Concordia University St. Paul

DigitalCommons@CSP

CUP Ed.D. Dissertations

Concordia University Portland Graduate Research

5-1-2019

A Case Study on Elementary Teachers' Experiences Teaching Computer Science

Steve T. Kong

Concordia University - Portland, skong@rusdlearns.net

Recommended Citation

Kong, S. T. (2019). *A Case Study on Elementary Teachers' Experiences Teaching Computer Science* (Thesis, Concordia University, St. Paul). Retrieved from https://digitalcommons.csp.edu/cup_commons_grad_edd/316

5-2019

A Case Study on Elementary Teachers' Experiences Teaching Computer Science

Steve T. Kong
Concordia University - Portland

Concordia University–Portland

College of Education

Doctorate of Education Program

WE, THE UNDERSIGNED MEMBERS OF THE DISSERTATION COMMITTEE

CERTIFY THAT WE HAVE READ AND APPROVE THE DISSERTATION OF

Steve Tae Ho Kong

CANDIDATE FOR THE DEGREE OF DOCTOR OF EDUCATION

Edward Kim, Ph.D., Faculty Chair Dissertation Committee

Tom Cavanagh, Ph.D., Content Specialist

Heather Miller, Ph D., Content Reader

A Case Study on Elementary Teachers’ Experiences Teaching Computer Science

Steve Tae Ho Kong

Concordia University–Portland

College of Education

Dissertation submitted to the Faculty of the College of Education

in partial fulfillment of the requirements for the degree of

Doctor of Education in

Instructional Leadership

Edward Kim, Ph.D., Faculty Chair Dissertation Committee

Tom Cavanagh, Ph.D., Content Specialist

Heather Miller, Ph D., Content Reader

Concordia University–Portland

2019

Abstract

Computer science is an academic discipline that provides a new foundational skill for all students. The discipline helps students develop critical thinking skills and teaches students how to move beyond merely using technology to becoming creators. The importance of computer science has been recognized across the country as states are working to train staff, adopt standards, and create implementation plans. As the demand for computer science learning opportunities increase, elementary teachers need the necessary supports to help ensure equitable access for all students. The purpose of this qualitative case study was to gain an understanding elementary teachers' experiences teaching computer science. Two research questions guided this study: What barriers do elementary teachers experience in teaching computer science to students? What strategies do elementary teachers develop in their experience teaching computer science to students? