. Yet, Chapter 6 explored the making of a hegemonic subjectivity in computer science exemplified by the members of the student hacker group. This subjectivity was focused on passion and entrepreneurialism, intersecting with a hegemonic masculinity associated with Singapore as a global city (Cheng, Yeoh, and Zhang 2014; Connell 2005). As students were called upon to compete with one another, they were also summoned to make themselves marketable and employable persons, which worked to render focused – and narrow – selfhoods where every facet of their life was ideally focused on the specific knowledges, interests, skills, values, and practices desirable to employers.

Zhang Wei, like many other students, contended with his implication in multiple and intra-acting discourses, structures, norms, and values that both led to his current place as a computer science student and continue to shape his possible futures and aspirations. He contends with his interests, which do not always align with the discipline's in-depth focus on technological knowledge to the exclusion of other activities. Incidentally, his discussion illustrates that men can also be ambivalent about their interest or passion for computer science, a situation that is not often considered in research on the "woman problem" in computing, discussed in the preceding chapter. Given the opportunity, Zhang Wei would also like to study abroad in the US or Europe, but he contends with a six-year bond, a legal obligation to work in Singapore after graduation, which was the basis of his original opportunity to study in the city-state. He recognizes the multiple interests, values, and obligations – for himself, for Singapore, and for China – that are tied up with his scholarship and bond. In this regard, Zhang Wei also has filial duties he wishes to fulfill as an only child, to both his parents and his "motherland."

This chapter explores the experiences of students like Zhang Wei and their responses to the multiple demands on their lives and selves, which have been examined throughout the preceding chapters. I consider how students' lives and selves are torqued through these multiple material-symbolic discourses and practices (Bowker and Star 1999). At the same time, I consider how the intra-actions of these discourses and practices create spaces where students challenge and reconfigure norms, values, and practices. As Karen Barad suggests, agency "is not an attribute but the ongoing reconfigurings of the world" (Barad 2007, 141). As such, this chapter explores the agencies enacted and produced through students' intra-actions with technologies, policies, and norms.

I first consider the paradoxes produced through students' intra-actions with the government policies and hidden curricula discussed in Chapters 5 and 6 that cultivate students as entrepreneurial subjects. I then consider the intra-actions of these policies, and of performances relating to the hegemonic subjectivity in computer science, with the reproduction of gender norms. In particular, I consider the conflicts produced through the ideal of mobile "talent" in intra-action with demands of family and citizenship, as well as how students contested various gender norms, both drawing on facets of the hegemonic subjectivity and contesting them.

8.1 The Paradoxes of Mobile Talent

As discussed in Chapter 6, the Singaporean government has repeatedly called for Singapore and Singaporeans to become more risk-taking, creative, and entrepreneurial – more like Silicon Valley. These calls, however, have had a paradoxical effect of making many computer science students themselves look to Silicon Valley for their future aspirations and to view Singapore as lacking, by comparison. I was surprised when I started my research to hear from students that computer science was a “marginal field” in Singapore, as one student told me. I was continually asked why I came to Singapore to study computer science education, often with the unspoken implication that Singapore was not an expected place to study computer science. Instead, students told me about how law, medicine, and business were the prestigious and lucrative fields to study and how there were few opportunities find good jobs in computer science in Singapore.

A second-year student, for example, explained to me how computing and engineering positions were often treated “as lower end, as a path to management position. So… Singapore is not the place I want to be” (Guoliang 2014). This student already had aspirations to become a software engineer in Silicon Valley within 5 years’ time, along with an internship lined up for the summer at a well-known tech company in San Francisco. Shirley, a fourth-year student whose discussions I have explored in preceding chapters, similarly commented “one problem about Singapore is all the big companies don’t have tech teams here, so there are not really big IT companies in Singapore, unless for those who sound really boring” (Shirley 2014). Shirley had a bond to work in Singapore, but afterwards she also hoped to go to Silicon Valley like some of her friends.

Even a first-year student who did want to work in Singapore when she graduated explained: “But it’s a really small place. For IT… I think it’s not a lot of jobs. It’s either you go in a specific company and you just code all day or you, yeah that’s basically all you can do. It’s not like Google where they – overseas – where they can let you be free.

Singapore doesn't really have this sort of thing" (Naomi 2014). Most large multinational companies only have sales rather than development offices in the Singapore, although several new offices focused on software development have opened in the past couple of years. For these students, far from the expansive and innovative visions of technological development promoted by the policy reports, Singapore was not seen as a place of opportunity and possibility. The technology industry was seen as limited and job opportunities and places of work as "lower-end" and "boring" where you "code all day."

Instead, for many students, Silicon Valley is where they aspire to work, seeing it as the wellspring and epitome of entrepreneurial culture, providing inspiration and insight into achieving similar potential for themselves through the "no collar" jobs offered there (Ross 2003). A second-year student, who wanted to work in Singapore ultimately, nevertheless explained why he wanted to go to Silicon Valley for a time: "It's the experience of course. Like going over there, that is like the headquarters of everything, for start-ups. The culture there is pretty strong. So just for the experience and see what I can bring back to Singapore … see what makes them so much better in terms of entrepreneurship, in terms of coming up with new ideas" (Xiaowen 2014). In this context, labour is envisioned as mobile, with the same frictionless freedom of movement as code and technologies, as discussed in Chapter 3. In considering the experiences of Indian IT workers in Germany, Sareeta Amrute similarly discusses how these workers argued that, like code, they too should be free to move across places free of restrictions like visa regulations (Amrute 2014, 109). The freedom of movement into Singapore offered to foreign talent seems to support this perspective, and students seek the same freedom to move out of Singapore as these "talents" have to move into the country.

Following Aihwa Ong's research on Chinese professionals, computing students therefore pursue a kind of "flexible citizenship" where they seek "to both circumvent *and* benefit from different nation-state regimes by selecting different sites for investments, work, and family relocation" (Ong 1999, 112). For Ong's professionals, this flexibility worked as a means of gaining security for the future and upward mobility (Ong 1999, 151, 2006, 2007). Researchers have shown how students and other migrants *to* Singapore pursue such "flexible" strategies (Cheng 2014; Collins et al. 2014; Ong 2007; Yeoh and Lin 2013). Many computer science students' aspirations and strategies of flexibility, however, focused on migration *from* Singapore, and primarily in relation to work, since most did not yet have investments, spouses, or children, although I discuss below students' concerns for supporting their parents in future years and other connections or obligations to family.

Shirley Sun has also explored how parents who are concerned with the stresses of raising children in the competitive environment of Singapore are emigrating to countries where there are more social benefits provided by the state (Sun 2012, 139). Alternatively, among those who stay, citizens often choose to raise fewer children than required for population replacement or growth, despite the state's efforts in to promote procreation among its citizens, due to the competitive climate, their economic insecurity as labourers, the limited and non-egalitarian support offered by government policies. These actions by parents and students frustrate the Singaporean government's work towards promoting the local tech ecosystem and at governing the local population, particularly given the government's concern with the below replacement level fertility rate and on developing the "talent" of its population. Thus, while their population is seen as the Singaporean state's primary resource, it is never fully under their control.

The government's calls for citizens and workers to become innovative, entrepreneurial, and creative has thus produced a contradiction. In working as both planner and visionary the government's visions and desires are unstable, because they do not always refer to their origins in Singapore despite their focus on national development. The Singaporean government's masterplans, discussed in Chapter 6, produce visions of Singapore as a globally competitive "Smart Nation" with technological infrastructure embedded in all facets of society, and they summon and entangle students and others within these visions. The government has cultivated persons who work to become their ideal "technopreneurial" subjects; students in the hacker group and others make themselves so as to be the creative and innovative citizens and workers that the government desires. But concomitantly that subject-making has produced desires to be mobile and work and learn in other places, like Silicon Valley.

Students seek to become "pied-a-terre" subjects in other "mega-cities," thereby becoming (elsewhere) the very foreign talents that the government has promoted and privileged within Singapore (Ong 2007). More generally, the entrepreneurial and creative subjectivities that government policies and hidden curricula in computer science and at Temasek University foster are implicitly but inherently mobile. As Ong suggests of foreign talents in Singapore and other "mega-cities": "for the globetrotting professional, the interstitial phase in a particular city is necessary step to the next occupational rung, perhaps back in the West" (Ong 2007, 91). Students seek to benefit from the opportunities envisioned of Silicon Valley as offering freedom in their work environment, experience in relation to the thriving tech ecosystem and production of "creative" ideas, and possibilities for future mobility.

The conflict between citizenship and the cultivation of such "global" workers is a problem and paradox recognized by the government, as they now seek to attract those workers back to Singapore, while also calling anew for Singaporeans to become more and better by valuing engineering and computing professions and supporting or joining the local technological ecosystem (Yong 2016; see also Ong 2007, 87). The government also uses practices such as bonds associated with scholarships to trap entrepreneurial labour and expertise within the city-state, conflicting with their own policies to cultivate entrepreneurial citizens and subjects.

8.1.1 Legal Bonds

There are two kinds of such bonds in Singapore. One is for foreign scholars who study in Singapore funded through scholarships by the Singaporean government. For example, Singaporean and Chinese governments have a series of scholarship schemes known as "SM1" "SM2" and "SM3." Students undergo a selection process and those who pass are offered scholarships by the Singaporean government to study in Singapore. The separate schemes represent different levels of education and support, with SM2 and SM3 focusing on high school students and university freshmen in China, respectively.[79] In all

[79] The SM3 scheme was discontinued in 2011, meaning several students from China that I spoke with were among the last cohorts studying in Singapore via these scholarships.

cases, the scholarships include tuition fees and a stipend for living expenses. In some cases, the scholarships also include airfare for a limited number of trips. Following the completion of their studies, these students are bonded to work in Singapore, or for a Singaporean registered company, for six years for the SM2 and SM3 schemes.

The second type are prestigious government-funded scholarships for Singaporean students, known as PSC (Public Service Commission) Scholarships. These are often offered in association with a particular government agency or area of service, including healthcare, legal service, teaching service, and uniformed service (Police Force or Armed Forces). Singaporean students are funded to study in Singapore or abroad via these scholarships, with a bond to work for the Singaporean civil service for four to six years following the completion of their studies. The length of bond depends on the place and subject of study, with study abroad and advanced degrees entailing longer bonds. The scholarships cover tuition fees, living allowances, and airfare.

These scholarship bonds are a way for the Singaporean government to secure labour, which is otherwise promoted as free and mobile. In the case of PSC scholars, the bond cultivates civil servants to contribute to the administration and success of the nation state. The bonds for foreign scholars are also ways of cultivating labour resources and new citizens. Unsurprisingly, however, students have ambivalent and multivalent intra-actions with the bonds as administrative – and moral – actors (Yang 2014b). Peidong Yang suggests that the majority of Chinese scholars, known as "PRC scholars," stay in Singapore following the completion of their studies as their social networks and resources are situated there (Yang 2014a, 364). Yet, as discussed in Chapter 6 and above, both

Singaporean and foreign scholars on bonds are encouraged throughout their education in Singapore to aspire to and be mobile entrepreneurial subjects. Students' desires, summoned through entrepreneurial discourses, to go to Silicon Valley, become an entrepreneur, and change the world are thus constrained by these bonds.

Students who do not want to fulfill their bonds can repay them, as specified in their contract, or discreetly leave Singapore and "break" the bond. Financially, the fines for repayment are beyond the means of many or entail significant sacrifices on the part of students' families. Zhang Wei, for example, pointed to the high cost of providing monetary assurance of his future return to Singapore should he choose to pursue further studies abroad as a significant consideration in his future plans. For some, opportunities outside of Singapore including the chance of employment or further study in the US or Europe are sufficient motivation for such sacrifice. The tensions among cosmopolitan desires, bond obligations, citizenship and familial obligations are also clear in Meisi's discussion of her bond and future plans:

> **Sam**: Do you want to stay in Singapore after graduation.
> **Meisi**: I think it all depends if I get a job and it have opportunity to send me to foreign country. Maybe not back to China, maybe to Europe or America. I always want to explore more to get into a new country or new environment. But if my job is very stable in Singapore, [then] Singapore. If I don't want to have my own company, I think I will stay in Singapore. But if I plan to have my own career, maybe I want to be a boss and to program a game to get it published, maybe I will go back to China.
> …
> **S**: If you didn't want to stay is it feasible to you to break the – like to pay the penalty, or is that not really an option.
> **M**: Yes, if some family is rich they can afford to pay the $200,000 punishment. But as I said, if I don't take the scholarship I won't have the chance to study abroad, which means my family are not very rich. In my home town, if we sell the apartment in my home town we can afford to pay that punishment. But I don't think it is worth it to sell our home to just escape from Singapore. Singapore's not

> scary or not very [horrible.] I think it's fair. It's modern. It's energetic. Yeah. The most important is it's fair, so I want to stay in Singapore. (Meisi 2014)

For Meisi, the cost of repayment is not feasible for her parents, yet she does want to work in US or Europe if she has that opportunity. At the same time, she sees Singapore as a fair place to live and work. Given her constraints, like Zhang Wei who explained "what I get I want," she shapes her desires according to the obligations of her bond.

In contrast, Vince, a foreign student from a Southeast Asian country who had a three-year bond, explained how his perspective changed throughout his studies:

> The tuition grant, in exchange for the 40% tuition grant, you stay. You have to work in Singapore for 3 years or any Singapore registered company. Of course, when I heard that I was like, 'oh that's really good, it means you get to work in Singapore, Singapore's a pretty awesome place.' Back then I thought it was a pretty awesome place. I still think it is a pretty awesome place. I would be hesitant to live here for the rest of my life, but that's a different story. But now a lot of my friends are, I mean we've gotten good enough to go to the Valley and intern and work at startups and stuff, and big companies, and we all have offers from Silicon Valley companies. So the bond is actually holding us back. We totally didn't think about this or expect this when we were young – freshmen prancing about (Vince 2013).

Vince embodies the entrepreneurial "talent" that the Singaporean government is seeking to cultivate, and his desires and opportunities epitomize the paradox discussed in the section above. Vince's desires and opportunities thus conflict with the obligations of his bond, but he also did not speak about breaking or repaying it.

Yang points out that these bonds are seen to entail a moral obligation as much as a legal one (Yang 2014b, 365–66). "Bond-breakers are spoken of as ingrates and, in worse cases, as 'thieves' who steal Singaporean tax-payers' money that funded their education" (Yang 2014a, 365). While Yang points to the connection of these moralizations to stereotypes of Chinese immigrants in Singapore, Singaporean scholars who break their bonds face similar judgments. In 1998, a PSC scholar who had been studying overseas and broke his bond in order to continue postgraduate degrees in the US was named (and shamed) in Singaporean public media and Parliament. A statement by now Prime Minister Lee Hsien Loong (then Deputy Prime Minister) argued that "when a scholar breaks a bond, it is not just a matter of contracts and liquidated damages. It involves deeper issues of right and wrong, moral integrity, a sense of shame at breaking a solemn personal undertaking" (The Straits Times 1998). Moreover, "when the obligation is refused, shame is brought not only to the scholar, but also to his family, his principals, teachers and everyone else who shared in his success" (The Straits Times 1998). While bond-breakers are branded as morally destitute and face significant financial penalties, those who earn PSC scholarships and fulfill their bonds, in contrast, present themselves as having gained "opportunity, honor, and prestige" from the scholarships (Tan 2008, 18).

Geoffrey C. Bowker and Susan Leigh Star (1999) draw on cases of historical tuberculosis diagnosis and classification and race (re)classification in apartheid South Africa, showing how peoples' lives and identities are torqued through these processes: "in both cases, biographies and categories fall along often conflicting trajectories. Lives are twisted, even torn, in the attempt to force one into the other" (Bowker and Star 1999, 28). The ways students' lives and identities are torqued by the scholarship bonds are clear in the narratives by Zhang Wei, Meisi, and Vince. Their financial situations, legal contracts, and moral obligations shape their desires and aspirations.

In this way, students' experiences with bonds embody the tension between cosmopolitanism as a practice and as an aspiration, where students' transnational entrepreneurial aspirations closely align with cosmopolitan pursuits (Amit and Gardiner Barber 2015; Beck 2004; Werbner 2008). As Vered Amit and Pauline Gardiner Barber discuss in their introduction to a special issue of *Identities* on mobility and cosmopolitanism, "circumstances… may foster the development of a cosmopolitan 'disposition', political project or engagement, but this potential is neither inevitably or easily realised nor secure even when it is attained by *some* people at *some* times" (Amit and Gardiner Barber 2015, 545). In this regard, another student who had more financial resources seemed less inclined to shape her desires to fit within the constructed legal and moral obligations of her bond. For her, six years was too much of a commitment to Singapore, and she suggested that she would repay the bond after two or three years in order to pursue a career in Silicon Valley.

Students' experiences with scholarship bonds also reflect directional frictions. Chapter 4 discussed the ease with which knowledges, persons, and objects travelled from the US to Singapore and elsewhere; they are "mobile from birth" (Takhteyev 2012, 42). In contrast, bonds reflect the relative difficulty of mobility by persons, knowledges, and objects from Singapore. Alongside bonds, students from China discussed the challenges of obtaining visas to work in Europe or the US. The issue of visas also highlights tensions and instability discussed in Chapters 5 and 6 and above in the government's promotion of foreign talent and desires to be a global "mega-city" and its desire to cultivate the national industry and expertise (Ong 2007). With $3000 to $3,300 per month as the minimum qualifying salary for new Employment Pass applications, students with bonds are further constrained by the employment requirements of their bonds and requirements of visa passes. A university administrator told me that this produced much anxiety and concern for students.

One student told me that they could be released from their bond if they proved to the government that they been dedicated but unsuccessful in their search for a job. While such a release could offer the opportunity to search for jobs elsewhere, it seemed students in this position were more likely to return to their home countries, at least at first. As such, these students' struggles additionally exemplify the insecurity of cosmopolitan pursuits. Yet, they also point to a different kind of bond in tension with the promotion and pursuit of entrepreneurial and cosmopolitan personhoods – students' bonds to their families and home countries or places.

8.1.2 Family Bonds

While many students wanted to work at large renowned companies in the US and elsewhere, many also discussed how they were tied to Singapore because of their family and connections to place. As Singaporean student, Susan, explained when I asked if she wanted to work in Singapore, "I'm still keeping an open mind, yeah, but I'm most probably gonna work in Singapore where my family and friends are. Unless there's a really good prospect for me to venture overseas, then I might consider it as an option" (Susan 2014). While Susan prioritizes her connection to people in Singapore, she also leaves open the possibility of becoming mobile and working abroad for a "really good prospect." Ariff, another Singaporean student, similarly pointed to the intersection of family and place in shaping his decision to study in Singapore:

> **Sam**: Did you definitely want to stay in Singapore or did you think of going overseas?
> **Ariff**: My parents would prefer me to stay in Singapore. But for me, I would prefer to stay in Singapore also. It's nothing personal lah but yeah, I just really prefer Singapore lah. It's easy here, easier than going overseas. There's more halal food here also lah. Oh and what else. Oh and all my family and friends are also here so. It would be harder to adapt overseas so I just stay here. It was a good choice also.
> **S**: You're happy with it.
> **A**: Maybe, at most, I just go on overseas exchange program, at least find out how it is to study overseas for like three months. (Ariff 2014)

Ariff points to his interest in some overseas travel through an exchange program, indicating his engagement with discourses of cosmopolitanism and mobility, discussed above. Yet, his desires are limited to this brief kind of exchange due to his attachment to friends and family in Singapore, and intersecting concerns such as access to Halal food facilitated in particular places.

The Singaporean government has promoted duty to family as part of the "Shared Values" discussed in Chapter 4, and the value "family as a basic unit of society" (The President of the Republic of Singapore 1991, 10). The first shared value additionally promotes "Nation before community and society above self" (The President of the Republic of Singapore 1991, 10). In this regard, while the government promotes and cultivates students as mobile and entrepreneurial, students' selves – including their labour – are first meant to serve the nation "above self." Students thus contend with the conflicting intra-actions produced through government's efforts to bond citizens and subjects through scholarship bonds and commitments to family and place, and the promotion of entrepreneurialism.

A summary of the Shared Values from the National Library Board, for example, explains how these values, in part, "reminds Singaporeans not to shun the responsibility of caring for the aged" (Lim n.d.). The desire and duty to care for their parents was expressed by multiple students of different nationalities, as seen in Zhang Wei's discussion above. For students from Singapore, these duties to their parents influenced their desire to stay in Singapore, or perhaps return after a stint abroad, as seen in Ariff's discussion. For students from China, India, and other countries, obligations to parents often meant they planned to eventually return to their home countries where their parents lived, or to create the possibility for their parents to move to Singapore. Zhang Wei, for example, further explained his plans for the future, including in relation to his parents:

> **Sam**: Say 10 years from now, do you have a specific vision of what you want your life to be?
> **Zhang Wei**: I will work for Singapore, no matter if I study further abroad or not, because I appreciate the scholarship. As a return I have to work.… I want to be in at least a middle class… middle-level in a company, so that [it's] more easy for me to work for the company in the China branch, which I mentioned I want to take care of my parents… And I think I should have an apartment... so that I can live. Mm, maybe my parents also can come here with me. (Zhang Wei 2013)

Zhang Wei's plans for his career centre around his plans for caring for his parents, and his legal bond and sense of moral duty to the Singaporean government for his scholarship. While both women and men discussed caring for their elder parents, this responsibility was more often discussed by men.

On the other hand, women more often discussed their responsibility to their families in terms of marriage and children, although many women also discussed their plans not to marry. Shu Wen, a student from Singapore, also discussed her desire for children, implicitly also suggesting she prefers sons over daughters:

> **Sam**: Do you hope to get married?
> **Shu Wen**: I haven't had a boyfriend before… But of course I will need to get married, cause I want my parents to see their grandsons and so yeah. (Shuwen 2014)

I also discuss below how Meisi contends with her parents' expectation for her to marry. These family obligations and desires also entail place-based bonds, to China for Zhang Wei with a possibility of moving his family to Singapore, and to Singapore for Shuwen. These bonds produce a conflict between students' aspirations for successful and mobile careers, and their desires or obligations to marry and/or have children. This tension is also at the heart of conflicting government policies, discussed in Chapter 7, which sought to promote women's inclusion in the workforce, while also arguing that motherhood is women's primary role and responsibility.

This conflict is seen in some students' discussions around prioritizing their careers or their families. Third-year Singaporean student Aindri, explained:

> **Sam**: Do you have any aspirations to have a family, at some point?
> **Aindri**: At some point, definitely. But not now. I think it's really important that I establish my career in a way before think about or even start having a family. That is definitely something that I would want. Right, [because] once I do have a family, for example, I would want to put my focus in my family, because I already know that my career has been established. (Aindri 2013)

In this regard, students like Aindri anticipate similar conflicts of identity discussed by Heather Paxson in relation to modern motherhood in Greece:

> Being a woman and being a mother have come to symbolize the paradoxically opposed poles of female adulthood. As modern *adults*, women are supposed to act independently, exercise reasoned decision making, and assume responsibility for their actions. But as adult *women* (which includes mothers or potential mothers),

> they are to subordinate their own interests to others', be swayed by emotion, and act cunningly (Paxson 2004, 10).

Some of the traits of Greek women, such as cunning, differ in relation to norms of Singaporean womanhood. Nonetheless, women studying computer science in Singapore experience paradoxes and conflicts in terms their subjectivation towards becoming independent learners, who are also ideally mobile and passionate entrepreneurial subjects, and discourses calling upon them to serve both their family and the nation as mothers.

With the heterosexual family treated as the foundation of the Singaporean nation, Heng and Devan point to the essential role of patriarchal authority, particularly in relation to Lee Kwan Yew's efforts to promote women graduates to marry, where "reproducing power and the power to reproduce" are intimately intertwined (Heng and Devan 1995, 201). While Yeoh et al. (2000) point to the masculinist assumptions and ideals embedded in the construction of global cities, such as Singapore, the Singaporean government's promotion of mobile entrepreneurial talent is thus crosscut by its policies on family and gender roles. For the Singaporean nation-state to reproduce itself, Singaporean women must both remain in Singapore and have children (within the confines of the heterosexual family). At the same time, already mobile "foreign talents," such as Chinese students studying in Singapore, contend with similar obligations to their home countries and families elsewhere.

Even students who were members of the student hacker group – who come to exemplify the hegemonic computer science personhood – suggested potential future conflicting intra-actions between their career aspirations and their familial relationships. Vince, who is a member of the student hacker group and who discussed above how he had job offers in Silicon Valley, as well as how he planned to start his own company, nonetheless pointed to the ways his relationship with a future spouse could “cut” the intra-actions that have produced his mobile and entrepreneurial desires (Strathern 1996). When I asked how he envisioned his life in 10 years, he explained:

> I really, I don’t know. I could be in any city. I’ve already made a decision to stay in Asia for a bit, like 5 years, to try to do a startup here starting from like first principles of what is logical, what advantages are available to a startup here. Then, like, getting a girl-friend, getting a fiancé, [getting] married and stuff might completely change my plans. I really don’t know. I have issues with raising a family in Singapore, I don’t like the environment, I don’t think. High pressure academics, being judged from an early age, streamed into different streams according to ability. I’m not sure I want that for my kids, if I have kids. That’s another thing, see there’s so many unknowns, whether I have kids or not depends on my partner and what she wants as well. And like, it’s not my decision alone. So I really don’t know (Vince 2013).

Vince discusses his conditional future with numerous uncertainties, but it is clear that he sees himself as mobile and entrepreneurial with the possibility of living “in any city” and plans for a startup. At the same time, he expresses concern about the values of independent and competitive learning and networking, which he nonetheless has learned to embody throughout his studies. He also comments that his future plans depend on his future partner’s plans and aspirations.

In this regard, the conflicting intra-actions and futures with which Vince contends further illustrates how embodying and becoming the hegemonic personhood is never fully and finally achievable. As Butler suggests of gender, the ideal is “‘a compelling illusion, an object of *belief*’” (Butler 1988, 521). Freeman similarly suggests that entrepreneurial selves are always a “work in *formation*” (Freeman 2014, 2). Students are summoned to work towards this ideal, either through their passionate dedication to the values of entrepreneurialism and technological interest, or through the coercive demands of a tech industry that seeks to hire the "techiest of the techies," as discussed in Chapter 6.

Paul Willis's study of lads showed how their self-damnation was achieved through the intersection of class culture with gender norms, where "the brutality of the working situation is re-interpreted into a heroic exercise of manly confrontation with *the* task" (Willis 1981, 150). Chapter 7 briefly considered the ways that masculinity intersects with the norms and values of what it means to become a "good" computer scientist, including passion and logical thinking. Moreover, a form of hegemonic masculinity in Singapore is associated with being educated and financially successful and mobile professionals (Cheng, Yeoh, and Zhang 2014, 5; Connell 2005). As such, the intra-actions of gender with computing and class norms can reproduce students' passionate dedication to entrepreneurial and precarious (over)work.

Yet, as seen in the discussions above, the intra-actions of this hegemonic subjectivity with gender, family, and citizenship norms and values also complicates this neoliberal and patriarchal "reproduction machine" (Kondo 1990, 222). As Barad, following Butler (1993), explains: "the juncture of contradictory discursive demands on the subject prevents the subject from following them in strict obedience" (Barad 2007, 213–14). Computer science students, including those in the student hacker group, are enmeshed in contradictory demands for their selves, aspirations, and futures. These conflicting intra-actions, and students' challenges and reconfigurations enacted through them, are further seen in relation to students' enactments of individual gender (Harding 1998).

8.2 Gendered Persons

I consider here the experiences, identities, and performances of students who contest or reconfigure gender norms and the hegemonic personhood discussed in Chapters 6 and 7. While many of these students stated that there were "not much gender issues," as suggested by Naomi in the preceding chapter, they nonetheless discussed complex and multiple facets of their identities that related to gender and where they experienced frictions or torqueing in their lives. However, these discussions were not obviously about bias or inequality between sexes/genders, and as such they do not easily fit within the dominant framing of what constitutes a gender "issue."

Students contested their need or desire to become "hard-core" computer scientists and, in that regard, discussed their varied goals and aspirations for the future. Students also expressed ambiguity or fluidity in relation to their gender identities in different circumstances, and in general, complicated normative gender binaries and oppositional stereotypes. In some cases, these students explicitly recognized the hegemonic subjectivity discussed in Chapter 6 and stated that it was not who or what they wanted to be. As such, these students point towards alternative ways of becoming and being a computer scientist.

8.2.1 Contesting the Hegemonic Personhood

Many students in computer science cannot or, alternatively, refuse to meet summons to make themselves measurable and comparable. As discussed in Chapters 5 and 6, academic practices relating to grades and the bell-curve work to render students in ways that can be assessed and compared. Students additionally learn to judge themselves and others based on their performance of various qualities, most notably the performance of passion. Qiaohui seemed to be thriving according to these measurements. She took multiple modules that were widely considered to be "hard," she worked as a TA for a first-year class, and had spoken earlier in our interview about how she was proud to be studying computer science and succeeding. In this regard, it seemed to me at first that Qiaohui was successfully coming to embody and perform the kind of subjectivity that was hegemonic in the program and the discipline; she was learning to become a "good" computer scientist. She also stated that "no it doesn't bother me" that there are fewer women in computer science than men.

I was then surprised at the end of the interview when Qiaohui explicitly disclaimed this identity and performance:

> **Sam**: Do you have anything you would like tell me that you think I should know?
> **Qiaohui**: Another point I want you to know, even if I'm a TA, even if I did well in this module previously last year, it doesn't mean I'm a hard core computer scientist. I'm not. I thought I would be, I used to think I was. I finally found out I wasn't, I'm not.
> **S**: Why not?
> **Q**: Because I find, there are way more fascinating stuff outside of computer science... it's a good thing for me, but it doesn't mean I need to devote as much time as I used to… If you want to be a hard-core computer student, you have devote long time. And so, I am not, I don't want to devote that time. I will not. (Qiaohui 2013)

As discussed in Chapter 6, passion is often taken to mean singularly devoted interest, to the exclusion of other activities and the over-dedication of time and energy to computing projects. When I went back to this interview to transcribe and analyze, I realized Qiaohui had discussed with detail and enthusiasm earlier in the interview how she had many interests other than computing, but she struggled to find the time to pursue other academic and extra-curricular subjects and activities, while also expressing consternation at demands of computer science and the predominance of men in the discipline.

Qiaohui also discussed how she wanted to travel and seemed to want freedom from demands of a profession and other expectations for a single-minded discipline. When I asked Qiaohui how she would envision her life in 10 years if she could do whatever she wanted, she commented:

> **Qiaohui**: I don't even want to be a professional woman. I don't know. Just disappear in some random corner in this world.
> **Sam**: Do you want to go travelling?
> **Q**: Oh, yeah, definitely, yeah I like travelling. Yeah, just doing - I like random adventures, something out of the blue.
> **S**: Do you want to have a family, is that something?
> **Q**: No, no. I don't. I mean, I find it very hard for me to really commit to something, really settle down. Because after a long period, I'm always thinking about, no matter how glamorous my current situation is, I won't stick to it for a long time. It's so hard for me. I would feel like restless about things. (Qiaohui 2013)

Qiaohui's desires for travel fit easily with the cultivation of mobile and cosmopolitan subjects by both the Singaporean government and the transnational discipline of computer science and tech industry. Qiaohui is also originally from China and studying in Singapore on a scholarship with a bond and as such is part of government projects discussed above to attract and develop foreign talents in Singapore. Yet, her desire to "disappear in some random corner in this world" and not be a "professional woman" indicates that her longing to travel are not only, or even primarily, about becoming a "nomadic" professional talent contributing to Singapore or other global cities (Ong 2007, 83). Rather, she seeks to distance herself from the pressures of computer science and professions in general and just "disappear" even if she is in a seemingly "glamorous" situation.

Qiaohui's discussions reveal conflicts relating to the governance of students to embody particular hegemonic subjectivities. These conflicts emerge, at least in part, through their intersection with norms of gender and femininity. Qiaohui's discussion early in her interview points to this conflict:

> **Sam**: So, do you think, being female, do you think that has shaped your studies in any way, or shaped your experiences?
> **Qiaohui**: I think it challenges my, [pause] I think sometimes it makes me think about you know my identity sometimes. Because you know as a female because computer scientist it's not a very typical profession for a girl, a female. So sometimes I was really thinking about it, really, am I seriously going to be a computer scientist or do anything related, in this computing field when I graduate, or just for the next 10 years of my life. Am I seriously going to do that? Sometimes, I struggle, I'm still not very clear about this question… But on the other hand I'm really proud. Proud of the fact that I'm studying computer science… I'm really glad that I get this training and I think it's quite a privileged training. (Qiaohui 2013)

As mentioned above, Qiaohui is proud of her accomplishments as a computer science student. During a final group project presentation by Qiaohui and others, the professor observed that they were all "strong programmers," confirming her evident success in learning and displaying computer science knowledge thus far. Yet, women's numerical minority and status, in part, prompts Qiaohui to question her belonging in the discipline. More than that, Qiaohui's comments point towards disjunctures between the identities she wants to cultivate for herself and the aspirations she holds for the future, on the one hand, and the sometimes contradictory possibilities for being a woman and a "good" computer scientist, on the other.

Qiaohui also does not seek to embody the norms of femininity, particularly relating to motherhood, in Singapore or in China, the latter of which will be discussed further below by Meisi. In particular, she says she does not want a family or long to "settle down." She also questions her desire to become a "professional woman." She thus resists both roles (mother and citizen-worker) that are promoted nationally for women in Singapore, where she is bonded to work for six years following graduation. Thus, while Qiaohui's discussions contest the norms of personhood and gender in computer science in ways that are largely framed by heteronormativity, she also seeks an alternative identity and life for herself that is free (or, at least, not so bound) materially and symbolically by those norms.

Susan, whom I discuss further below in relation to her sense of self as fitting well in computing, partially because she sees herself as "boyish," similarly contests the demands of singular dedication entailed in the hegemonic personhood. When I asked Susan whether she ever thought about joining the student hacker group, she said no both because she thought it would require "advanced programming skill" and because programming was not something she wanted to do in her free time. She suggested that people "should spend more time instead on the things that you don't have, you usually don't find time to do… like discover new perspectives and stuff like that... like your hobby. If you want to learn all this programming stuff, just go take up course yeah" (Susan 2014). Unlike the members of the student hacker group, Susan clearly cordons off programming as a part of her life associated with studies or work, and separate from her personal or "hobby" time. Susan does not contest the values of passion, as she repeated multiple times that she was following her interests, explaining that despite her struggles learning computer science, "I'm really very sure that my passion still lies with mathematics and somehow programming is like, kind of related to mathematics" (Susan 2014). Yet, she contests that a singular devotion to computer science as a part of what is necessary to demonstrate passion. She also implicitly contests the association between mobility and passion, through her plans to stay and work in Singapore because of friends and family.

As discussed in Chapter 6, in his ethnography of the lads, Willis explores their process of "differentiation" as a form of subject-formation through opposition to institutional norms (Willis 1981, 62). From the teachers' points of view, the lads become unruly and disobedient, yet from their own experiences and changes in bodily style and dress, Willis shows how the lads are able to embody an opposition and freedom from the structures of school. For Willis, this process of differentiation was part of adopting a "counter-school culture" that was also closely entwined with local working-class culture. Qiaohui's reflections on her new-found resistance to and differentiation from the idea of becoming a "hard core computer scientist," however, does not entail a rejection of schooling altogether, only of the singular demands of computer science and its particular gendered figurations. In this regard, Qiaohui might become one of the many women who are counted as part of the "leaky" or "shrinking" pipeline (Camp 1997).

Yet, such a measurement also fails to capture the multiple facets of governance that Qiaohui contends with (relating to gender, computer science norms and values, and national labour expectations). Moreover, it is through Qiaohui's conflict with particular norms of femininity, namely marriage and motherhood, that she further enacts this differentiation. Conversely, Susan distances herself from the norms of femininity, implicitly drawing on the association between passion and masculinity, even as she contests the need for that passion to be performed through singular dedication and mobile entrepreneurialism. Qiaohui and Susan are thus not enacting a singular class-based process of differentiation. Rather, Qiaohui is working to twist or branch out and away from the torqueing produced by multiple forms of governance (Bowker and Star 1999), while Susan is reconfiguring the intra-active meanings of gender, citizenship, and what it means to become a "good" computer scientist.

8.2.2 Complicating Gender Binaries

The effects of this process of torqueing are also seen significantly among students whose lives and selves complicate gender binaries. Take Meisi, a second year student also from China. She always wears skirts and blouses and wears her hair long, which she said she does purposefully to appear feminine. Her choice of dress differs from the majority of students in computer science who often wear shorts and sporting-style t-shirts. As one post on the university social media confession page commented:

> *wears a dress*
> Other faculties: "Nice dress!"
> Computing: "Later going out ah?"/"Got event later?"/"Going dating ah?"

The post suggests that feminine dress is unusual in computer science, reinforcing the "otherness" of femininity in relation to the discipline. Meisi stated explicitly, however, that when thinking about dressing and self-presentation, she thinks of herself "as a girl" in this context and worked to perform as such. She also was clear that she identified as heterosexual in that when “I look at boys,” she said, “I think of myself as a girl.” When I asked about her gender identity, however, Meisi explained that:

> Actually, I don’t always think of myself as a male or a female. I don’t, when I think about something, I don’t keep reminding me of I’m a girl so I have to think in this way. If I choose clothes I will think of myself as a girl, but in programming or other things I don’t give myself a gender (Meisi 2014).

Meisi is in Singapore on a scholarship, which provided her the opportunity to study abroad even though she also has a 6-year bond after she finishes. Among her options, computer science offered the best chance to incorporate her love of music with her studies.

Meisi also discussed her mother’s expectations for her, which point to the norms of femininity she felt she was pressured to embody, at least in certain contexts. “Something my mother always says, you seem too happy. She wants me to do things more quietly. So, speak less or not smile all so often, maybe don’t appear to be so happy, that’s the way to be a lady,” Meisi explained (Meisi 2014). In addition, while, like Qiaohui, Meisi was not directly affected by the Singaporean government’s policy expectations, she was nonetheless being called upon to prioritize motherhood as part of her gender role. Meisi also explained how her mother “says being a programmer is not all the things I have to do in the future, after all I have to get married so have a happy family, have a good husband is even more important than being a good programmer” (Meisi 2014). In this regard, Meisi’s also explained that her mother constantly told her that computer science was not an appropriate subject for girls.

In some ways, Meisi’s statement that she does not give herself a gender when programming partakes in the idea that programming and femininity are at odds with one another, but she also resists the idea of gender as totalizing. Researchers have explored similar contradictions that women experience between "masculine" fields of computer science, as well as engineering, and their identities as women (Bury 2011; Faulkner 2007; Powell, Bagilhole, and Dainty 2009; Stonyer 2002). As such, Meisi's statement reproduces binary distinctions and associations even as she resists and complicates them. These masculine associations were certainly a part of some women's discussions about their sense of fit or belonging in computer science. As mentioned above, first-year Singaporean student Susan, for example talked about her sense of being "boyish" and how that related to her study of programming. When I asked if she experienced any barriers or issues relating to gender in computer science, she explained:

> **Susan**: I think it's basically about a stereotype and this perception of females towards programming. So it's like before I come to University, I also have a stereotype towards programming as well, like it's guy stuff. Like only those nerds will do it and stuff like that. Or, yeah, I mean on TV you only see like guys doing programming in front of their computer and all those are really smart guys, like typing code and stuff like that and they are really cool. Yeah, so it's like everyone have this perception that only guys can do programming. For me, I don't believe in this because I'm also a bit boyish and I like cool stuff, and I really don't mind doing guy stuff or whatever, but yeah, I just really want to do what I like, what I have an interest in. So, basically, no barrier.
> **Sam**: What would you call guy stuff that you do?
> **Su**: Uh, guy stuff like maybe hanging out with guys, like doing guy stuff like doing programming, playing arcade games, you know like computer games… And do stuff like really boyish, wear t-shirt and pant all the time, not wearing skirt, not wearing dress and all that, act really boyish. No need to act in a really girlish manner or maybe like wearing makeup and all that. So, yeah, I find myself a bit boyish as compared to other girls because I don't put makeup, I don't wear skirt, I don't wear a dress, except for a special occasion, and I'm a bit rough also, compared to girls lah. Because usually girls are like gentle and stuff like that, but I'm not. (Susan 2014)

Susan clearly and explicitly associates programming with "guy stuff" and identifies with programming in that context.

Meisi, however, was explicit in her discussion of gender, explaining that "every time I buy a meal, I go to lecture, I do homework, I program, I ask question online, search, I think anything to do with the programming is maybe not related to whether I'm a girl or a boy" (Meisi 2014). As such, Meisi suggests a more complicated and contextually dependent performance and identification with femininity and gender. Note that she suggests in a general sense that programming is not related to being a boy either. Yet, as discussed above, Meisi also points out that "I think the school have separated the girls with [from] the boys" pointing to the course where groups were forced to have at least one girl for their projects. In this context, Meisi experienced the stereotype threat. As such, she said that she sometimes also felt that boys were better at programming, or at least that they displayed a great deal of passion for computer science, which she felt she lacked. Meisi thus contends with gender norms expressed by her mother, as well as implied in her interactions with other students, TAs, and professors. She is trying to make her life and identity in a context where there are constraints and norms about the role of women in computer science and about the nature of sex and gender.

Bowker and Star discuss how classification systems become naturalized as they are intertwined with infrastructures, hiding the political and social projects and struggles that underlay their creation (G. C. Bowker and Star 1999, 196). Binary gender classifications and gender norms are deeply embedded in infrastructures in Singapore and elsewhere, from distinct bathrooms to rules about housing allocation by the Housing Development Board, as discussed in Chapter 4. The torqueing produced in Meisi's life by gender classifications and expectations are clear in her discussions of her future goals.

When I asked how she envisioned her life in 5 years, Meisi offered two visions. The first was to get a job at the bank, because there she can "start to learn new things," but also suggested that "maybe because it's an idea my parents gave me… they always say if you can get in a bank it's always very good and after all bank job is stable." The second vision was to become a game developer or to design special effects. Here too, she preferred a large company because "it's about how my parents and the friends of my parents will judge me. If I get to a small company they will say look at her how she perform in high school middle school, she's always the top and now she goes to a small company." In fifteen years' time, however, Meisi suggested that if it was financially viable she would be interested in teaching computing, which would then give her enough time to pursue her music.

Meisi had many plans – or "possible selves" – contingent on various future paths and eventualities, as she tries to find ways to pursue her interests in music, while also meeting the desires of her parents and her own wishes for a financially stable future (Hardgrove, Rootham, and McDowell 2015; Markus and Nurius 1986). Henrik Vigh uses the concept of navigation as "motion within motion" to elucidate the ways that people work to shape their lives amidst unsettled and moving circumstances and structures: "we move in social environments of actors and actants, individuals and institutions, that engage and move us as we move along" (Vigh 2009, 420). In other words, movement or agency is premised upon and occurs through intra-action. Meisi seeks to navigate her possible selves and futures, recognizing and navigating the multiple "capacities, compulsions, and consequences" of her various aspirations (Gilbertson 2017, 21). There is flexibility entailed in this sense of navigation and movement (Vigh 2009, 424). As such, Meisi's possible selves and futures could be seen as an enactment of flexible citizenship, and of neoliberal flexibility more generally, which are aligned with values of mobility and entrepreneurialism summoned through the hegemonic subjectivity (Ong 1999).

Yet, Meisi's varied plans also indicate a sense of the compulsions that she faces, including her own and her family's reputation based on the type of company she works for, and the desire or need for stable employment. In this regard, to Meisi, her gender identity (or at least its relevance), like her aspirations for the future, varied based on context. She also had several different plans relating to marriage, explaining to me the significance of marriage in China and commenting that "I think getting married is also the big issue for girls. Maybe I don't want to get married, but my parents want [me] to get married, like I'm getting married for them." She explained further her perspective suggesting she could get married "if I can discover such a person in my life" but "if I can't so that's ok" (Meisi 2014). Meisi clearly feels caught in many ways – by the contending expectations of her parents and by disciplinary norms and gender norms associated with computer science – in relation to marriage, her career, and her performance of femininity.

Meisi's suggestion that gender does not matter in relation to programming could point to a freedom of possibilities she sees in programming, enabled and produced through the computing worlds that she is able to create. Part way through the interview, Meisi excitedly showed me the game she was creating for one of her courses, and similarly did so again at a project showcase later in the year. Like Qiaohui, Meisi pursued a variety of "hard" courses in computer science, while also "overloading" her courses and taking six in a term instead of the usual five. Foucault's concept of subjectivation suggests that "the capacity for action is enabled and created by specific relations of subordination" (Mahmood 2005, 29). In this regard, Meisi shapes her desires based on the varied capacities and compulsions with which she contends. She may wish to pursue a career in music, but she is currently navigating her studies in computer science and the possibilities for the future that might (but might not) emerge from it. Meisi's present and possible selves are thus themselves a way of navigating; they are motions within (potential) motions.

The complexities of Meisi's personhood and aspirations defy the renderings and categorizations that measure "women in computer science." Her contextually dependent approach to gender also complicates gender binaries. Singaporean student Naomi offered a similar complication to heteronormativity in computer science and in Singapore:

> **Naomi**: For me I have this warped sense of gender. Basically I don't really genderize myself or other people. So like, might be because I keep reading books with male characters, but yeah, I don't really see anyone in likes [terms] of gender and stuff like that.
> **Sam**: So you think that's not a normal thing?
> **N**: I think most people don't do that right, they don't see themselves as I'm a girl, I'm a boy. They just go like, I'm me and I'm doing this, and I can't do it. They don't blame gender for anything. So like, I'm just bad at math full stop, it's more personal. (Naomi 2014)

Naomi contests the tendency to view herself and others in terms of binary genders, as well as the tendency to stereotype skills based on gender. In terms of performance, Naomi presented herself as feminine in some ways, such as wearing her hair long, which was much more common among women, but she also wore sporting attire such as sneakers, shorts, and a t-shirt during our interview, which were common among men and women students. Naomi had also clearly been positioned as a girl/woman by her teachers and peers in the past. For example, she continued her discussion from above explaining how she was told she could follow her interests rather than work to make money, which was implicitly more of a concern for men because they were responsible for supporting a family. Naomi also mentioned that she had no plans to get married, so did not need to concern herself with supporting a family.

Naomi's assertion that "I'm me and I'm doing this" points to an individualized sense of self, contesting the distinctiveness and mutual exclusion of gender categories, as well as emphasizing the diversity that can occur within particular categories. While saying mathematical skill and gender identity are "more personal" rejects or ignores the multiple intra-actions that shape how mathematical skill is assessed and constructed in gendered ways, Naomi's statement nevertheless asserts her own agency and distinctiveness as a person who is not (or does not want to be) automatically "genderized." In part, this may represent a kind of "lateral agency" where, following Lauren Berlant, "we need to think about agency and personhood not only in normative terms but also as actively exercised within spaces of ordinariness that does not always or even usually follow the literalizing logic of effectuality, bourgeois dramatics, and lifelong accumulation or fashioning" (Berlant 2007, 758). Ordinary decisions and intra-actions such as wearing a dress, not putting on makeup, or even working on and completing weekly math or programming assignments (whether or not one is good at math) entail agency in their repetitiveness and regularity, interrupting the progressive logic of self-cultivation tied to the hegemonic personhood (Berlant 2007, 779–80).

At the same time, Naomi's assertions, along with the navigation of present and possible selves and futures by Meisi, Qiaohui, and others also work as a way of "talking back" to the categories of gender and claiming their multi-faceted and embodied selves as "me" (Hooks 1986). Students' discussions of their selves and possible futures weave in and out of contending with gender norms, while also trying to create their own identities that work around heteronormative structures and normative ideas of femininity and masculinity, even as they contend with the disciplinary norms, values, and practices and work to "become" computer scientists.

8.3 Conclusion

This chapter has explored the paradoxes and conflicts produced through the intra-actions of gender, government policy, citizenship, and students' experiences and selfhoods. Government promotion of entrepreneurialism has cultivated students to become mobile entrepreneurial subjects in places like Silicon Valley, frustrating the government's plans for national economic and technological development. The government thus seeks to trap labour in Singapore explicitly and implicitly through scholarship bonds and the promotion of "shared values" relating to family and citizenship. These various bonds produce conflicts for students, including those in the student hacker group, as they intra-act with aspirations – and summons – to become the hegemonic subjectivity, both in terms of singular dedication and individual mobile entrepreneurialism.

These bonds, along with gender norms, torque student lives and identities, twisting them to fit with classificatory performances of heteronormative femininity, as well as the demands of labour and citizenship. Nonetheless, students such as Meisi perform complex, dynamic, and contextually dependent gender identities. There is no simple solution to the gender structures and labour systems that constrain Meisi's identities and future possibilities based on her bond in Singapore, her mother's wishes for her career, and her own aspirations as an artist and computer scientist. In its foundational assumptions of binary genders and particular forms of femininity operating in "otherness" to masculinity, the anti-politics machine of research on "the woman problem" in computer science, discussed in Chapter 7, hides and reproduces these multiple forms of governance. Yet, Qiaohui's branching away from classificatory torques, Susan's rejection of the singularity of passion, Meisi's moving and contextually dependent aspirations and performances of gender, and Naomi's rejection of a gender identity complicate, contest, and reconfigure these technical renderings through their performative enactments of different ways of becoming a computer scientist.

Chapter 9 : Conclusion

When I returned to Singapore in 2016, many of the first- and second-year students whom I had met during my fieldwork were nearing the completion of their studies. These students had now chosen specializations in computer science and were looking for jobs and careers or had applied to graduate school. They had taken on some of the confidence that I saw in Shirley and other senior students during my initial fieldwork, having much greater clarity of purpose and less ambivalence about their studies. Students who had seemed unsure about their place in computer science several years prior explained their subfields to me, be it algorithms research, human-computer interaction, or graphics and games, and spoke, often enthusiastically, about their plans and futures in those fields.

That did not make everything easy, as students were still contending with multiple challenges. Jobs were not always forthcoming; startup companies demanded labour from applicants, such as tailored coding projects, that students were unwilling to provide and felt were unreasonable; despite its celebration, entrepreneurship was not a sufficient form of labour to meet visa and bond requirements; scholarship bonds were burdensome either in the requirements to stay in Singapore or in the financial penalties for leaving; and family connections and obligations shaped students' time and desires, as did students' goals for relationships, marriage, and children. In spite of, or because of, these challenges, however, it seemed that most students I spoke with had found a place to belong in computer science; they had in some sense "become" computer scientists.

This "becoming" represents progression for these students towards embodying and performing a computer science *habitus* that is aligned with the hegemonic personhood discussed in Chapter 6. They performed interest and passion, technical knowledge, and purpose in relation to their selves and careers, not as hackers, but as particular computing experts and professionals. As such, the performative qualities of the hegemonic personhood were in some sense dynamic, able to accommodate a variety of forms of expression as students progressed through their studies and transition to work, while at the same time summoning a sameness in students' intra-actions with computer science knowledge, projects of self-cultivation, and affective gendered performances. Yet, diversity does not mean equality, and entrepreneurs and hackers continued to represent exceptionalism – both in terms of being different from, and better than, the average student – embodying the right-tail end of the bell-curve in terms of grades, passion, and performance.

I have explored throughout this dissertation how knowledges, gender, and personhoods are rendered technical as reality is translated into bounded and defined problems and framings that computer scientists and others have the power to manipulate and solve (Li 2007). This process is two-fold: first, reality is wrought and rendered and, second, these renderings diffract in ways that (re)shape, (re)produce, and transform the ways we understand and intra-act with the world (Myers 2014, 156). In relation to these rendering practices, Chapter 3 explored how the constitution, teaching, and learning of computer science knowledge effects technical renderings, as reality is defined, modelled, and programmed. Chapter 4 then considered renderings of gender, as a binary of women and men, are reproduced through teaching practices, Singaporean government policy, framings of algorithms and computing "problems." Chapter 5 considered how students' lives and identities are rendered technical as they are evaluated according to their grades and compared via bell-curves, or "normal" distributions. In Chapter 6, I consider how this comparison extends to evaluating performances of passion, which are often taken as proxies for skill and work-ethic. Chapter 7 then considered how binary renderings of gender in turn categorically render masculinities and femininities, and shape research on gender to centre on the "problem" of "women in" computing.

Each chapter also considers some of the diffractions produced through these renderings, with Chapters 6, 7, and 8 focusing on the affects and effects of reality rendered technical for students. Chapter 3 considered how reality is translated into computing worlds and correspondingly rendered natural to become part of (or treated the same as) reality, but wherein magic is real and computer scientists are the magicians. Chapter 4 considered how binary renderings of gender shape the ways students think about, understand, and do computer science and, more generally, the kinds of problems and solutions that are conceptualized and pursued in the discipline of computer science. Chapter 5 considered the ways that students learn to judge code, themselves, and one another according to technical criteria, fostering a sense of (trans)national competition. Chapter 6 further explores practices of self-judgment through the cultivation and performance of passion as (some) students work towards embodying a hegemonic personhood in computer science. This personhood was exemplified in the student hacker group, who worked to cultivate a *habitus* of independent learning and entrepreneurialism. The centring of research on "the woman problem," discussed in Chapter 7, is itself a rendered diffraction of renderings of binary gender, discussed in Chapter 4. Finally, Chapter 8 considered how the intra-action of hegemonic values of entrepreneurial passion and gender norms torque students' future aspirations and plans, as well as their (gendered) senses of self.

A theme that has woven throughout this dissertation is that of intra-actions across scales. I explore the meeting of national policies of the Singaporean government with transnational discourses and practices of the discipline of computer science and the tech industry, and how these are enacted by particular professors and students at Temasek University. Research on globalization (and on technologies) has explored the local and particular ways that international practices, brands, products, and technologies have been adopted, adapted, and reconfigured in particular contexts and by particular people (Caldwell 2004; Oudshoorn and Pinch 2003; Grinshpun 2014; Tsing 2005). I have similarly considered how knowledges, curricula, discourses, and practices of the computer science discipline and the tech industry, which have tended to originate in the US, have been adopted and adapted in Singapore as part of projects of national economic and technological development. I have also explored how the computer science department at Temasek University has centered its curriculum around guidelines from international organizations such as the ACM and the IEEE, as well as ubiquitously using internationally developed programming languages, computing technologies, and discourses, narratives, and histories based in American computer science.

Yet, as I have also shown, these transnational intra-actions entail directional frictions, such that persons and expertise, knowledges, and objects (e.g. textbooks) from the US are comparatively "mobile at birth" and able to move more easily to Singapore and other places than movement in the opposite direction (Takhteyev 2012; Tsing 2005). Students are driven to compete for grades and employment, both with foreign talents, whom Singaporean discourse and policy treat as privileged subjects, and with (sometimes imagined) other students around the world with comparable education and expertise. At the same time, Silicon Valley is valorized as the vanguard of future-making (Suchman 2011), continually positioning Singapore and its citizens – and most people and places outside of Silicon Valley – as entrepreneurially deficient, while also justifying and (re)producing such directional frictions. Scholarship bonds, visa restrictions, and connections to people and place, discussed in Chapter 8, further constrain the mobility of actors in and from Singapore. Moreover, these transnational intra-actions privilege and reproduce values and practices from the US, and other "Western" countries.

Ethnographic research on computing and technology design and development has shown the revolutionary new social forms that these practices engender (Coleman 2013; Kelty 2008). Chapters 3 and 4, however, additionally considered the colonization of places such as Singapore with specifically rendered and gendered (US-centric) computer science knowledges, practices, curricula, and technologies, adding to the growing body of literature in anthropology critically examining practices and cultures of computing design and development (Amrute 2014, 2010, 2016; Biao 2006; Forsythe 2001; Suchman 2007b, 2011a, Takhteyev 2012). In this regard, Chapter 4, for example, considered the (re)production of heteronormative gender binaries as part of (trans)national computer science education. Moreover, the computer science discipline is implicitly reproduced as a teleological narrative of American and Western "progress."

In the case of heteronormative gender binaries, knowledges and practices in transnational computer science largely align with Singaporean state policies relating to gender. However, the growing promotion of LGBTQ+ rights in the US and within the international tech industry point to the potential for conflicting intra-actions in the future, illustrating tensions across scales. Similarly, the government's adoption of Silicon Valley as a model for technological and economic success towards national technological development, alongside the promotion of passion and entrepreneurialism, produces the paradoxical effect of summoning in students the desire to *go to* Silicon Valley and become those very mobile passionate and entrepreneurial subjects.

The operation of hidden or implicit curricula in computer science and at Temasek University (re)produces these values, behaviours, and tensions. I have sought to elucidate the multiple and conflicting operations of hidden curricula along different intersecting lines – gender, citizenship and transnationalism, and neoliberalism. However, as with the reproduction of heteronormative gender binaries, these discourses and practices in computer science education at Temasek University often dovetailed to reinforce hegemonic norms and values, adding to the large body of scholarship showing the role of education in shaping student subjectivities and futures in particular and unequal ways (Apple 1980, 2004; Bowles and Gintis 1976; Fife 1992, 1994; Giroux and Purpel 1983; MacLeod 2009; Willis 1981). In particular, Chapters 5 and 6 showed how seemingly conflicting discourses relating to competition and cooperation in learning and performing expertise are infused with and shaped by logics of neoliberal individuality.

Students are fostered as independent learners and networkers who are compared and must compete with one another (in terms of grades and employment, for example). Students learn to mobilize resources as part of that competition and independent pursuit of knowledge. Moreover, students were summoned to *want* to learn technological knowledge and pursue personal projects above and beyond the requirements of courses or work, due to their focused and singular "passion." These values were represented in the hegemonic personhood, performatively enacted by members of the student hacker group, to which students compared themselves and their level of "passion." Passion thus worked as a kind of "cruel optimism" (Berlant 2006), normalizing precarious (over)work for themselves and others, but based on and enacted through their affective passion for technological pursuits and desires to "change the world".

Research on hidden curricula connects neatly with research from human-computer interaction and feminist technology studies, which points to the ways gender, and other values and norms, become embedded in and are co-produced with the design, production, and use of technologies (Cockburn and Ormrod 1993; Faulkner 2001; Wajcman 1991). Feminist technology studies scholars, however, have argued that the construction of gender has largely gone unaddressed in studies of gender and technology of gender-technology co-production (Landström 2007; Vitores and Gil-Juárez 2015). Chapters 4, 7, and 8 thus explore the gendered norms, values, and behaviours that are reproduced through computer science knowledges, technologies, and education. In particular, the multi-scalar intra-actions of computing technologies with computer science's overt and hidden curricula work to mutually reinforce gender norms, particularly heteronormative gender binaries, as well as norms relating to masculinities and femininities. This mutual (re)production is achieved through the intra-action of transnational traditions of framing and understanding computer science knowledge in terms of binary genders (seen, for example, in the "stable marriage problem" and stories about "Alice" and "Bob" sending encrypted messages to one another), dovetailing with teaching practices (including language usage and teaching examples), discourses and practices in the computer science department at Temasek University and by the (trans)national tech industry, and research on gender and computer science (or, more often, women in computer science).

Moreover, practices of rendering natural – enacted through teaching computer science knowledge as largely ahistorical and acontextual, and rendering and evaluating students and other persons in terms of statistical comparisons of men and women and their "qualities" – (re)produce an anti-politics of gender and computing. As discussed in Chapters 3 and 4, discourses and practices that mutually reinforce and (re)produce gender binaries and norms, along with systems of thought and practice cultivated in computer science of framing computing as based on solving problems, govern the possible ways for students and professors to think about and do computer science, and to think about and do gender in relation to computer science. The tendency to categorize and render technical recursively reproduces a focus on gender binaries, and the "problem" of women in computing, foreclosing possibilities for thinking about and doing gender and computing differently, and about what are valid or real problems in computer science.

Chapter 8 brings these multiple threads and themes together, while simultaneously pulling them apart. I consider how students' lives and identities are torqued as they are summoned to inhabit norms of femininity in Singapore, for example, which center on marriage and motherhood and support patriarchal renderings of family and nation (Bowker and Star 1999; Heng and Devan 1995). At the same time, norms in computer science call upon students to become unencumbered mobile entrepreneurial persons with a singular dedication and passion for computing and technology. Conflicts produced through the intra-action of gender norms and values of this hegemonic personhood, however, open spaces for reconfigurations. Some students contested these norms, worked to branch away from their constraints, and navigated moving futures. These students thereby complicated gender binaries and the relationship between passion and becoming a "good" computer scientist, and contested the necessity and desire for entrepreneurial mobility. Students thus come to inhabit, aspire to, and reach for gender and computer science norms (Mahmood 2005, 23), even as some work to reconfigure the boundaries and meanings of their subjectivation.

Donna Haraway argues that technologies are materializations of particular "figurations": they assemble tropes, discourses, images, and associations that enact or "figure" subjects and objects in particular ways (Haraway 1997, 11). Computing worlds concomitant with practices of rendering technical, and their diffractions, have suffused reality. We use apps and devices track our every footstep and heartbeat; we record our personal thoughts and memories, posting them on social media or in digital journals; we make financial transactions, consume and create media, learn, teach, read, write, and communicate, all using computing technologies. There is an increasing imperative for "smart" cities, where our every movement is tracked, recorded, and analyzed – our data "fed" into the network to ostensibly make it even "smarter." Computer scientists (and computer science as a discipline) have thus, in a sense, created a self-fulfilling prophecy wherein the significance of computer science in and to the world is produced through and by computer science theory and practice (Merton 1948). This dissertation has, however, shown how computer science also figures and (re)produces inequalities relating to gender, neoliberal labour, and citizenship, which torque students' lives and subjectivities.

At the same time, Lucy Suchman suggests – in relation to human-machine relationships in Artificial Intelligence practices – that "one form of intervention into current practices of technology development, then, is through a critical consideration of how humans and machines are currently figured in those practices and how they might be figured – and *configured* – differently" (Suchman 2007b, 227). As such, this dissertation has also shown how, in the spaces made through conflicts between gender norms and neoliberal values, some students challenged and reconfigured these structures through their performances of different ways of doing computer science and becoming/being a computer scientist. Students' aspirations, dissatisfactions, passions, and resistances thus point to possible points of intervention for reconfiguring the intra-actions of gender, computer science education, and (trans)national citizenship.

Bibliography

"passion, n." 2017. *OED Online*. Oxford University Press. http://www.oed.com./view/Entry/138504?isAdvanced=false&result=1&rskey=K9D7oU&, accessed 3 October 2017.

"About Hackerspace.SG." n.d. *hackerspace.sg.* https://hackerspace.sg/about/, accessed 5 November 2017.

"About Us." 2016. *TechInAsia.* https://www.techinasia.com/about-us, accessed November 5, 2017.

"Avoidance of marriages between persons of the same sex." 2008. *Women's Charter (Chapter 353), Part III Solemnization of Marriages*. Singapore: Attourney General's Chambers. http://statutes.agc.gov.sg/aol/search/display/view.w3p;ident=4bb9e463-04f9-430b-ad5c-68492deeb27a;page=0;query=DocId%3A%22f0897dd7-1f3a-45a9-b1e7-ba30fef2dbba%22%20Status%3Ainforce%20Depth%3A0;rec=0#pr12-he-.

"Making Google a Workplace for Everyone." 2014. *Google Diversity*. http://www.google.com/diversity/at-google.html#tab=tech.

"Outrages on Decency." 2008. *XVI - Offences Affecting the Human Body*. Singapore: Attourney General's Chambers. http://statutes.agc.gov.sg/aol/search/display/view.w3p;ident=c834c73a-3531-48b0-8040-4450d41d1351;page=0;query=DocId:%22025e7646-947b-462c-b557-60aa55dc7b42%22 Status:inforce Depth:0;rec=0#pr377A-he-.

"Women's Charter." 2008. *Women's Charter (Chapter 353).* Singapore: Attourney General's Chambers. http://statutes.agc.gov.sg/aol/search/display/view.w3p;page=0;query=DocId%3A%22f0897dd7-1f3a-45a9-b1e7-ba30fef2dbba%22%20Status%3Ainforce%20Depth%3A0;rec=0;whole=yes#legis.

Abbate, Janet. 2012. *Recoding Gender: Women's Changing Participation in Computing*. Cambridge, MA: MIT Press.

———. 2017. "From Handmaiden to 'Proper Intellectual Discipline': Creating a Scientific Identity for Computer Science in 1960s America." In *Communities of Computing: Computer Science and Society in the ACM*, edited by Thomas J Misa, 25–48. New York, NY and San Rafael, CA: Association for Computing Machinery and Morgan & Claypool Publishers.

Abu-Lughod, Lila. 1990. "Can There Be a Feminist Ethnography?" *Women and Performance: A Journal of Feminist Theory* 5 (1): 7–27.

———. 1991. “Writing Against Culture.” In *Recapturing Anthropology: Working in T He Present*, edited by Richard G. Fox, 137–62. Santa Fe, NM: School of American Research Press.

Acker, Sandra, and Keith Oatley. 1993. “Gender Issues in Education for Science and Technology: Current Situation and Prospects for Change.” *Canadian Journal of Education / Revue Canadienne de L’éducation* 18 (3): 255–72.

ACM Curriculum Committee on Computer Science. 1968. “Curriculum 68: Recommendations for Academic Programs in Computer Science.” *Communications of the ACM* 11 (3): 151–97.

Akkuş, Mahmut. 2015. “Answer to what can I do to be a good computer scientist?” *Quora.* 7 October 2015. https://www.quora.com/What-can-I-do-to-be-a-good-computer-scientist/answer/Mahmut-Akku%C5%9F, accessed 5 November 2017.

Akrich, Madeleine. 1992. “The De-Scription of Technological Objects.” In *Shaping technology/Building Society:Studies in Sociotechnical Change*, edited by Wiebe E. Bijker and John Law, 205–24. Cambridge, MA: MIT Press.

Alsheikh, Tamara, JA Rode, and SE Lindley. 2011. “(Whose) Value-Sensitive Design: A Study of Long-Distance Relationships in an Arabic Cultural Context.” *Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work (CSCW 2011)*, 75–84. http://dl.acm.org/citation.cfm?id=1958836.

Ambikaipaker, Mohan. 2015. “Failed Asian Values Subjects and Neoliberal State Discourse in Malaysia and Singapore.” *Journal of Intercultural Studies* 36 (1): 1–16.

Ames, Elizabeth. “Teaching kids to code isn’t just about tech jobs – it’s about preparing for the future.” *Mashable.* 25 September 2017. http://mashable.com/2017/09/25/yes-kids-should-learn-to-code/#w2XTI0iyGOqK, accessed 23 January 2018.

Amit, Vered, and Pauline Gardiner Barber. 2015. “Mobility and Cosmopolitanism: Complicating the Interaction between Aspiration and Practice.” *Identities* 22 (5): 543–50.

Amrute, Sareeta. 2010. “Living and Praying in the Code: The Flexibility and Discipline of Indian Information Technology Workers (ITers) in a Global Economy.” *Anthropological Quarterly* 83 (3): 519–50.

———. 2014. “Proprietary Freedoms in an IT Office: How Indian IT Workers Negotiate Code and Cultural Branding.” *Social Anthropology* 22 (1): 101–17.

———. 2016. *Encoding Race, Encoding Class: Indian IT Workers in Berlin*. Durham, NC: Duke University Press.

Ang, Ien, and Jon Stratton. 1995. “The Singapore Way of Multiculturalism: Western Concepts/Asian Cultures.” *Sojourn: Journal of Social Issues in Southeast Asia* 10 (1): 66–89.

Appadurai, Arjun. 1986. “Theory in Anthropology: Center and Periphery.” *Comparative Studies in Society and History* 28 (2): 356–74.

Apple, Michael W. 1980. “The Other Side of the Hidden Curriculum: Correspondence Theories and the Labor Process.” *Interchange* 11 (3): 5–22.

———. 1992. “The Text and Cultural Politics.” *Educational Researcher* 21 (7): 4–19. doi:10.3102/0013189X021007004.

———. 2004. *Ideology and Curriculum*. 3rd ed. New York, NY: Routledge.

Aronson, Joshua, Geoffrey Cohen, and Wendy McColskey. 2009. “Reducing Stereotype Threat in Classrooms: A Review of Social-Psychological Intervention Studies on Improving the Achievement of Black Students.”

Aronson, Joshua, Carrie B. Fried, and Catherine Good. 2002. “Reducing the Effects of Stereotype Threat on African American College Students by Shaping Theories of Intelligence.” *Journal of Experimental Social Psychology* 38 (2): 113–25.

Arun, Mahizhnan, and Mui Teng Yap. 2000. “Singapore: The Development of an Intelligent Island and Social Dividends of Information Technology.” *Urban Studies* 37 (10): 1749–56.

Ashcraft, Catherine, and Sarah Blithe. 2010. “Women in IT: The Facts.” Boulder, CO: National Center for Women & Information Technology (NCWIT). www.ncwit.org/thefacts.

Ashcraft, Catherine, Elizabeth Eger, and Michelle Friend. 2012. “Girls in IT : The Facts.” Boulder, CO.

Austin, Linda. 1983. “Computers in Class: S’Pore Could be No.1.” *The Straits Times*, 22 March 1983. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19830322-1.2.35.aspx.

Backus, John, Jan Lee, and George Ryckman. 1981. “The History of I, II and III.” In *History of Programming Languages*, edited by Richard L. Wexelblat, 25–45. New York, NY: Academic Press, Inc.

Barad, Karen. 1998. “Getting Real: Technoscientific Practices and the Materialization of Reality.” *Differences: A Journal of Feminist Cultural Studies* 10 (2): 87–128.

———. 2003. “Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter.” *Signs* 28 (3). The University of Chicago Press: 801–31.

———. 2007. *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Durham, NC: Duke University Press.

———. 2010. “Quantum Entanglements and Hauntological Relations of Inheritance: Dis/continuities, Spacetime Enfoldings, and Justice-to-Come.” *Derrida Today* 3 (2): 240–68.

Barr, MichaelD. 2006. “Racialised Education in Singapore.” *Educational Research for Policy and Practice* 5 (1): 15–31.

Beauvoir, Simone de. 1953. *The Second Sex*. Edited by H. M. Parshley. London, UK: Jonathan Cape.

Beck, Ulrich. 2004. “Cosmopolitical Realism: On the Distinction between Cosmopolitanism in Philosophy and the Social Sciences.” *Global Networks* 4 (2): 131–56.

Beckwith, L., and M. Burnett. 2004. “Gender: An Important Factor in End-User Programming Environments?” In *Proceedings of the 2004 IEEE Symposium on Visual Languages and Human Centric Computing (VLHCC’04)*, 107–14. Rome: IEEE. doi:10.1109/VLHCC.2004.28.

Beckwith, Laura, Margaret Burnett, Susan Wiedenbeck, Curtis Cook, Shraddha Sorte, and Michelle Hastings. 2005. “Effectiveness of End-User Debugging Software Features.” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’05*, 10. doi:10.1145/1054972.1055094.

Ben-Amos, Dan. 1984. “The Seven Strands of Tradition : Varieties in Its Meaning in American Folklore Studies.” *Journal of Folklore Research* 21 (2/3): 97–131.

Benjamin, Geoffrey. 1997. “The Cultural Logic of Singapore’s ‘Multiracialism.’” In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 67–85. Singapore: Times Academic Press.

Benner, Katie. 2017. “Women in Tech Speak Frankly on Culture of Harassment.” *The New York Times.* 30 June 2017. https://www.nytimes.com/2017/06/30/technology/women-entrepreneurs-speak-out-sexual-harassment.html.

Bereznak, Alyssa. 2017. “Can Software Help Eliminate Workplace Bias? Anonymity in the office could work wonders.” *The Ringer.* 23 February 2017. https://theringer.com/workplace-bias-baloonr-software-ed0ce6f00a8c, accessed 27 July 2017.

Berg, AJ. 1996. “Digital Feminism.” Trondheim: Norweigian University of Science and Technology.

———. 1999. “A Gendered Socio-Technical Construction: The Smart House.” In *The Social Shaping of Technology*, edited by Donald Mackenzie and Judy Wajcman, 301–13. Buckingham, UK and Philadelphia, PA: Open University Press.

Bergenhenegouwen, G. 1987. “Hidden Curriculum in the University.” *Higher Education* 16 (5): 535–43.

Berlant, Lauren. 2006. “Cruel Optimism.” *Differences* 17 (3): 20–36. doi:10.1215/10407391-2006-009.

———. 2007. “Slow Death (Sovereignty, Obesity, Lateral Agency).” *Critical Inquiry* 33: 754–80.

Berry, David M. 2011. *The Philosophy of Software: Code and Mediation in the Digital Age*. Basingstoke, UK: Palgrave MacMillan.

Bettie, Julie. 2000. “Women without Class : Chicas , Cholas , Trash , and the Presence / Absence of Class Identity.” *Signs* 26 (1): 1–35.

Beyer, Sylvia, Kristina Rynes, Julie Perrault, Kelly Hay, and Susan Haller. 2003. “Gender Differences in Computer Science Students.” *SIGCSE Bulletin* 35 (1): 49–53. doi:10.1145/792548.611930.

Biao, Xiang. 2006. *Global “Body Shopping.”* Princeton, NJ: Princeton University Press.

Blickenstaff, Jacob Clark. 2005. “Women and Science Careers: Leaky Pipeline or Gender Filter?” *Gender and Education* 17 (4): 369–86.

Bornat, Richard. 2014. “Camels and Humps: A Retraction.” London, UK. http://www.eis.mdx.ac.uk/staffpages/r_bornat/papers/camel_hump_retraction.pdf.

Bort, Julie. 2016. “One of Docker's star engineers got so many death and rape threats that the company hired private detectives to protect her.” *Business Insider.* 25 April 2016. http://www.businessinsider.com/death-rape-threats-for-docker-engineer-2016-4.

Bourdieu, Pierre. 1967. “Systems of Education and Systems of Thought.” *International Social Science Journal* 19 (3): 338–58.

———. 1977. *Outline of a Theory of Practice*. Translated by Richard Nice. Cambridge, UK: Cambridge University Press.

———. 1984. *Distinction: A Social Critique of the Judgement of Taste*. Translated by Richard Nice. Cambridge, MA: Harvard University Press.

Bourdieu, Pierre, and Jean-Claude Passeron. 1977. *Reproduction in Education, Society and Culture*. Translated by Richard Nice. London, UK: Sage Publications.

Bowker, Geof. 1993. "How to Be Universal: Some Cybernetic Strategies, 1943-70." *Social Studies of Science* 23 (1): 107–27.

Bowker, Geoffrey C, and Susan Leigh Star. 1999. *Sorting Things Out: Classification and Its Consequences*. Cambridge, MA: MIT Press.

Bowles, Samuel, and Herbert Gintis. 1976. *Schooling in Capitalist America: Educational Reform and the Contradictions of Economic Life*. New York, NY: Basic Books.

Brahnam, Sheryl, Marianthe Karanikas, and Margaret Weaver. 2011. "(Un)dressing the Interface: Exposing the Foundational HCI Metaphor 'Computer Is Woman.'" *Interacting with Computers* 23 (5): 401–12. doi:10.1016/j.intcom.2011.03.008.

Brandes, Stanley. 1980. *Metaphors of Masculinity: Sex and Status in Andalusian Folklore*. Philadelphia, PA: University of Pennsylvania Press.

Bray, Francesca. 2007. "Gender and Technology." *Annual Review of Anthropology* 36 (1): 37–53.

Breslin, Samantha. 2011. "Living with Music: An Ethnography of Sessions in St. John's, Newfoundland." St. John's, NL: Memorial University of Newfoundland.

———. 2013. "01010000 01001100 01000001 01011001: Play Elements in Computer Programming." *American Journal of Play* 5 (3): 357–82.

———. 2015. "The Anti-Politics of Women in Tech." *Platypus: The CASTAC Blog.* http://blog.castac.org/2015/12/anti-politics/.

———. 2017. "The song that never ends: About men and women in tech." *The Independent.* 16 August, 2017. http://theindependent.ca/2017/08/16/the-song-that-never-ends-about-men-and-women-in-tech/.

Breslin, Samantha, and Bimlesh Wadhwa. 2014a. "EnGendering Interaction Design." In *3rd International Conference on User Science and Engineering (I-USEr)*, 292–95. doi:10.1109/IUSER.2014.7002719.

———. 2014b. "Exploring Nuanced Gender Perspectives within the HCI Community."

In *Proceedings of the India HCI 2014 Conference on Human Computer Interaction - IHCI '14*, 45–54. New York, New York, USA: ACM Press. doi:10.1145/2676702.2676709.

———. 2015. “Towards a Gender HCI Curriculum.” In *CHI'15 Extended Abstracts on Human Factors in Computing Systems*, 1091–96. Seoul, South Korea: ACM Press.

———. 2017. “Gender and Human-Computer Interaction.” In *The Wiley Handbook for Human-Computer Interaction*, edited by Kent Norman and Jurek Kirakowski, 71-88. Wiley & Sons, Ltd.

Briggs, Jean L. 1998. *Inuit Morality Play: The Emotional Education of a Three-Year-Old*. New Haven, CT: Yale University Press.

Bronner, Simon, ed. 2005. *Manly Traditions: The Folk Roots of American Masculinities*. Bloomington, IN: Indiana University Press.

Brooks Jr., Frederick P. 1995. *The Mythical Man-Month*. 2nd ed. Boston, MA: Addison-Wesley.

Brubaker, Rogers. 2004. *Ethnicity without Groups*. Cambridge, MA: Harvard University Press.

Bryant, Rebecca. 2001. “An Aesthetics of Self: Moral Remaking and Cypriot Education.” *Comparative Studies in Society and History* 43 (3): 583–614.

———. 2005. “The Soul Danced into the Body: Nation and Improvisation in Istanbul.” *American Ethnologist* 32 (2): 222–38.

Burnett, Margaret, SD Fleming, Shamsi Iqbal, Gina Venolia, Vidya Rajaram, Umer Farooq, Valentina Grigoreanu, and Mary Szerwinski. 2010. “Gender Differences and Programming Environments: Across Programming Populations.” In *ESEM '10 Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. New York, NY: ACM. http://dl.acm.org/citation.cfm?id=1852824.

Burrell, Jenna. 2012. *Invisible Users: Youth in the Internet Cafes of Urban Ghana*. Cambridge, MA: MIT Press.

Bury, Rhiannon. 2011. “She's Geeky : The Performance of Identity among Women Working in IT.” *International Journal of Gender, Science and Technology* 3: 34–53.

Butler, Judith. 1988. “Performative Acts and Gender Constitution: An Essay in Phenomenology and Feminist Theory.” *Theatre Journal* 40 (4): 519–31.

———. 1993. *Bodies That Matter: On the Discursive Limits of "Sex." Routledge*. Vol. 36. New York: Routledge.

———. 1999. *Gender Trouble: Feminism and the Subversion of Identity*. 10th Anniv. New York, NY: Routledge.

Caldwell, Melissa L. 2004. "Domesticating the French Fry: McDonald's and Consumerism in Moscow." *Journal of Consumer Culture* 4 (1): 5–26.

Camp, Tracy. 1997. "The Incredible Shrinking Pipeline." *Communications of the ACM* 40 (10): 103–10.

Campbell, Scott. 2017. "The Development of Computer Professionalization in Canada." In *Communities of Computing: Computer Science and Society in the ACM*, edited by Thomas J Misa, 173–98. Association for Computing Machinery and Morgan & Claypool Publishers.

Carrigan, Coleen. 2017. "'Different Isn't Free': Gender @ Work in a Digital World." *Ethnography* 0 (0): 1–24. doi:10.1177/1466138117728737.

Carrithers, M., M. Candea, K. Sykes, M. Holbraad, and S. Venkatesan. 2010. "Ontology Is Just Another Word for Culture: Motion Tabled at the 2008 Meeting of the Group for Debates in Anthropological Theory, University of Manchester." *Critique of Anthropology* 30 (2): 152–200.

Carver, Terrell. 2010. "Materializing the Metaphors of Global Cities: Singapore and Silicon Valley." *Globalizations* 7 (3): 383–93.

Cauterucci, Christina. 2015. "New Chrome App Helps Women Stop Saying 'just' and 'sorry' in emails" *Slate*. 29 December 2015. http://www.slate.com/blogs/xx_factor/2015/12/29/new_chrome_app_helps_women_stop_saying_just_and_sorry_in_emails.html, accessed 27 July 2017.

Cech, Erin, Brian Rubineau, Susan Silbey, and Caroll Seron. 2011. "Professional Role Confidence and Gendered Persistence in Engineering." *American Sociological Review* 76 (5): 641–66.

Ceci, Stephen J, Wendy M Williams, and Susan M Barnett. 2009. "Women's Underrepresentation in Science: Sociocultural and Biological Considerations." *Psychological Bulletin* 135 (2): 218–61.

Ceruzzi, Paul E. 1998. *A History of Modern Computing.* Cambridge, MA: MIT Press.

Chan, Heng Chee. 2015. "National Report for the Universal Periodic Review Prepared for the Human Rights Council." Singapore.

Chasin, Alexandra. 1995. “Class and Its Close Relations: Identities Among Women, Servants, and Machines.” In *Posthuman Bodies*, edited by Judith Halberstram and Ian Livingston, 73–96. Bloomington: Indiana University Press.

Cheng, Kenneth and Toh Ee Ming. 2016. “NUS suspends student-organised orientation activities after 'inappropriate' behaviour.” *Today Online.* 29 July 2016. https://admin.todayonline.com/singapore/nus-suspends-all-student-organised-freshman-orientation-activities-following-inappropriate.

Cheng, Yi’En. 2014. “Time Protagonists: Student Migrants, Practices of Time, and Cultural Construction of the Singapore-Educated Person.” *Social & Cultural Geography* 15 (4): 385–405.

Cheng, Yi’En, Brenda S.A. Yeoh, and Juan Zhang. 2014. “Still ‘breadwinners’ and ‘providers’: Singaporean Husbands, Money and Masculinity in Transnational Marriages.” *Gender, Place & Culture*, 1–17. doi:10.1080/0966369X.2014.917282.

Cheryan, Sapna, Victoria C. Plaut, Caitlin Handron, and Lauren Hudson. 2013. “The Stereotypical Computer Scientist: Gendered Media Representations as a Barrier to Inclusion for Women.” *Sex Roles* 69 (1–2): 58–71.

Chia, Alice. 2015. “100 participate in hackathon to develop apps on haze-related issues.” *Channel News Asia*. 13 October 2015. http://www.channelnewsasia.com/news/singapore/100-participate-in-hackathon-to-develop-apps-on-haze-related-iss-8246784, accessed 27 July 2017.

Chia, Yeow Tong. 2011a. “The Elusive Goal of Nation Building: Asian/Confucian Values and Citizenship Education in Singapore During The 1980s.” *British Journal of Educational Studies* 59 (4): 383–402.

———. 2011b. “The Loss of the ‘World-Soul’? Education , Culture and the Making of the Singapore Developmental State , 1955 – 2004.” University of Toronto.

Chong, Terence. 2005. “Singapore’s Cultural Policy and Its Consequences.” *Critical Asian Studies* 37 (4): 553–68.

Chua, Alfred. 2014. “NLB pulls two children’s books that ‘don’t promote family values’.” *Today Online*. 9 July 2014. http://www.todayonline.com/singapore/nlb-pulls-two-childrens-books-dont-promote-family-values?singlepage=true.

Chua, Alfred. 2017. “Foreign companies’ application to support Pink Dot rejected.” *Today Online*. 16 June 2017. http://www.todayonline.com/singapore/foreign-companies-application-support-pink-dot-rejected.

Chua, Grace. 2013. "3 girls who shaved head bald for charity told to wear wigs in school by principal." *The Straits Times.* 2 August 2013. www.straitstimes.com/singapore/3-girls-who-shaved-head-bald-for-charity-told-to-wear-wigs-in-school-by-principal.

Churchill, Elizabeth F. 2010. "Sugared Puppy-Dog Tails: Gender and Design." *Interactions* 12(2): 52-56.

Clancey, Gregory. 2012. "Intelligent Island to Biopolis: Smart Minds, Sick Bodies and Millennial Turns in Singapore." *Science, Technology and Society* 17 (1): 13–35.

Clawson, Rosalee A. 2000. "The Race Coding of Poverty in American Government College Textbooks." *Howard Journal of Communications* 11 (3): 179–88.

Clifford, James, and George E. Marcus, eds. 1986. *Writing Culture: The Poetics and Politics of Ethnography*. Berkeley, CA: University of California Press.

Cockayne, Daniel G. 2015. "Entrepreneurial Affect: Attachment to Work Practice in San Francisco's Digital Media Sector." *Environment and Planning D: Society and Space* 34 (3): 456–73.

Cockayne, Daniel G., and Lizzie Richardson. 2017. "Queering Code/space: The Co-Production of Socio-Sexual Codes and Digital Technologies." *Gender, Place & Culture* 524 (July). Routledge: 1–17. doi:10.1080/0966369X.2017.1339672.

Cockburn, Cynthia, and Susan Ormrod. 1993. *Gender and Technology in the Making*. Thousand Oaks, CA: Sage.

Cohn, Carol. 1987. "Sex and Death in the Rational World of Defense Intellectuals." *Signs: Journal of Women in Culture and Society* 12 (4): 687–718.

Coleman, Gabriella. 2009. "Code Is Speech: Legal Tinkering, Expertise, and Protest among Free and Open Source Software Developers." *Cultural Anthropology* 24 (3): 420–54.

———. 2013. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton, NJ: Princeton University Press.

Coleman, Gabriella, and Alex Golub. 2008. "Hacker Practice: Moral Genres and Cultural Articulations of Liberalism." *Anthropological Theory* 8 (3): 255–77.

Collins, Francis L., Ravinder Sidhu, Nick Lewis, and Brenda S.A. Yeoh. 2014. "Mobility and Desire: International Students and Asian Regionalism in Aspirational Singapore." *Discourse: Studies in the Cultural Politics of Education* 35 (5): 661–76.

Commit Strip. 2014. “When I help a rookie coder fix his queries.” 1 August 2014. http://www.commitstrip.com/en/2014/08/01/when-i-help-a-rookie-coder-fix-his-queries, accessed 5 November 2017.

Comunian, Roberta, and Can-Seng Ooi. 2015. “Global Aspirations and Local Talent: The Development of Creative Higher Education in Singapore.” *International Journal of Cultural Policy*, 1–22. doi:10.1080/10286632.2015.1101085.

Connell, R. W. 1987. *Gender and Power: Society, the Person and Sexual Politics.* Cambridge, UK: Polity Press.

———.2005. *Masculinities*. 2nd ed. Berkeley, CA: University of California Press.

Cooper, Alan. 2004. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Indianapolis, IN: Sams Publishing. doi:10.1007/978-3-322-99786-9_1.

Cooper, Frederick, and Rogers Brubaker. 2005. “Identity.” In *Colonialism in Question: Theory, Knowledge, History*, edited by Frederick Cooper, 59–90. Berkeley: University of California Press.

Coopmans, Catelijne, Janet Vertesi, Michael Lynch, and Steve Woolgar, eds. 2014. *Representation in Scientific Practice Revisited*. Cambridge, MA: MIT Press.

Cormen, Thomas H., Ronald L. Rivest, Clifford Stein, and Charles E Leiserson. 2009. *Introduction to Algorithms*. 3rd ed. Cambridge, MA: MIT Press.

Cotton, Debby, Jennie Winter, and Ian Bailey. 2013. “Researching the Hidden Curriculum: Intentional and Unintended Messages.” *Journal of Geography in Higher Education* 37 (2): 192–203.

Crapanzano, Vincent. 1986. “Hermes’ Dilemma: The Masking of Subversion in Ethnographic Description.” In *Writing Culture: The Poetics and Politics of Ethnography*, 51–76. Berkeley, CA: University of California Press.

Crenshaw, Kimberle. 1989. “Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics.” *University of Chicago Legal Forum* 1989 (1): 139–67.

———. 1991. “Mapping the Margins: Intersectionality, Identity Politics, and Violence against Women of Color.” *Stanford Law Review* 43 (6). Stanford Law Review: 1241–99. doi:10.2307/1229039.

Currier, Dianne. 2003. “Feminist Technological Futures.” *Feminist Theory* 4 (3): 321–38.

Cutts, Quintin, Emily Cutts, Stephen Draper, Patrick O’Donnell, and Peter Saffrey. 2010.

“Manipulating Mindset to Positively Influence Introductory Programming Performance.” In *SIGCSE ’10 Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 431–35. Milwaukee, WI: ACM. doi:10.1145/1734263.1734409.

Dambrot, Faye H., Michelle A. Watkins-Malek, S. Marc Silling, Rodney S. Marshall, and Jo Ann Garver. 1985. “Correlates of Sex Differences in Attitudes toward and Involvement with Computers.” *Journal of Vocational Behavior* 27 (1): 71–86.

Damore, James. 2017. “Google’s Ideological Echo Chamber: How bias clouds our thinking about diversity and inclusion.” July 2017. https://assets.documentcloud.org/documents/3914586/Googles-Ideological-Echo-Chamber.pdf, accessed 22 January 2017.

Denning, Peter J. 2009. “The Profession of IT Beyond Computational Thinking.” *Communications of the ACM* 52 (6): 28–30.

Desjarlais, Robert. 1996. “Struggling Along.” In *Things as They Are: New Directions in Phenomenological Anthropology*, edited by Michael Jackson, 70–93. Bloomington, IN: Indiana University Press.

Devlin, Hannah. 2017. “Two-year-olds should learn to code, says computing pioneer.” *The Guardian*, 20 August 2017, https://www.theguardian.com/technology/2017/aug/20/two-year-olds-should-learn-to-code-says-computing-pioneer, accessed 23 August 2017.

Deyo, Frederic C. 1997a. “Creating Industrial Community: Towards a Corporate Paternalist Society.” In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 363–73. Singapore: Times Academic Press.

———. 1997b. “The Emergence of Bureaucratic-Authoritarian Corporatism in Labour Relations.” In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 353–62. Singapore: Times Academic Press.

Dodson, Ll, S Sterling, and Jk Bennett. 2013. “Minding the Gaps: Cultural, Technical and Gender-Based Barriers to Mobile Use in Oral-Language Berber Communities in Morocco.” In *Proceedings of the Sixth International Conference on Information and Communication Technologies and Development*, 79–88. Cape Town: ACM. doi:10.1145/2516604.2516626.

Do, Anh Minh. 2013. “Failure versus Face: How Can Asia Produce Startups Despite its Culture?” *Tech in Asia.* 26 September 2013. https://www.techinasia.com/failure-face-asia-produce-startups-culture.

Douglas-Jones, Rachel, and Christopher Gad. 2015. “Computational Thinking as an

Ontological Program? Ethnographic Engagements with Digital Alterity." In *American Anthropological Association Annual Meetings*. Denver, CO.

Downey, Gary Lee. 1998. *The Machine in Me: An Anthropologist Sits Among Computer Engineers*. New York, NY: Routledge.

Dreeben, Robert. 1968. *On What Is Learned in School*. Reading, MA: Addison-Wesley Pub. Co.

Durndell, A. 1991. "The Persistence of the Gender Gap in Computing." *Computers & Education* 16 (4): 283–87.

Dweck, Carol S. 2008. "Mindsets and Math / Science Achievement." *The Opportunity Equation: Transforming Mathematics and Science Education for Citizenship and the Global Economy*. New York, NY: Carnegie Corporation of New York-Institute for Advanced Study Commission on Mathematics and Science Education. www.opportunityequation.org.

Dweck, Carol S. 2006. *Mindset: The New Psychology of Success*. New York, NY: Ballantine Books.

Dziallas, Sebastian, and Sally Fincher. 2017. "The History and Purpose of Computing Curricula (1960s–2000s)." In *Communities of Computing: Computer Science and Society in the ACM*, edited by Thomas J. Misa, 91–109. New York, NY and San Rafael, CA: Association for Computing Machinery and Morgan & Claypool Publishers.

Edwards, Laurie D. 1995. "Microworlds as Representations." In *Computers and Exploratory Learning*, edited by Andrea A. DiSessa, Celia Hoyles, Richard Noss, and Laurie D. Edwards, 127–54. Berlin, Heidelberg: Springer-Verlag.

Edwards, Paul N. 1990. "The Army and the Microworld: Computers and the Politics of Gender Identity" 16 (1): 102–27.

———. 1997. *The Closed World: Computers and the Politics of Discourse in Cold War America.* Cambridge, MA: MIT Press.

El-Bahey, Rehab, and Amir Zeid. 2013. "Women in Computing: A Case Study About Kuwait." In *Frontiers in Education Conference, 2013 IEEE*. Oklahoma City, OK: IEEE.

Ensmenger, Nathan. 2010a. "Making Programming Masculine." In *Gender Codes: Why Women Are Leaving Computing*, edited by Thomas J Misa, 115–42. Hoboken, NJ: IEEE Computer Society and John Wiley & Sons, Inc.

———. 2010b. *The Computer Boys Take over: Computers, Programmers, and the Politics of Technical Expertise*. Cambridge, MA: The MIT Press.

———. 2015. "'Beards, Sandals, and Other Signs of Rugged Individualism': Masculine Culture within the Computing Professions." *Osiris* 30 (1): 38–65.

Escobar, Arturo. 1994. "Welcome to Cyberia: Notes on the Anthropology of Cyberculture." *Current Anthropology* 35 (3): 211–31.

Everett, Caleb. 2013. "Independent Cross-Cultural Data Reveal Linguistic Effects on Basic Numerical Cognition." *Language and Cognition* 5 (1): 99–104.

Everett, Caleb, and Keren Madora. 2012. "Quantity Recognition among Speakers of an Anumeric Language." *Cognitive Science* 36 (1): 130–41.

Faulkner, Wendy. 2000. "The Power and the Pleasure? A Research Agenda for 'Making Gender Stick' to Engineers." *Science, Technology & Human Values* 25 (1): 87–119.

———. 2001. "The Technology Question in Feminism: A View from Feminist Technology Studies." *Women's Studies International Forum* 24 (1): 79–95.

———. 2007. "`Nuts and Bolts and People': Gender-Troubled Engineering Identities." *Social Studies of Science* 37 (3): 331–56.

Fekeke, George. 2014. "How to be a Good Developer." *Sitepoint*. 10 October 2014. www.sitepoint.com/good-developer, accessed 5 November 2017.

Ferguson, James. 1990. *The Anti-Politics Machine: Development and Depoliticization of Bureaucratic Power in Lesotho*. Cambridge: Cambridge University Press.

Ferguson, James, and Akhil Gupta. 2002. "Spatializing States : Toward an Ethnography of Neoliberal Governmentality." *American Ethnologist* 29 (4): 981–1002.

Field, John, and Natalie Morgan-Klein. 2010. "Studenthood and Identification : Higher Education as a Liminal Transitional Space." In *40th Annual Standing Conference on University Teaching and Research in the Education of Adults*. Stirling, Scotland.

Fife, Wayne. 1991. "A Certain Kind of Education: Education, Culture and Society in West New Britain." McMaster University.

———. 1992. "Crossing Boundaries: Dissolution as a Secondary Message of Education in Papua New Guinea." *International Journal of Educational Development* 12 (3): 213–21.

———. 1994. "Education in Papua New Guinea: The Hidden Curriculum of a New Moral Order." *City & Society* 7 (1): 139–62.

———. 2005. *Doing Fieldwork: Ethnographic Methods for Research in Developing Countries and Beyond*. New York, NY: Palgrave Macmillan.

Fischer, Michael M. J. 2013. “Biopolis: Asian Science in the Global Circuitry.” *Science Technology & Society* 18 (3): 379–404.

Fischer, Michael M.J. 1999. “Emergent Forms of Life: Anthropologies of Late or Postmodernities.” *Annual Review of Anthropology* 28: 455–78.

Florida, Richard, Charlotta Mellander, and Karen King. 2015. “The Global Creativity Index 2015.” Toronto, ON.

Foo Mey Kien. 1984. “Problem-solving will be stressed in Masters course.” *Singapore Monitor.* 11 January 1984. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/singmonitor19840111-2.2.6.11.

Foor, Cindy E., Susan E. Walden, Randa L. Shehab, and Deborah a. Trytten. 2013. “‘We Weren’t Intentionally Excluding Them…just Old Habits’: Women, (Lack of) Interest and an Engineering Student Competition Team.” In *2013 IEEE Frontiers in Education Conference (FIE)*, 349–55. IEEE. doi:10.1109/FIE.2013.6684846.

Forsythe, Diana. 2001. *Studying Those Who Study Us: An Anthropologist in the World of Artificial Intelligence*. Stanford, CA: Stanford University Press.

Fortes, Meyer. 1938. “Social and Psychological Aspects of Education in Taleland.” *Africa* 11(4): Supplement, 5-62.

Foucault, Michel. 1982. “The Subject and Power.” *Critical Inquiry* 8 (4): 777–95.

———. 1984. *Power: Essential Works of Foucault 1954-1984*. Edited by James D. Faubion. Translated by Robert Hurley and Others. New York, NY: The New Press.

———. 1990. *The Use of Pleasure: Volume 2 of The History of Sexuality*. Translated by Robert Hurley. New York, NY: Vintage Books.

———. 1991. *The Foucault Effect: Studies in Governmentality*. Edited by Graham Burchell, Colin Gordon, and Peter Miller. Chicago, IL: University of Chicago Press.

———. 1997a. *Ethics: Subjectivity and Truth*. Edited by Paul Rabinow. Translated by Robert Hurley and Others. *Essential Works of Foucault, 1954-1984*. New York, NY: The New Press.

———. 1997b. *“Society Must Be Defended”: Lectures at the College de France 1975-1976*. Edited by Mauro Bertani and Alessandro Fontana. Translated by David

Macey. New York, NY: Picador.

———. 2008. *The Birth of Biopolitics: Lectures at the College de France, 1978-79*. Edited by Michel Sellenart. Translated by Graham Burchell. New York, NY: Palgrave Macmillan.

Fowler, Susan J. 2017. “Reflecting on One Very, Very Strange Year at Uber.” *Susan J. Fowler: Blog*. 19 February 2017. https://www.susanjfowler.com/blog/2017/2/19/reflecting-on-one-very-strange-year-at-uber, accessed 5 November 2017.

Frankenberg, Ruth. 1993. *White Women, Race Matters: The Social Construction of Whiteness*. London, UK: Routledge.

Franklin, Sarah. 1995. “Science as Culture, Cultures of Science.” *Annual Review of Anthropology* 24: 163–84.

Freeman, Carla. 2014. *Entrepreneurial Selves : Neoliberal Respectability and the Making of a Caribbean Middle Class*. Durham, NC: Duke University Press.

Gaboury, Jacob. 2013. “On Uncomputable Numbers : The Origins of a Queer Computing.” *Media-N: Journal of the New Media Caucus* 9 (2).

Gair, Marina, and Guy Mullins. 2001. “Hiding in Plain Sight.” In *The Hidden Curriculum in Higher Education*, edited by Eric Margolis, 21–42. New York, NY: Routledge.

Gale, D., and L.S. Shapley. 1962. “College Admissions and the Stability of Marriage, 69-1, Jan 1962, p9.pdf.” *The American Mathematical Monthly* 69 (1): 9–15.

Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Gardner, Andrew, and David M. Hoffman, eds. 2006. *Dispatches from the Field: Neophyte Ethnographers in a Changing World*. Long Grove, IL: Waveland Press, Inc.

Geertz, Clifford. 1973. *The Interpretation of Cultures*. New York: Basic Books.

Gennep, Arnold van. 1960. *The Rites of Passage*. Translated by Monika B. Vizedom and Gabrielle L. Caffe. Chicago, IL: University of Chicago Press.

Gilbertson, Amanda. 2017. “Aspiration as Capacity and Compulsion: The Futures of Urban Middle-Class Youth in India.” In *Anthropological Perspectives on Student Futures: Youth and the Politics of Possibility*, edited by Amy Stambach and Kathleen D. Hall, 19–32. New York, NY: Palgrave MacMillan.

Gilroy, Paul. 1991. *"There Ain"t No Black in the Union Jack': The Cultural Politics of Race and Nation*. Chicago, IL: University of Chicago Press.

Giroux, Henry A, and David E Purpel. 1983. *The Hidden Curriculum and Moral Education: Deception or Discovery?* Berkeley, CA: McCutchan Pub. Corp.

Goh, Chin Lian and Mokhtar, Maryam. "Large turnout at Speakers' Corner for protest against Population White Paper." *The Straits Times.* 16 February 2013. http://www.straitstimes.com/singapore/large-turnout-at-speakers-corner-for-protest-against-population-white-paper.

Goh, Chok Tong. 1994. "Prime Minister's National Day Rally Speech." Singapore: Prime Minister's Office Singapore. http://www.nas.gov.sg/archivesonline/data/pdfdoc/1994NDRenglishspeech.pdf.

Goh, Chok Tong. 1997. "Shaping our future: Thinking Schools, Learning Nation." Singapore: Ministry of Education. http://ncee.org/wp-content/uploads/2017/01/Sgp-non-AV-2-PM-Goh-1997-Shaping-Our-Future-Thinking-Schools-Learning-Nation-speech.pdf.

Goh, Chok Tong. 1999. "National Day Rally Address by Prime Minister Goh Chok Tong, Speech in English on 22 August 1999 : First-World Economy, World-Class Home." Singapore: Prime Minister's Office Singapore. http://www.nas.gov.sg/archivesonline/speeches/view-html?filename=1999082202.htm

Goh, Daniel P.S. 2008. "From Colonial Pluralism to Postcolonial Multiculturalism: Race, State Formation and the Question of Cultural Diversity in Malaysia and Singapore." *Sociology Compass* 2 (1): 232–52.

Goh, Evelyn. 1993. "No short crop girls, say some schools." *The Straits Times*. 13 April 1993. http://www.lexisnexis.com /lnacui2api/api/version1/getDocCui?lni=3SJD-RN60-0058-X038&csi=408232,365160,144965,11432&hl=t&hv=t&hnsd=f&hns=t&hgn=t&oc=00240&perma=true.

Golumbia, David. 2003. "Computation, Gender and Human Thinking." *Differences: A Journal of Feminist Cultural Studies* 14 (2): 27–48.

Google. 2017. Diversity: Our Workplace. https://diversity.google/commitments/ - accessed 28 March 2018.

Gopinathan, S. 1989. "University Education in Singapore: The Making of a National University." In *From Dependence to Autonomy: The Development of Asian Universities*, edited by Philip G Altbach and Viswanathan Selvaratnam, 207–26.

Dordrecht: Kluwer Academic Publishers.

———. 2007. “Globalisation, the Singapore Developmental State and Education Policy: A Thesis Revisited.” *Globalisation, Societies and Education* 5 (1): 53–70.

Gopinathan, Saravanan. 2001. “Globalisation, the State and Education Policy in Singapore.” In *Education and Political Transition: Themes and Experiences in East Asia*, edited by Bray Mark and Wing On Lee, 2nd ed, 21–36. Hong Kong: Comparative Education Research Centre (CERC), The University of Hong Kong.

Gordon, Peter. 2013. “Thinking about Economic Growth: Cities, Networks, Creativity and Supply Chains for Ideas.” *Annals of Regional Science* 50: 667–84.

Graham, Elspeth, Peggy Teo, Brenda S.A. Yeoh, and Susan Levy. 2002. “Reproducing the Asian Family Across the Generations: ‘Tradition’, Gender and Expectations in Singapore.” *Asia-Pacific Population Journal* 17 (2): 61–86.

Gramsci, Antonio. 1988. *An Antonio Gramsci Reader*. Edited by David Forgacs. New York, NY: Schocken Books.

Greenhill, Pauline, and Diane Tye, eds. 2014. *Unsettling Assumptions: Tradition, Gender, Drag*. Logan, UT: Utah State University Press.

Grinshpun, Helena. 2014. “Deconstructing a Global Commodity: Coffee, Culture, and Consumption in Japan.” *Journal of Consumer Culture* 14 (3): 343–64.

Gupta, Akhil. 1995. “Blurred Boundaries : The Discourse of Corruption , the Culture of Politics , and the Imagined State.” *American Ethnologist1* 22 (2): 375–402.

Gupta, Akhil, and James Ferguson. 1992. “Beyond ‘Culture’: Space, Identity, and the Politics of Difference.” *Politics* 7: 6–23. doi:10.1525/can.1992.7.1.02a00020.

———. 1997. “Discipline and Practice: ‘The Field’ as Site, Method and Location in Anthroplogy.” In *Anthropological Locations: Boundaries and Grounds of a Field Science*, 1–46.

Guzdial, Mark. 2008. “Paving the Way for Computational Thinking.” *Communications of the ACM* 51 (8): 25–27.

———. 2016. *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. Williston, VT: Morgan & Claypool Publishers.

Hacking, Ian. 1990. *The Taming of Chance: Ideas in Context*. Cambridge: Cambridge University Press.

———. 2006. “Making Up People.” *The LRB* 29 (16): 2–7.

Haigh, Thomas. 2010. “Masculinity and the Machine Man: Gender in the History of Data Processing.” In *Gender Codes: Why Women Are Leaving Computing*, edited by Thomas J Misa, 51–71. Hoboken, NJ: John Wiley & Sons, Inc.

Halberstam, Judith. 1991. “Automating Gender : Postmodern Feminism in the Age of the Intelligent Machine.” *Feminist Studies* 17 (3): 439–60.

Handler, Richard. 1988. *Nationalism and the Politics of Culture in Quebec*. Madison, WI: University of Wisconsin Press.

Handler, Richard, and Jocelyn Joceyln Linnekin. 1984. “Tradition, Genuine or Spurious.” *The Journal of American Folklore* 97 (385): 273–90.

Hanisch, Carol. 1970. “The Personal is Political.” In *Notes from the Second Year*, edited by Shulamith Firestone, 85 – 86. New York: Radical Feminism.

———. 2006. “The Personal Is Political.” http://www.carolhanisch.org/CHwritings/PersonalIsPol.pdf.

Håpnes, Tove, and Bente Rasmussen. 1991. “The Production of Male Power in Computer Science.” In *Proceedings of the IFIP-Conference on Women, Work and Computerization*, edited by I.V. Eriksson, B.A. Kitchenham, and K.G. Tijdens, 395–406. Amsterdam: Elsevier Science Publishers.

Haraway, Donna. 1992. “The Promises of Monsters: A Regenerative Politics for Inappropriate/d Others.” In *Cultural Studies*, edited by Lawrence Grossberg and Cary Nelson Paula A. Treichler, 295–337. New York, NY: Routledge.

Haraway, Donna Jeanne. 1988. “Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective.” *Feminist Studies* 14 (3): 575–99.

———. 1991a. “A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century.” In *Simians, Cyborgs and Women: The Reinvention of Nature*, 149–82. New York, NY: Routledge.

———. 1991b. *Simians, Cyborgs, and Women*. New York, NY: Routledge.

———. 1994. “A Game of Cat’s Cradle: Science Studies, Feminist Theory, Cultural Studies.” *Configurations* 2 (1): 59–71.

———. 1997. *Modest_Witness@Second_Millenium.FemaleMan(C)_Meets_OncoMouse(TM): Feminism and Technoscience*. New York, NY: Routledge.

———. 2003. *The Companion Species Manifesto: Dogs, People, and Significant*

Otherness. Chicago: Prickly Paradigm Press.

Hardgrove, Abby, Esther Rootham, and Linda McDowell. 2015. "Possible Selves in a Precarious Labour Market: Youth, Imagined Futures, and Transitions to Work in the UK." *Geoforum* 60: 163–71.

Harding, Sandra. 1986. *The Science Question in Feminism*. Ithaca: Cornell University Press.

———. 1991. *Whose Science? Whose Knowledge?: Thinking from Women's Lives*. Ithaca, NY: Cornell University Press.

———. 1998. *Is Science Multicultural? Poscolonisms, Feminisms, and Epistemologies.* Bloomington, IN: Indiana University Press.

Harding, Susan. 1987. "Convicted by the Holy Spirit: The Rhetoric of Fundamental Baptist Conversion." *American Ethnologist* 14 (1): 167–81.

Harper, Douglas. 2002. "Talking About Pictures: A Case for Photo Elicitation." *Visual Studies* 17 (1). Routledge: 13–26.

Hartsock, Nancy C. M. 1983. "The Feminist Standpoint: Developing the Ground for a Specifically Feminist Historical Materialism." In *Discovering Reality*, edited by Sandra Harding and Merrill B. Hintikka, 283–310. Dordrecht: Springer Netherlands.

Harvey, David. 2005. *A Brief History of Neoliberalism*. Oxford, UK: Oxford University Press.

———. 2006. "Neoliberalism as Creative Destruction." *Geografiska Annaler: Series B, Human Geography* 88 (2): 145–58.

Hayes, Caroline Clarke. 2010. "Computer Science: The Incredible Shrinking Woman." In *Gender Codes: Why Women Are Leaving Computing*, edited by Thomas J Misa, 25–49. Hoboken, NJ: John Wiley & Sons, Inc.

Hayes, Patrick, and Kenneth Ford. 1995. "Turing Test Considered Harmful." In *IJCAI'95 Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, 972–77. San Francisco, CA: Morgan Kaufmann Publishers Inc. researchgate.net.

Hayles, N Katherine. 1999. *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics*. Chicago, IL: University of Chicago Press.

Helmreich, Stefan. 1998. *Silicon Second Nature: Culturing Artificial Life in a Digital World*. Berkeley, CA: University of California Press.

Heng, Geraldine, and Janadas Devan. 1995. "State Fatherhood: The Politics of Nationalism, Sexuality, and Race in Singapore." In *Bewitching Women, Pious Men*, edited by Aihwa Ong and Michael G Peletz, 195–215. Berkeley, CA: University of California Press.

Henwood, Flis. 1998. "Engineering Difference: Discourses on Gender, Sexuality and Work in a College of Technology." *Gender and Education* 10 (1): 35–49.

———. 2000. "From the Woman Question in Technology to the Technology Question in Feminism: Rethinking Gender Equality in IT Education." *European Journal of Women's Studies* 7 (2): 209–27.

Heyzer, Noeleen. 1997. "International Production and Social Change: An Analysis of the State, Employment and Trade Unions in Singapore." In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 374–95. Singapore: Times Academic Press.

Hicks, Marie. 2013. "De-Brogramming the History of Computing [Think Piece]." *IEEE Annals of the History of Computing* 35 (1): 86–88.

———. 2017. *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing.* Cambridge, MA: MIT Press.

Hicks, Robin. 2013. "Singapore journalist on self-censorship: we can't be controversial, we have to play the game." *Mumbrella Asia*, 10 July 2013. http://www.mumbrella.asia/2013/07/self-censorship-in-singapore-2, accessed 5 November 2017.

Hill, Catherine, Christianne Corbett, and Andresse St. Rose. 2010. *Why So Few? Women in Science, Technology, Engineering, and Mathematics*. *American Association of University Women*. Washington, DC: American Association of University Women.

Hill, Michael, and Kwen Fee Lian. 1995. *The Politics of Nation Building and Citizenship in Singapore*. London, UK and New York, NY: Routledge.

Ho, Karen. 2005. "Situating Global Capitalisms: A View from Wall Street Investment Banks." *Cultural Anthropology* 20 (1): 68–96.

Ho, Li-Ching. 2010. "'Don't Worry, I'm Not Going to Report You': Education for Citizenship in Singapore." *Theory & Research in Social Education* 38 (2): 217–47.

———. 2012. "Sorting Citizens: Differentiated Citizenship Education in Singapore." *Journal of Curriculum Studies* 44 (3): 403–28.

Ho, Vanessa, Sherry Sherqueshaa, and Darius Zheng. 2016. "The Forced Sterilization of

Transgender and Gender Non-Conforming People in Singapore." *LGBTQ Policy Journal* VI: 54–75.

Hobsbawm, Eric. 1992. "Introduction: Inventing Traditions." In *The Invention of Tradition*, edited by Eric Hobsbawm and Terrence Ranger. Cambridge, UK: Cambridge University Press.

Hobsbawm, Eric, and Terrence Ranger, eds. 1992. *The Invention of Tradition*. Cambridge: Cambridge University Press.

Hoe, Pei Shan. 2014. "Sex change operations dwindling in Singapore." *The Straits Times.* 28 December 2014. http://www.straitstimes.com/singapore/sex-change-operations-dwindling-in-singapore.

Holland, Dorothy C, and Margaret A Eisenhart. 1990. *Educated in Romance: Women, Achievement, and College Culture*. Chicago, IL: University of Chicago Press.

Hoofd, Ingrid. 2010. "Singapore: Bridgehead of the West or Counterforce? The S[t]imulation of Creative and Critical Thought in Singapore's Higher Education Policies." *Globalisation, Societies and Education* 8 (2): 295–305.

Hooks, Bell. 1986. "Talking Back." *Discourse* 8: 123–28.

———. 2013. "Dig Deep: Beyond Lean In." *The Feminist Wire*. http://www.thefeministwire.com/2013/10/17973/.

Hopkins, Sharon. 2001. "Camels and Needles: Computer Poetry Meets the Perl Programming Language." http://budi.insan.co.id/courses/el2001/plpaper.pdf.

Hospers, Gert-Jan. 2003. "Creative Cities: Breeding Places in the Knowledge Economy." *Knowledge, Technology, & Policy* 16 (3): 143–62.

Huang, Shirlena, and Brenda S.A. Yeoh. 2003. "The Difference Gender Makes: State Policy and Contract Migrant Workers in Singapore." *Asian and Pacific Migration Journal* 12 (1–2): 75–98.

Huat, Chua Beng. 1997. "Not Depoliticized but Ideologically Successful: The Public Housing Programme in Singapore." In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 307–27. Singapore: Times Academic Press.

———. 1998. "Racial-Singaporeans: Absense after the Hyphen." In *Southeast Asian Identities: Culture and the Politics of Representation in Indonesia, Malaysia, Singapore, and Thailand*, edited by Joel S Kahn, 28–50. New York, NY and Singapore: St. Martin's Press ; Institute of Southeast Asian Studies.

———. 2011. “Singapore as Model: Planning Innovations, Knowledge Experts.” In *Worlding Cities: Asian Experiments and the Art of Being Global*, edited by Ananya Roy and Aihwa Ong, 29–54. Malden, MA: Wiley-Blackwell.

Huff, C, and J Cooper. 1987. “Sex Bias in Educational Software: The Effect of Designers’ Stereotypes on the Software They Design.” *Journal of Applied Social Psychology* 17 (6): 519–32.

Huizinga, Johan. 1949. *Homo Ludens: A Study of the Play-Element in Culture*. London, UK: Routledge & Kegan Paul.

Huysse, Janet van. 2014. “Building a Twitter We Can Be Proud of.” *Twitter Blog*, July 23. https://blog.twitter.com/2014/building-a-twitter-we-can-be-proud-of.

iN2015 Steering Committee. 2006. “iN2015: Imagine Your World.” Singapore.

Infocomm Development Authority of Singapore (IDA). 2000. “Infocomm 21: Singapore Where the Digital Future Is.” Singapore.

Infocomm Development Authority of Singapore (IDA). 2016. “Manpower Development.” https://www.ida.gov.sg/Learning/Manpower-Development, accessed May 20, 2016.

Infocomm Media Masterplan Steering Committee (IMMSC). 2015a. “Infocomm Media 2025.” Singapore.

———. 2015b. “Infocomm Media 2025: Supplementary Information.” Singapore.

Irani, Lilly. 2015. “Hackathons and the Making of Entrepreneurial Citizenship.” *Science, Technology & Human Values* 40 (5): 1–26.

J.P. Morgan-SMU. 2016. “Singapore Faces Skills Challenges in ICT, Electronics & Engineering Manufacturing and Fintech: J.P. Morgan-SMU Study.” 1 November 2016. https://www.smu.edu.sg/sites/default/files/smu/news_room/Singapore_final.pdf, accessed August 10, 2017.

Jackson, Philip W. 1990. *Life in Classrooms*. New York, NY: Teachers College Press.

Jaswal, Balli Kaur. 2017. “The Censors’ Disappearing Vibrator.” *The New York Times*. 21 May 2017. https://www.nytimes.com/2017/05/21/opinion/singapore-media-censorship.html, accessed 5 November 2017.

Jeffrey, Annabelle. 2016. “NUS Undergrads Couldn't Track When Laundry Machines Were Used, So They Created A Bot To Help.” *Vulcan Post*. 14 September 2016.

https://vulcanpost.com/587994/nus-computing-undergrads-create-a-laundry-bot, accessed 27 July 2017.

John, Alan. 1987. “Have 3, or more if you can afford it.” *The Straits Times.* 2 March 1987. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19870302-1.2.2.aspx.

Joint Task Force on Computing Curricula. 2001. “Computing Curricula 2001 Computer Science.”

Keller, Evelyn Fox. 1985. *Reflections on Gender and Science*. New Haven, CT: Yale University Press.

Kelty, Christopher M. 2008. *Two Bits: The Cultural Significance of Free Software*. Durham, NC and London, UK: Duke University Press.

Khan, Shamus, and Colin Jerolmack. 2013. “Saying Meritocracy and Doing Privilege.” *The Sociological Quarterly* 54: 9–19.

Khan, Shamus Rahman. 2011. *Privilege: The Making of an Adolescent Elite at St. Paul’s School*. Princeton, NJ: Princeton University Press.

Kheng, Lim Gek. 1989. “Women’s Participation in the Computer Industry in Singapore.” In *Forming New Alliances: Proceedings of the Inernational Conference on Women, Work, and Computerization*, edited by K Tijdens, 309–14. Amsterdam: Elsevier Science Publishers.

———. 1990. “WIT: The Singaporean Way.” *Computer Bulletin* 2: 18–21.

Kho, Ee Moi. 2013. *Construction of Femininity in a Postcolonial State: Girls’ Education in Singapore*. Amherst, NY: Cambria Press.

Kimmel, Michael S. 2002. “Introduction: Toward a Pedagogy of the Oppressor.” In *Privilege: A Reader*, edited by Michael S. Kimmel and Abby L. Ferber, 1–10. Westview Press.

Kirkup, Gill, Anna Zalevski, Takao Maruyama, and Isma Batool. 2010. “Women and Men in Science, Engineering and Technology : The UK Statistics Guide 2010.” Bradford.

Knott, Christine. 2016. “Contentious Mobilities and Cheap(er) Labour: Temporary Foreign Workers in a New Brunswick Seafood Processing Community.” *Canadian Journal of Sociology/Cahiers Canadiens de Sociologie* 41 (3): 375–98.

Knuth, Donald E. 1974. “Computer Programming as an Art.” *Communications of the*

ACM 17 (12). ACM: 667–73.

———. 1997a. *The Art of Computer Programming, Vol 1*. 3rd ed. Boston, MA: Addison-Wesley.

———. 1997b. *The Art of Computer Programming, Vol 3*. 2nd Ed. Vol. 2. Upper Saddle River, NJ: Addison-Wesley.

———. 1998. *The Art of Computer Programming, Vol 2*. 3rd ed. Upper Saddle River, NJ: Addison-Wesley.

———. 2005. *The Art of Computer Programming, Vol 4 Part I*. Upper Saddle River, NJ: Addison-Wesley.

Koh, Aaron. 2004. "Singapore Education in 'New Times': Global/local Imperatives." *Discourse: Studies in the Cultural Politics of Education* 25 (3): 335–49.

———. 2004. "The Singapore Education System: Postcolonial Encounter of the Singaporean Kind." In *Disrupting Preconceptions: Postcolonialism and Education*, edited by Anne Hickling-Hudson, Julie Matthews, and Annette Woods, 155–72. Flaxton, Australia: Post Pressed.

———. 2014. "Doing Class Analysis in Singapore's Elite Education: Unravelling the Smokescreen of 'Meritocratic Talk.'" *Globalisation, Societies and Education* 12 (2): 196–210.

Koh, Tommy. 2008. "Differences in Asian and European Values." *Asian Mass Communication Bulletin* 29 (5): 10–11.

Kohn, Eduardo. 2013. *How Forests Think: Toward an Anthropology Beyond the Human*. Berkeley, CA: University of California Press.

Kondo, Dorinne K. 1990. *Crafting Selves: Power, Gender, and Discourses of Identity in a Japanese Workplace*. Chicago, IL: University of Chicago Press.

Kong, Chiew Seen. 1997. "The Socio-Cultural Framework of Politics." In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 86–106. Singapore: Times Academic Press.

Krishna, V.V., and Sohan Prasad Sha. 2015. "Building Science Community by Attracting Global Talents: The Case of Singapore Biopolis." *Science Technology & Society* 20 (3): 389–413.

Kuo, Eddie C.Y., and Huey-tsyh Chen. 1987. "Toward an Information Society: Changing Occupational Structure in Singapore." *Asian Survey* 27 (3): 355–70.

Kuo, Eddie C.Y., and Linda Low. 2001. "Information Economy and Changing Occupational Structure in Singapore." *The Information Society* 17 (4): 281–93.

Lagesen, Vivian A. 2008. "A Cyberfeminist Utopia?: Perceptions of Gender and Computer Science among Malaysian Women Computer Science Students and Faculty." *Science, Technology & Human Values* 33 (1): 5–27.

Lagesen, Vivian Anette. 2005. "Extreme Make-over? The Making of Gender and Computer Science." Norwegian University of Science and Technology.

Landström, Catharina. 2007. "Queering Feminist Technology Studies." *Feminist Theory* 8 (1): 7–26.

Latour, Bruno. 1987. "Centres of Calculation." In *Science in Action: How to Follow Scientists and Engineers through Society*, 215–57. Cambridge, MA: Harvard University Press.

———. 1993. *We Have Never Been Modern*. Translated by Catherine Porter. Cambridge, MA: Harvard University Press.

———. 2004. "Why Has Critique Run out of Steam? From Matters of Fact to Matters of Concern." *Critical Inquiry* 30 (2): 225–48.

———. 2005. *Reasembling the Social: An Introduction to Actor-Network Theory*. Oxford, UK and New York, NY: Oxford University Press.

Latour, Bruno, and Catherine Porter. 1996. *Aramis or the Love of Technology*. Cambridge, MA: Harvard University Press.

Lave, Jean. 2011. *Apprenticeship in Critical Ethnographic Practice*. Chicago and London: The University of Chicago Press.

Lave, Jean, and Etienne Wenger. 1991. *Situated Learning*. Cambridge, UK: Cambridge University Press.

Law, John. 2014. "Working Well with Wickedness." *CRESC Working Paper*. http://www.cresc.ac.uk/publications/working-well-with-wickedness.

Lee, Bakmi. 2015. "Successful vs unsuccessful people (Infographic)." *Tech in Asia*. 20 June 2015. https://www.techinasia.com/talk/successful-unsuccessful-people-infographic, accessed 5 November 2017.

Lee, Hsien-Loong. 2013. "Prime Minister Lee Hsien Loong's National Day Rally 2013 (Speech in English)." Singapore: Prime Minister's Office Singapore. http://www.pmo.gov.sg/mediacentre/prime-minister-lee-hsien-loongs-national-day-rally-2013-speech-english.

Lee, Hsien-Loong. 2014. “Prime Minister Lee Hsien Loong's National Day Rally 2014 (Speech in English).” Singapore: Prime Minister’s Office Singapore. http://www.pmo.gov.sg/mediacentre/prime-minister-lee-hsien-loongs-national-day-rally-2014-speech-english.

Lee, Hsien Loong. 2016. “PM Lee Hsien Loong at the Research, Innovation and Enterprise Council (RIEC) Press Conference.” Singapore: Prime Minister’s Office Singapore. http://www.pmo.gov.sg/mediacentre/pm-lee-hsien-loong-research-innovation-and-enterprise-council-riec-press-conference.

Lee, Kuan Yew. 1983. “Talent for the Future: Prepared text of the Prime Minister, Mr Lee Kuan Yew’s speech at the National Day Rally last night.” *The Straits Times.* 15 August 1983. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19830815-1.2.32.1.

Lee, Regina Marie. 2016. “‘Traditional values’ wear white campaign returning on Pink Dot weekend.” *Today Online*. 23 May 2016. http://www.todayonline.com/singapore/network-churches-revives-campaign-wear-white-pink-dot-weekend.

Leow, Joanne. 2016. “Journalism and Jiujitsu.” *Catapult*. 21 October 2016. https://catapult.co/stories/journalism-and-jiujitsu, accessed 5 November 2017.

Levinson, Bradley A., Douglas E. Foley, and Dorothy C. Holland, eds. 1996. *The Cultural Production of the Educated Person : Critical Ethnographies of Schooling and Local Practice*. Albany, NY: State University of New York Press.

Levy, Steven. 2010. *Hackers: Heroes of the Computer Revolution*. 25th Anniv. New York, NY: O’Reilly Media.

Lewis, Patricia. 2006. “The Quest for Invisibility: Female Entrepreneurs and the Masculine Norm of Entrepreneurship.” *Gender, Work and Organization* 13 (5): 453–69.

Li, Tania Murray. 1989. *Malays in Singapore: Culture, Economy and Ideology*. New York, NY and Singapore: Oxford University Press.

———. 1998a. “Constituting Capitalist Culture: The Singapore Malay Problem and Entrepreneurship Reconsidered.” In *Market Cultures: Society and Morality in the New Asian Capitalisms*, edited by Robert W Hefner, 147–72. Boulder, CO: Westview Press.

———. 1998b. “Working Separately but Eating Together: Personhood, Property, and Power in Conjugal Relations.” *American Ethnologist* 25 (4): 675–94.

———. 2007. *The Will to Improve: Governmentality, Development, and the Practice of Politics*. Durham, NC: Duke University Press.

Liew, Warren Mark. 2014. “Sex (Education) in the City: Singapore’s Sexuality Education Curriculum.” *Discourse: Studies in the Cultural Politics of Education* 35 (5). Taylor & Francis: 705–17.

Light, Ann. 2011. “HCI as Heterodoxy: Technologies of Identity and the Queering of Interaction with Computers.” *Interacting with Computers* 23 (5): 430–38.

Light, Ann, Ilda Ladeira, Jahmeilah Roberson, Nicola J Bidwell, Nimmi Rangaswamy, Nithya Sambasivan, and Shikoh Gitau. 2010. “Gender Matters: Female Perspectives in ICT4D Research.” In *4th IEEE/ACM Conference on Information and Communication Technologies and International Development (ICTD)*. London, UK.

Light, Jennifer S. 1999. “When Computers Were Women.” *Technology and Culture* 40 (3): 455–83.

Lim, Leonard and Andrea Ong. 2013. “Population White Paper Debate: Who’s Xenophobic?” *Asia One*. 21 Feb, 2013. http://news.asiaone.com/News/Latest+News/Singapore/Story/A1Story20130221-403658.html.

Lim, Yan Liang. 2015. “Revisiting Operation Coldstore.” *The Straits Times.* 13 April 2015. http://www.straitstimes.com/opinion/revisiting-operation-coldstore.

Linnekin, Jocelyn S . 1983. “Defining Tradition : Variations on the Hawaiian Identity.” *American Ethnologist* 10 (2): 241–52.

Lovegrove, Gillian, and Barbara Segal, eds. 1991. *Women into Computing: Selected Papers 1988-1990*. London, UK: Springer-Verlag.

Lynch, Michael, and Steve Woolgar, eds. 1990. *Representation in Scientific Practice*. Cambridge, MA: MIT Press.

Lyons, Lenore. 2004. *A State of Ambivalence: The Feminist Movement in Singapore*. Boston, MA: Brill.

Macdonald, Averil. 2014. “‘Not for People like Me?’ Under-Represented Groups in Science , Technology and Engineering.” Bradford, UK.

Mackenzie, Adrian. 2005. “The Performativity of Code: Software and Cultures of Circulation.” *Theory, Culture & Society* 22(1): 71-92

———. 2006. *Cutting Code: Software and Sociality*. New York, NY: Peter Lang.

MacLeod, Jay. 2009. *Ain't No Makin' It: Aspirations and Attainment in a Low-Income Neighbourhood*. Boulder, CO: Westview Press.

MacMillan, D. 2012. The Rise of the 'Brogrammer'. *Bloomberg Business*. 1 March 2012. http://www.bloomberg.com/bw/articles/2012-03-01/the-rise-of-the-brogrammer.

Macpherson, C.B. 1962. *The Political Theory of Possessive Individualism: Hobbes to Locke*. Oxford, UK: Clarendon Press.

Mahmood, Saba. 2005. *Politics of Piety: The Islamic Revival and the Feminist Subject*. Princeton, NJ: Princeton University Press.

Malaby, Thomas M. 2009. *Making Virtual Worlds: Linden Lab and Second Life*. Ithaca: Cornell University Press.

Marcus, George E. 1995. "Ethnography In/of the World System: The Emergence of Multi-Sited Ethnography." *Annual Review of Anthropology* 24: 95–117.

———. 2006. "Assemblage." *Theory, Culture & Society* 23 (2–3): 101–6. doi:10.1177/0263276406062573.

Margolis, Jane, and Allan Fisher. 2002. *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA: MIT Press.

Markus, H., and P. Nurius. 1986. "Possible Selves." *American Psychologist* 41 (9): 954–69.

Marsden, Nicola, and Maren Haag. 2016. "Stereotypes and Politics: Reflections on Personas." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, 4017–31. San Jose, CA: ACM.

Martin, Emily. 1991. "The Egg and The Sperm : How Science Has Constructed a Male-Female Roles." *Journal of Women in Culture and Society* 16 (3): 485–501.

———. 1994. *Flexible Bodies. The Role of Immunity in American Culture from the Days of Polio to the Ages of AIDS*. Boston, MA: Beacon Press.

Mauss, Marcel. 1966. *The Gift: Forms and Functions of Exchange in Archaic Societies*. Translated by Ian Cunnison. London, UK: Cohen & West LTD.

Mcpherson, Tara. 2012. "Why Are the Digital Humanities so White?: Or Thinking the Histories of Race and Computation." In *Debates in the Digital Humanities*, edited by Matthew K. Gold. University of Minnesota Press. http://dhdebates.gc.cuny.edu/debates/text/29.

McRobbie, Angela. 1978. “Working Class Girls and the Culture of Femininity.” In *Women Take Issue*, edited by Women’s Studies Group, 96–108. London: Hutchinson.

———. 2009. *The Aftermath of Feminism: Gender, Culture and Social Change*. London, UK: Sage Publications Ltd.

Mead, Margaret. 1928. *Coming of Age in Samoa: A Psychological Study of Primitive Youth for Western Civilisation*. New York, NY: William Morrow & Company.

Mellström, Ulf. 2002. “Patriarchal Machines and Masculine Embodiment.” *Science, Technology & Human Values* 27 (4): 460–78.

———. 2003. *Masculinity, Power, and Technology: A Malaysian Ethnography*. Aldershot, Hampshire, England; Burlington, VT: Ashgate.

———. 2009. “The Intersection of Gender, Race and Cultural Boundaries, or Why Is Computer Science in Malaysia Dominated by Women?” *Social Studies of Science* 39 (6): 885–907.

Merton, Robert K. 1948. “The Self-Fulfilling Prophecy.” *The Antioch Review* 8 (2): 193–210.

Ministry of Education (MOE). 2016. “MOE Framework for Sexuality Education.” https://www.moe.gov.sg/education/programmes/social-and-emotional-learning/sexuality-education/moe-framework-for-sexuality-education, accessed 5 November 2017.

Ministry of Family and Social Development (MSF). n.d. “FamilyMatters@Schools.” Singapore.

Ministry of Manpower (MOM). 2013. “Firms to Consider Singaporeans Fairly for Jobs.” http://www.mom.gov.sg/newsroom/press-releases/2013/firms-to-consider-singaporeans-fairly-for-jobs, accessed October 3, 2017.

Ministry of Manpower (MOM). 2017. “Summary Table: Unemployment.” http://stats.mom.gov.sg/Pages/Unemployment-Summary-Table.aspx, accessed August 10, 2017.

Ministry of National Development (MND). 2013. “A High Quality Living Environment for all Singaporeans: Land Use Plan to Support Singapore’s Future Population.” http://www.mnd.gov.sg/landuseplan/e-book/files/assets/basic-html/index.html, accessed August 10, 2017

Misa, Thomas J, ed. 2010. *Gender Codes: Why Women Are Leaving Computing*.

Hoboken, N.J.: IEEE Computer Society and John Wiley & Sons, Inc.

Missio, Erik. “Why kids should learn to code (and how to get them started).” *CBC News*. 9 June 2015. http://www.cbc.ca/parents/learning/view/why-kids-should-learn-to-code-and-how-to-get-them-started, 23 January 2018.

Mitchell, Timothy. 1991. “The Limits of the State : Beyond Statist Approaches and Their Critics.” *American Political Science Review* 85 (1): 77–96.

———. 2002. *Rule of Experts: Egypt, Techno-Politics, Modernity*. Berkeley: University of California Press.

Mol, Annemarie. 2002. *The Body Multiple: Ontology in Medical Practice*. Durham, NC: Duke University Press.

Montanelli Jr, Richard G, and Sandra A Mamrak. 1976. “The Status of Women and Minorities in Academic Computer Science.” *Communications of the ACM* 19 (10): 578–81.

Mosbergen, Dominique. 2015. “How One Of The World’s Richest Countries Is Limiting Basic Human Rights.” *The World Post.* 13 October 2015. http://www.huffingtonpost.com/entry/lgbt-singapore_us_561633d5e4b0e66ad4c67fe7.

Murphy, Laurie, and Lynda Thomas. 2008. “Dangers of a Fixed Mindset: Implications of Self-Theories Research for Computer Science Education.” *ACM SIGCSE Bulletin* 40 (3): 271–75.

Myers, Natasha. 2014. “Rendering Machinic Life.” In *Representation in Scientific Practice Revisited*, edited by Catelijne Coopmans, Janet Vertesi, Michael Lynch, and Steve Woolgar, 153–76. Cambridge, MA: MIT Press.

———. 2015. *Rendering Life Molecular: Models, Modelers, and Excitable Matter*. Durham, NC and London, UK: Duke University Press.

Nader, Laura. 1972. “Up the Antropologist: Perspectives Gained from Studying Up.” In *Reinventing Anthropology*, edited by Dell H. Hymes, 284–311. New York, NY: Pantheon Books.

Nafus, Dawn. 2011. “‘Patches Don’t Have Gender’: What Is Not Open in Open Source Software.” *New Media & Society* 14 (4): 669–83.

Narayan, Kirin. 1993. “How Native Is a ‘Native’ Anthropologist?” *American Anthropologist* 95 (3): 671–86.

National Computer Board (NCB). 1992. “A Vision of an Intelligent Island: The IT2000

Report." Singapore: SNP Publishers Pte Ltd.

National Population and Talent Division. 2013. "A Sustainable Population For A Dynamic Singapore: Population White Paper." Singapore: Prime Minister's Office.

Nespor, Jan. 1994. *Knowledge in Motion: Space, Time, and Curriculum in Undergraduate Physics and Management*. London, UK: Falmer Press.

———. 2000. "Anonymity and Place in Qualitative Inquiry." *Qualitative Inquiry* 6 (4): 546–69.

Nirmala, M. 1996. "Two moves to attract women science grads." *The Straits Times*. 21 April 1996. http://www.lexisnexis.com/lnacui2api/api/version1/getDocCui?lni=3SJD-NY30-0058-X1SR&csi=408232,365160,144965,11432&hl=t&hv=t&hnsd=f&hns=t&hgn=t&oc=00240&perma=true.

O'Connor, Kevin, Frederick A. Peck, and Julie Cafarella. 2015. "Struggling for Legitimacy: Trajectories of Membership and Naturalization in the Sorting Out of Engineering Students." *Mind, Culture, and Activity* 22 (2): 168–83.

O'Donnell, Casey. 2014. *Developer's Dilemma: The Secret World of Videogame Creators*. Cambridge, MA: MIT Press.

O'Regan, Gerard. 2012. *A Brief History of Computing Second Edition*. 2nd ed. London, UK: Springer.

Olds, Kris. 2007. "Global Assemblage: Singapore, Foreign Universities, and the Construction of a 'Global Education Hub.'" *World Development* 35 (6): 959–75.

Ong, Aihwa. 1999. *Flexible Citizenship: The Cultural Logics of Transnationality*. Durham, N.C.: Duke University Press.

———. 2005. "Ecologies of Expertise: Assembling Flows, Managing Citizenship." In *Global Assemblages: Technology, Politics, and Ethics as Anthropological Problems*, edited by Aihwa Ong and Stephen J Collier, 337–53. Malden, MA: Blackwell.

———. 2006. *Neoliberalism as Exception: Mutations in Citizenship and Sovereignty*. Durham, N.C.: Duke University Press.

———. 2007. "Please Stay: Pied-a-Terre Subjects in the Megacity." *Citizenship Studies* 11 (1): 83–93.

Ong, Aihwa, and Stephen J. Collier, eds. 2005. *Global Assemblages: Technology,* *Politics, and Ethics as Anthropological Problems*. Malden, MA: Blackwell Publishers Ltd.

Ong, Jason. 2016. “Singapore should kill 'kiasu' culture: NMP Kuik Shiao-Yin.” *Channel News Asia*. 5 April, 2016. http://www.channelnewsasia.com/news/singapore/singapore-should-kill/2667816.html.

Oost, Ellen van. 2003. “Materialized Gender: How Shavers Configure the Users’ Femininity and Masculinity.” In *How Users Matter: The Co-Construction of Users and Technologies*, edited by Nelly Oudshoorn and Trevor Pinch, 193–208. Cambridge, MA: MIT Press.

Ortner, Sherry B. 2016. “Dark Anthropology and Its Others: Theory since the Eighties.” *HAU: Journal of Ethnographic Theory* 6 (1): 47–73.

Oudshoorn, Nelly, and Trevor J. Pinch, eds. 2003. *How Users and Non-Users Matter: The Co-Construction of Users and Technology*. *How Users Matter - The Co-Construction of Users and Technology*. Cambridge, MA: MIT Press.

Oudshoorn, Nelly, Els Rommes, and Marcelle Stienstra. 2004. “Configuring the User as Everybody: Gender and Design Cultures in Information and Communication Technologies.” *Science, Technology, & Human Values* 29 (1): 30–63.

Papert, Seymour. 1980. *Mindstorms: Children, Computers and Powerful Ideas*. Vol. 1. New York, NY: Basic Books, Inc.

Paxson, Heather. 2004. *Making Modern Mothers: Ethics and Family Planning in Urban Greece*. Berkeley: University of California Press.

Peletz, Michael G. 2007. *Gender, Sexuality, and Body Politics in Modern Asia*. Ann Arbor, MI: Association for Asian Studies.

Pells, Rachael. 2018. “Asia’s Top University Makes Computationally Thinking.” *Times Higher Exucation*, 13 March 2018. https://www.timeshighereducation.com/news/asias-top-university-makes-computational-thinking-compulsory, accessed 18 March 2018.

Phillips, Robert. 2013. “‘We Aren’t Really That Different’: Globe-Hopping Discourse and Queer Rights in Singapore.” *Journal of Language and Sexuality* 2 (1): 122–44.

Phillips, Robert Francis. 2008. “Queering Online : Transnational Sexual Citizenship in Singapore by.” Irvine, CA: University of California, Irvine.

Pink Dot Sg. 2016. "About Pink Dot SG." http://pinkdot.sg/about-pink-dot/, accessed 5 November 2017.

Pollard, Amy. 2009. "Field of Screams: Difficulty and Ethnographic Fieldwork." *Anthropology Matters* 11: 1–24.

Powell, Abigail, Barbara Bagilhole, and Andrew Dainty. 2009. "How Women Engineers Do and Undo Gender: Consequences for Gender Equality." *Gender, Work and Organization* 16 (4): 411–28.

Priestley, Mark. 2011. *A Science of Operations: Machines, Logic and the Invention of Programming*. London: Springer.

PuruShotam, Nirmala. 1998. "Disciplining Difference: Race in Singapore." In *Southeast Asian Identities: Culture and the Politics of Representation in Indonesia, Malaysia, Singapore, and Thailand*, edited by Joel S Kahn, 51–94. New York, Singapore: St. Martin's Press and Institute of Southeast Asian Studies.

Purushothaman, Aparna. 2013. "Empowering Women Through Learning To Use The Internet - An Ethnographic Action Research Project To Address The Second Order Digital Divide." Aalborg, Denmark: UniPrint - Aalborg University.

Quek, Christopher. 2015. "The Dark Side of Tech Development in Singapore." *Tech in Asia*. 21 October 2015. https://www.techinasia.com/talk/dark-side-tech-development-singapore.

Quinn, Michelle. 2014. "Hey Apple Health, did you forget about women?" *San Jose Mercury News.* 12 August 2014. https://www.mercurynews.com/2014/12/19/quinn-hey-apple-health-did-you-forget-about-women/.

Ray, Celeste. 2005. *Transatlantic Scots*. Tuscaloosa: University of Alabama Press.

Reddy, Deepa S. 2009. "Caught! The Predicaments of Ethnography in Collaboration." In *Fieldwork Is Not What It Used To Be: Learning Anthropology's Method in a Time of Transition*, edited by James D. Faubion and George E. Marcus, 89–112. Ithaca, NY: Cornell University Press.

Reyes Jr., Vincente Chua, and S. Gopinathan. 2015. "A Critique of Knowledge-Based Economies : A Case Study of Singapore Education Stakeholders." *International Journal of Educational Reform* 24 (2): 136–59.

Riskin, Jessica. 2003. "The Defecating Duck, Or, the Ambiguous Origins of Artificial Life." *Critical Inquiry* 29: 599–633.

Ritchie, Dennis M. 1996. "The Development of the C Programming Language." In

History of Programming Languages II, edited by Thomas J. Bergin, Jr. and Richard G. Gibson, Jr., 671–98. New York, NY: ACM Press.

Rittel, Horst W. J., and Melvin M. Webber. 1973. “Dilemmas in a General Theory of Planning.” *Policy Sciences* 4: 155–69.

Robertson, Maxine, Sue Newell, Jacky Swan, Lars Mathiassen, and Gro Bjerknes. 2001. “The Issue of Gender within Computing: Reflections from the UK and Scandinavia.” *Information Systems Journal* 11 (2): 111–26.

Rodan, Garry. 2003. “Embracing Electronic Media but Suppressing Civil Society: Authoritarian Consolidation in Singapore.” *The Pacific Review* 16 (4): 503–24.

Rodman, Margaret C. 1992. “Empowering Place : Multilocality and Multivocality.” *American Anthropologist* 94 (3): 640–56.

Rommes, E., C. Bath, and S. Maass. 2012. “Methods for Intervention: Gender Analysis and Feminist Design of ICT.” *Science, Technology & Human Values* 37 (6): 653–62. http://sth.sagepub.com/cgi/doi/10.1177/0162243912450343.

Rosaldo, Renato. 1993. *Culture & Truth: The Remaking of Social Analysis*. Boston, MA: Beacon Press.

Rose, Nikolas. 2004. *Powers of Freedom: Framing Political Thought*. Cambridge, UK: Cambridge University Press.

Ross, Andrew. 2003. *No-Collar: The Humane Workplace and Its Hidden Costs*. Philadelphia, PA: Temple University Press.

Russon, Mary-Ann. 2015. “A new app that lets users' friends 'virtually walk them home at night' is exploding in popularity.” *Business Insider*. 3 September 2015. http://www.businessinsider.com/campanion-app-surging-in-popularity-2015-9, accessed 27 July 2017.

Sackman, H., W.J. Erikson, and E.E. Grant. 1968. “Exploratory Experimental Studies Comparing Online and Offline Programmmg Performance.” *Communications of the ACM* 11 (1): 3–11.

Sahlins, Marshall. 1972. *Stone Age Economics*. Chicago, IL: Aldine Atherton, Inc.

Sami. 2015. “Being a Women in Tech is Hard.” *Words: Words and Things and Stuff*. 20 September 2015. http://words.samipeachey.com.au/?p=77, accessed 5 November 2017.

Sammet, Jean E. 1981. “The Early History of COBOL.” In *History of Programming Languages*, edited by Richard L. Wexelblat, 199–243. New York, NY: Times

Academic Press.

Sandberg, Sheryl. 2013. *Lean In: Women, Work, and the Will to Lead*. Toronto, ON: Random House of Canada.

Schwab, Klaus, and Xavier Sala-i-Martin. 2014. “The Global Competitiveness Report: 2014-2015.” *World Economic Forum Reports 2014*. Geneva: World Economic Forum.

Scott, James C. 2009. *The Art of Not Being Governed: An Anarchist History of Upland Southeast Asia*. New Haven: Yale University Press.

Ser, Tan Ern. 1993. “Theorizing the Dynamics of Industrial Relations and Trade Unionism: Lessons from Singapore.” In *Understanding Singapore Society*, edited by Ong Jin Hui, Tong Chee Kiong, and Tan Ern Ser, 396–408. Singapore: Times Academic Press.

Sethuraman, V.J., and C.P. Teo. n.d. “Linear Programming Brings Marital Bliss.” *Mathematical Medley* 25 (2): 89–91.

Sharma, Aishwarya. 2015. “Answer to what can I do to be a good computer scientist?” *Quora*. 7 October 2015. https://www.quora.com/What-can-I-do-to-be-a-good-computer-scientist/answer/Aishwarya-Sharma-38, accessed 5 November 2017.

Sharma, Nandita. 2006. *Home Economics: Nationalism and the Making of ‘Migrant Workers’ in Canada*. Toronto, Buffalo: University of Toronto Press.

Shashaani, Lily. 1997. “Gender Differences in Computer Attitudes and Use among College Students.” *Journal of Educational Computing Research* 16 (1): 119–30.

Sherry. 2015. “’What doesn’t kill me makes me stronger’ – A Transgender Singaporean Woman and her Ordeal with Job Discrimination.” 8 April 2015. *Project X*. http://theprojectx.org/2015/04/08/what-doesnt-kill-me-makes-me-stronger-a-transgender-singaporean-woman-and-her-ordeal-with-job-discrimination-2/, accessed 5 November 2017.

Shklovski, Irina, Janet Vertesi, and Silvia Lindtner. 2014. “Introduction to This Special Issue on Transnational HCI.” *Human-Computer Interaction* 29 (1): 1–21.

Shore, Cris, and Susan Wright, eds. 1997. *Anthropology of Policy: Critical Perspectives on Governance and Power*. London: Routledge.

———. 1999. “Audit Culture and Anthropology: Neo-Liberalism in British Higher Education.” *The Journal of the Royal Anthropological Institute* 5 (4): 557–75.

———. 2015a. “Governing by Numbers: Audit Culture, Rankings and the New World

Order." *Social Anthropology* 23 (1): 22–28. doi:10.1111/1469-8676.12098.

———. 2015b. "Audit Culture Revisited." *Current Anthropology* 56 (3): 421-431.

Sim, Loo Lee, Seow Eng Ong, Adeesh Agarwal, Ali Parsa, and Ramin Keivani. 2003. "Singapore's Competitiveness as a Global City: Development Strategy, Institutions and Business Environment." *Cities* 20 (2): 115–27.

Sim, Walter. 2015. "Blogger Roy Ngern ordered to pay PM Lee Hsien Loong $150,000 for defamation." *The Straits Times*. 17 Dec 2015. http://www.straitstimes.com/singapore/courts-crime/blogger-roy-ngerng-ordered-to-pay-pm-lee-hsien-loong-150000-for-defamation.

Simon, Herbert A. 1996. *The Sciences of the Artificial*. 3rd ed. Cambridge, MA: MIT Press.

Singapore Department of Statistics. 2011. *Census of Population 2010 Statistical Release 2: Households and Housing*. *Households and Housing*. Singapore: Department of Statistics, Ministry of Trade & Industry. http://www.singstat.gov.sg/pubn/popn/C2010sr2/cop2010sr2.pdf%5Cnpapers2://publication/uuid/239AB639-9D74-4834-B5B5-ACD3926D92CE.

Singapore Land Authority. 2017. "Total Land Area of Singapore." *Data.gov.sg*. https://data.gov.sg/dataset/total-land-area-of-singapore, accessed 10 August 2017.

Skelton, Alan. 1997. "Studying Hidden Curricula: Developing a Perspective in the Light of Postmodern Insights." *Curriculum Studies* 5 (2): 177–93.

Smith, Dorothy E. 1974. "Women's Perspective as a Radical Critique of Sociology." *Sociological Inquiry* 44 (1): 7–13.

Spencer, Steven J., Claude M. Steele, and Diane M. Quinn. 1999. "Stereotype Threat and Women's Math Performance." *Journal of Experimental Social Psychology* 35 (1): 4–28.

Spindler, George, and Louise Spindler. 2000. *Fifty Years of Anthropology and Education, 1950-2000: A Spindler Anthology*. Edited by George Spindler. Mahwah, NJ: Lawrence Erlbaum Associates.

Spykerman, Kimberly. 2016. "Perceptions on engineering need to change for Singapore to emulate Silicon Valley: PM Lee." *Channel News Asia.* 18 February 2016. http://www.channelnewsasia.com/news/singapore/perceptions-on/2524596.html.

Stallman, Richard. 2014. "Why Software Should Not Have Owners." *GNU Operating System: Philosophy*. http://www.gnu.org/philosophy/why-free.en.html, accessed 5

November 2017.

Stambach, Amy. 2017. “Student Futures and the Politics of Possibility: An Introduction.” In *Anthropological Perspectives on Student Futures: Youth and the Politics of Possibility*, edited by Amy Stambach and Kathleen D. Hall, 1–16. New York, NY: Palgrave Macmillan.

Stambach, Amy, and Kathleen D. Hall, eds. 2017. *Anthropological Perspectives on Student Futures: Youth and the Politics of Possibility*. New York, NY: Palgrave MacMillan.

Steedly, Mary Margaret. 1999. “The State of Culture Theory in the Anthropology of Southeast Asia.” *Annual Review of Anthropology* 28 (1): 431–54.

Stonyer, Heather. 2002. “Making Engineering Students-Making Women: The Discursive Context of Engineering Education.” *International Journal of Engineering Education* 18 (4): 392–99.

Strathern, Marilyn. 1996. “Cutting the Network.” *The Journal of the Royal Anthropological Institute* 2 (3): 517–35.

Strathern, Marilyn, ed. 2000. *Audit Cultures: Anthropological Studies in Accountability, Ethics and the Academy*. London, UK and New York, NY: Routledge.

Sturman, Susan Michele. 2009. “‘Women in Computing’ as Problematic: Gender, Ethics and Identity in University Computer Science Education.” University of Toronto.

Suchman, Lucy. 1994. “Working Relations of Technology Production and Use.” *Computer Supported Cooperative Work (CSCW)* 2: 21–39.

———. 2002. “Located Accountabilities in Technology Production.” *Scandinavian Journal of Information Systems* 14 (2): 91–105.

———. 2007a. “Feminist STS and the Sciences of the Artificial.” In *The Handbook of Science and Technology Studies*, edited by E Hackett, O Amsterdamska, M Lynch, and J Wajcman, 3rd ed., 139–63. Cambridge, MA: MIT Press.

———. 2007b. *Human-Machine Reconfigurations: Plans and Situated Actions*. Cambridge; New York: Cambridge University Press.

———. 2011a. “Subject Objects.” *Feminist Theory* 12 (2): 119–45.

———. 2011b. “Anthropological Relocations and the Limits of Design.” *Annual Review of Anthropology* 40 (1): 1–18.

Sun, David. 2016. “Games at NUS camps increasingly sexualised, say students.” *The New Paper*. 26 July 2016. http://www.tnp.sg/news/singapore-news/games-nus-camps-increasingly-sexualised-say-students.

Sun, Shirley Hsiao-Li. 2012. *Population Policy and Reproduction in Singapore: Making Future Citizens*. London; New York: Routledge.

Tai, Janice. 2016. “You don't need much space to have sex: Josephine Teo on ‘no flat, no child’ belief.” *The Straits Times*. 12 October 2016. http://www.straitstimes.com/singapore/ministers-rejoinder-to-no-flat-no-child-belief.

Takhteyev, Yuri. 2012. *Coding Places: Software Practice in a South American City*. Cambridge, MA: MIT Press.

Tan, Chan-Jin (Minister for the Ministry of Social and Family Development). 2016. “Rationale behind policy of not recognising children of unwed mothers as legitimate children.” Parliamentary Questions, Ministry of Social and Family Development. https://app.msf.gov.sg/Press-Room/Rationale-behind-policy-of-not-recognising-children-of-unwed-mothers-as-legitimate-children.

Tan, Chin Nam and Yeo Khee Leng. 1981. “A National Effort.” *The Business Times*. 16 July, 1981. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/biztimes19810716-1.2.62.7.aspx.

Tan, Chris K.K. 2009. “‘But They Are Like You and Me’: Gay Civil Servants and Citizenship in a Cosmopolitanizing Singapore.” *City & Society* 21 (1): 133–54.

———. 2011. “Go Home, Gay Boy! Or, Why Do Singaporean Gay Men Prefer To ‘go Home’ and Not ‘come Out’?” *Journal of Homosexuality* 58 (6–7): 865–82.

Tan, Desney S., Mary Czerwinski, and George Robertson. 2003. “Women Go with the (Optical) Flow.” *Proceedings of the Conference on Human Factors in Computing Systems - CHI ’03*, no. 5. New York, New York, USA: ACM Press: 209. doi:10.1145/642647.642649.

Tan, Jeanette. 2013. “Kitchen Closed.” *Yahoo News*, 9 December, 2013. https://sg.news.yahoo.com/blogs/singaporescene/kitchen-closed-161623269.html, accessed 5 November 2017.

Tan, Kenneth Paul. 2003. “Sexing Up Singapore.” *International Journal of Cultural Studies* 6 (4): 403–23.

———. 2007. “In Renaissance Singapore.” In *Renaissance Singapore? Economy,* *Culture, and Politics*, edited by Kenneth Paul Tan, 1–14. Singapore: NUS Press.

———. 2008. "Meritocracy and Elitism in a Global City: Ideological Shifts in Singapore." *International Political Science Review* 29 (1): 7–27.

———. 2012. "The Ideology of Pragmatism: Neo-Liberal Globalisation and Political Authoritarianism in Singapore." *Journal of Contemporary Asia* 42 (1): 67–92.

Tan, Kenneth Paul, and Gary Lee Jack Jin. 2007. "Imagining the Gay Community in Singapore." *Critical Asian Studies* 39 (2): 179–204.

Tan, See Seng. 2016. "The 'Singapore School' of Asian Values: Down But Not Out?" *Carnegie Council for Ethics in International Affairs.* 26 January 2016. https://www.carnegiecouncil.org/publications/articles_papers_reports/762#_ftn13.

Task Force on Gender Equality at UW-CS. 2007. "Report on Gender Equality Issues at UW-CS." Waterloo, Ontario: University of Waterloo, Department of Computer Science. http://www.cs.uwaterloo.ca/~wics/task-force-report.pdf.

Taylor, Valerie E. 2002. "Women of Color in Computing." *SIGSCE* 34 (2): 22–23.

Tegos, Michael. 2016. "How Singapore's Startup Ecosystem has Grown in the Last 5 Years (Infographic)." *Tech in Asia.* 1 June, 2016. https://www.techinasia.com/singapore-startup-ecosystem-growth-infographic.

Teh, Shi Ning. 2013. "MOM Raises Monthly Salary for New Employment Pass." *AsiaOne Business.* 23 September 2013. http://business.asiaone.com/career/news/mom-raises-monthly-salary-new-employment-pass.

Teo, Youyenn. 2010. "Shaping the Singapore Family, Producing the State and Society." *Economy and Society* 39 (3): 337–59.

Tham, Irene. 2017. "Cracking code for a tech future." *The Straits Times*. 18 April 2017. http://www.straitstimes.com/singapore/cracking-code-for-a-tech-future, accessed 23 January 2018.

The President of the Republic of Singapore. 1991. "Shared Values." Singapore.

The Straits Times. 1979. "Market for computers expected to expand by 13pc a year." *The Straits Times*. 29 July 1979. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19790729-1.2.102.19.

The Straits Times. 1981a. "Conquering technofear: It's either them or us." 31 July 1981. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19810731-1.2.130.21.1.

The Straits Times. 1981b. "Going On-Line." 3 Septemebr, 1981. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19810903-1.2.67.aspx.

The Straits Times. 1981c. "High growth rate needed for skilled manpower." 9 March 1981. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19810309-1.2.59.5.

The Straits Times. 1981d. "'Some Computer Men Don't Deserve Pay They're Getting'." 2 September, 1981. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19810902-1.2.25.aspx.

The Straits Times. 1982. "Still room for women in the world of computers." *The Straits Times*. 10 May 1982. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19820510-1.2.29.

The Straits Times. 1983. "Treat Exotic Technology with Care…" 21 April 1983. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19830421-1.2.47.aspx.

The Straits Times. 1986. "Seven-prong Approach to National IT Plan." 4 December 1986. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19861204-1.2.25.1.aspx.

The Straits Times. 1988. "NUS, NTI wooing more women for engineering." *The Straits Times*. 11 April 1988. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/straitstimes19880411-1.2.30.23.

The Straits Times. 1992. "The Balance of Power." 9 September 1992. http://www.lexisnexis.com/lnacui2api/api/version1/getDocCui?lni=3SJD-S290-0058-X2FP&csi=408232,365160,144965,11432&hl=t&hv=t&hnsd=f&hns=t&hgn=t&oc=00240&perma=true

TodayOnline. 2016. "Manufacturing, information and comms sectors hit hard by unemployment." *Today Online*. 30 November 2016.

http://www.todayonline.com/singapore/manufacturing-info-communications-hit-hard-unemployment.

Today Online. 2016. “NUS disciplines 30 students over inappropriate orientation activities.” *Today Online*. 14 October, 2016. http://www.todayonline.com/singapore/nus-disciplines-30-students-over-inappropriate-orientation-activities.

Trauth, Eileen M, Susah H Nielsen, and Liisa A von Hellens. 2003. “Explaining the IT Gender Gap: Australian Stories for the New Millenium.” *Journal of Research and Practice in Information Technology* 35 (1): 7–20.

Traweek, Sharon. 1988. *Beamtimes and Lifetimes: The World of High Energy Physicists*. Cambridge, MA: Harvard University Press.

Trouillot, Michel-Rolph. 2003. “The Anthropology of the State in the Age of Globalization: Close Encounters of the Deceptive Kind.” In *Global Transformations: Anthropology and the Modern World*, 79–96. New York, NY: Palgrave Macmillan.

Tsing, Anna Lowenhaupt. 1993. *In the Realm of the Diamond Queen: Marginality in an Out-of-the-Way Place*. Princeton: Princeton University Press.

———. 2005. *Friction: An Ethnography of Global Connection*. Princeton: Princeton University Press.

Turing, A.M. 1950. “Computing Machinery and Intelligence.” *Mind* 59 (236): 433–60.

Turkle, Sherry. 1988. “Computational Reticence Why Women Fear the Intimate Machine.” In *Technology and Women’s Voices: Keeping in Touch*, edited by Cheris Kramarae, 33–49. New York: Routledge.

———. 2005. *Second Self : Computers and the Human Spirit*. 20th Anniv. Cambridge, MA: MIT Press.

Turner, Fred. 2006. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. Chicago, IL: University of Chicago Press.

Turner, Phil, and Susan Turner. 2011. “Is Stereotyping Inevitable When Designing with Personas?” *Design Studies* 32 (1). Elsevier Ltd: 30–44.

Turner, Victor W. 1969. *The Ritual Process: Structure and Anti-Structure*. Chicago, IL: Aldine Publishers Co.

———. 2008. “Betwixt and Between: The Liminal Period in Rites de Passage.” In

Magic, Witchcraft, and Religion: An Anthropological Study of the Supernatural, edited by Pamela A. Moro, James E. Myers, and Arthur C. Lehmann, 7th ed., 91–100. Boston, MA: McGraw Hill.

United Nations Development Programme (UNDP). 2015. “Human Development Report 2015 Work for Human Development.” New York, NY.

Velayutham, Selvaraj. 2016. “Races without Racism?: Everyday Race Relations in Singapore.” *Identities*. Routledge, 1–19. doi:10.1080/1070289X.2016.1200050.

Velden, Maja van der, and Christina Mortberg. 2014. “Participatory Design and Design for Values.” In *Handbook of Ethics, Values, and Technological Design*, 1–22. Dordrecht: Springer Science + Business Media.

Vigh, Henrik. 2009. “Motion Squared: A Second Look at the Concept of Social Navigation.” *Anthropological Theory* 9 (4): 419–38.

Vitores, Anna, and Adriana Gil-Juárez. 2015. “The Trouble with ‘women in Computing’: A Critical Examination of the Deployment of Research on the Gender Gap in Computer Science.” *Journal of Gender Studies*, 1–15. doi:10.1080/09589236.2015.1087309.

Viveiros de Castro, Eduardo, Morten Axel Pedersen, and Martin Holbraad. 2014. “The Politics of Ontology: Anthropological Positions.” *Fieldsights - Theorizing the Contemporary, Cultural Anthropology Online*. http://culanth.org/fieldsights/462-the-politics-of-ontology-anthropological-position.

Wagner, Roy. 2009. “Mathematical Marriages: Intercourse between Mathematics and Semiotic Choice.” *Social Studies of Science* 39 (2): 289–308.

Wajcman, Judy. 1991. *Feminism Confronts Technology*. University Park, PA: Pennsylvania State University Press.

———. 2000. “Reflections on Gender and Technology Studies.” *Social Studies of Science* 30 (3): 447–64.

Waldby, Catherine. 2009. “Singapore Biopolis: Bare Life in the City-State.” *East Asian Science, Technology and Society: An International Journal* 3 (2–3): 367–83.

Weis, Lois. 1985. *Between Two Worlds: Black Students in an Urban Community College*. Boston: Routledge, Kegan and Paul.

Weisfeld, Matt. 2013. “Becoming a Programming Rock Star: 5 Traits that Make a Great Programmer” *InformIT.* 25 September 2013.

http://www.informit.com/articles/article.aspx?p=2135212, accessed 5 November 2017.

Werbner, Pnina. 2008. “Introduction: Towards a New Cosmopolitan Anthropology.” In *Anthropology and the New Cosmopolitanism*, edited by Pnina Werbner, 1–32. Oxford and New York: Berg.

Weyl, Hermann. 1949. “Almost Periodic Invariant Vector Sets in a Metric Vector Space.” *American Journal of Mathematics* 71 (1): 178–205.

WikiLeaks. 2009. “Journalists Frustrated by Press Controls.” *Wikileaks*. 16 January 2009. https://www.wikileaks.org/plusd/cables/09SINGAPORE61_a.html, accessed 5 November 2017.

Williams, Maxine. 2014. “Building a More Diverse Facebook.” *Facebook Newsroom*, June 25. http://newsroom.fb.com/news/2014/06/building-a-more-diverse-facebook/.

Williams, Raymond. 1965. *The Long Revolution*. Harmondsworth, UK: Penguin Books.

———. 1977. *Marxism and Literature*. Oxford, UK: Oxford University Press.

———. 1983. *Keywords: A Vocabulary of Culture and Society*. New York, NY: Oxford University Press. doi:10.1080/09502368908582062.

Willis, Paul. 1981. *Learning to Labor: How Working Class Kids Get Working Class Jobs*. New York, NY: Columbia University Press.

Wing, Jeannette M. 2006. “Computational Thinking.” *Communications of the ACM* 49 (3): 33–35.

Winner, Langdon. 1980. “Do Artifacts Have Politics ?” *Modern Technology: Problem or Opportunity (Winter, 1980)* 109 (1): 121–36.

Wolf, Eric R. 1990. “Distinguished Lecture : Facing Power - Old Insights, New Questions.” *American Anthropologist* 92 (3): 586–96.

———. 1997. *Europe and the People Without History*. Berkeley,CA: University of California Press.

———. 1999. *Envisioning Power: Ideologies of Dominance and Crisis*. Berkeley, CA: University of California Press.

Wong, Caroline Y.L., Carla C.J.M. Millar, and Chong Ju Choi. 2006. “Singapore in Transition: From Technology to Culture Hub.” *Journal of Knowledge Management* 10 (5): 79–91.

Wong, Kai Wen, and Tim Bunnell. 2006. "'New Economy' Discourse and Spaces in Singapore: A Case Study of One-North." *Environment and Planning A* 38 (1): 69–83.

Wong, Seng Hon. 1992. "Exploiting Information Technology: A Case Study of Singapore." *World Development* 20 (12): 1817–28.

xkcd. n.d. "Protocol." http://xkcd.com/1323/, accessed 5 November 2017.

Yanagisako, Sylvia, and Carol Delaney. 1995. "Naturalizing Power." In *Naturalizing Power: Essays in Feminist Cultural Analysis*, edited by Sylvia Junko Yanagisako and Carol Delaney, 1–24. New York, NY and London, UK: Routledge.

Yang, Peidong. 2014a. "'Authenticity' and 'Foreign Talent' in Singapore: The Relative and Negative Logic of National Identity." *SOJOURN: Journal of Social Issues in Southeast Asia* 29 (2): 408–37.

———. 2014b. "Privilege, Prejudice, Predicament: 'PRC Scholars' in Singapore — An Overview." *Frontiers of Education in China* 9 (3): 350–76.

Yao, Souchou. 2006. *Singapore: The State and the Culture of Excess*. London, UK: Routledge.

Ye, Rebecca, and Erik Nylander. 2015. "The Transnational Track: State Sponsorship and Singapore's Oxbridge Elite." *British Journal of Sociology of Education* 36 (1): 11–33.

Yencken, David. 1988. "The Creative City." *Meanjin* 47 (4): 18–19.

Yeoh, Brenda S A. 2006. "Bifurcated Labour: The Unequal Incorporation of Transmigrants in Singapore." *Tijdschrift Voor Economische En Sociale Geografie* 97 (1): 26–37.

Yeoh, Brenda SA, Shirlena Huang, and Joaquin Gonzalez III. 1999. "Migrant Female Domestic Workers: Debating the Economic, Social and Political Impacts in Singapore." *The International Migration Review* 33 (1): 114–36.

Yeoh, Brenda S.A., Shirlena Huang, and Katie Willis. 2000. "Global Cities, Transnational Flows and Gender Dimensions: The View from Singapore." *Tijdschrift Voor Economische En Sociale Geografie* 91 (2): 147–58.

Yeoh, Brenda S.A., and Weiqiang Lin. 2013. "Chinese Migration to Singapore : Discourses and Discontents in a Globalizing Nation-State." *Asian Pacific Migration Journal* 22 (1): 31–54.

Yon, Daniel A. 2000a. *Elusive Culture: Schooling, Race and Identity in Global Times*.

Albany: SUNY Press.

———. 2000b. “Urban Portraits of Identity: On the Problem of Knowing Culture and Identity in Intercultural Studies.” *Journal of Inercultural Studies* 21 (2): 143–57.

———. 2003. “Highlights and Overview of the History of Educational Ethnography.” *Annual Review of Anthropology* 32: 411–29.

Young, Michael. 1994. “Meritocracy Revisited.” *Society* 31 (6): 87–89.

———. 2008. *The Rise of Meritocracy*. New Brunswick, NJ: Transaction Publishers.

Yue, Audrey. 2007. “Hawking In The Creative City.” *Feminist Media Studies* 7 (4): 365–80.

Zhou, Mei. 1982. “The Software Side of Development.” *The Business Times*. 6 August 1982. http://eresources.nlb.gov.sg/newspapers/Digitised/Article/biztimes19820806-1.2.62.4.aspx

Appendix A : Interview Schedules

Student Interview Schedule

1) Could you tell me a little bit about yourself? Age? Where are you from? Where did you grow up? Where is your family from?
    a. Could you tell me about your school experiences before starting university?
    b. Do you have a partner? Are they students? What do they study?
    c. Could you tell me about your current living situation?
2) Could you tell me a story of how you first started to use and program computers?
3) How and why did you come to enroll in computer science?
    a. What do you hope to do with a computer science degree?
    b. What role did your family play in pursuing computer science?
4) Is it expensive to study computer science?
5) Do you feel you have support in your studies (by the department, friends, family)? Why or why not?
6) Could you tell me about your experiences studying computer science so far?
    a. What courses are you taking?
    b. Could you tell me a little bit about what you're learning in them?
    c. Could you tell me about a particularly powerful/positive learning experience relating to computing? What happened? Why did it work for you?
    d. Tell me about the most negative computing experience you've had. Why was it as bad as it was?
    e. Grad student: Could you tell me a bit about your research?
    f. What tools do you use for programming?
    g. If you could change anything about your current studies and program, what would it be?
7) Have you studied any other subjects? If so, what do you like or dislike about computer science in comparison?
8) Tell me the story of your day yesterday.
    a. In what ways was this a typical day? An atypical day?
9) Often people are doing a lot of stuff besides studying for their classes. What are the other things -- both fun and responsibilities -- that take major chunks of your time? Why do you choose to spend your time on these things?
    a. Have you ever participated in groups such as the Students' Computing Club or the Hackers group? Why or why not?
10) International Students: How do your studies in Singapore compare with your experiences in ____?
    a. How and why did you decide to study in Singapore?
    b. Do you think Singapore is a good place to study/practice computing? Why?
    c. Have you encountered any issues relating to being a foreign student while studying or at [Temasek]? In Singapore, more generally?
11) What do you like and dislike about computer science?
12) Grad Students/TAs: What is your role as a TA?

a. What practices or values do you emphasize to students?
b. What do you like and dislike about being a TA?
c. Do you think anyone can do computer science?
d. How does your TA work relate to the lectures and the organization of the course by the instructor?
e. If you were responsible for grading: What did you look for when grading programming assignments? What kind of feedback did you give to students?

13) What makes someone a good computer scientist?
    a. Can anyone become a computer scientist?
14) Have you ever felt like you didn't belong in this discipline?
    a. Similarly, have you ever felt like you definitely did belong?
15) Do you think there are different types or groups of students in computer science or at Temasek?
    a. Who do you think the different types are?
    b. Which group do you belong to?
16) What role do you think computer science plays in Singapore society? The world?
    a. What role do you think computer science should play in society?
17) Have you or anyone you know encountered any issues relating to gender while studying computer science?
18) What gender do you consider yourself?
    a. What does your gender mean to you? For example, does it entail certain ways of dressing, particular behaviours, or certain responsibilities?
19) Do you think gender is relevant to computing? Why or why not?
    a. Do you see different genders as doing computer science differently?
    b. Do you think your gender has shaped your experiences studying computer science in any way?
    c. Why do you think there are often fewer women in computer science?
20) Do you pay attention to government policies? To the news? In what ways are these significant to you?
21) What do you plan to do when you finish your degree?
22) How do you envision your life in the future? In 5 years? 10 years? 20 years?
    a. What kind of career/job do you want to pursue?
    b. What other goals for your life do you have?
    c. Do you plan to stay in Singapore? Why or why not?
    d. Do you get along with your family?
23) If you could change anything (about computer science, studying at [Temasek], university policies, government policies), what would it be? Why?
24) Photo elicitation – see Chapter 2.

Professor Interview Schedule

1) Could you tell me a little bit about yourself? Age? Where are you from? Where did you grow up? Where is your family from?

a. Do you have a family?
b. What does your partner/spouse do?

2) Could you tell me the story of how you first started to use and program computers?
    a. Did you feel you had support in your pursuit (by family, teachers, friends, etc.)?
    b. Is a computer science professor what you envisioned for yourself when you were little? Were there other careers/lives you thought you would pursue at the time?
3) Could you tell me about your early experiences learning computer science?
    a. How does education in computer science at Temasek compare or contrast to your experiences?
4) What do you like and dislike about computer science?
    a. About programming?
5) To what extent and in what ways do you think learning computer science has led you to think in a particular way?
    a. To what extent and in what ways do you think learning computer science has led you look at the world in a particular way? Could you elaborate?
6) How long have you been working at [Temask]? How did you come to work here?
    a. What led you to want to teach/research computer science?
    b. Why teach/research at [Temasek]? In Singapore?
    c. How has computing at [Temasek] changed since you've worked here?
    d. Do you think Singapore is a good place to study/teach computing? Why?
    e. Have you ever felt like you didn't belong in this discipline?
        i. Similarly, have you ever felt like you definitely did belong?
    f. How do you think being a foreign professor has shaped your experience at [Temasek] or in Singapore more generally?
7) Could you tell me about how you decide what material to teach in your courses?
    a. What resources do you use, if any, in designing course material, examples, assignments, etc.?
    b. How did you develop your teaching style?
        i. How does your teaching style differ between different courses?
    c. What values, if any, do you try to emphasize to students in your teaching?
    d. In what ways do you incorporate university or government policies into your teaching, if at all?
        i. Could you give me some examples?
        ii. Could you tell me about the significance of incorporating these policies to you?
8) What makes someone a good computer science professor?
9) What particular challenges do you face in your job?
10) What makes someone a good computer scientist?
    a. Can anyone become a computer scientist?
    b. Could you tell me why you think some people would not want to pursue computer science?
    c. Can someone be a computer scientist without having an interest or passion?
    d. Can you describe a student that is the ideal computer science student?

11) What gender do you consider yourself?
    a. What does your gender mean to you? For example, does it entail certain ways of dressing, particular behaviours, or certain responsibilities?
12) In what way is gender relevant to teaching or learning computer science, if at all?
    a. Is it something you attend to in designing or teaching computer science courses?
13) Do you see different genders as doing computer science differently?
14) How has your gender shaped your experiences doing computer science?
15) What role do you think computer science plays in Singapore society? The world?
    a. What role do you think computer science should play in society?
    b. How do you see yourself fitting in with that?
16) Do you pay attention to government policies? To the news? In what ways do these shape your goals and plans?
17) How do you envision your life in the future? In 5 years? 10 years?
    a. Do you plan to stay in Singapore? Why or Why not?
18) If you could change anything (about computer science, studying at [Temasek], university policies, government policies), what would it be? Why?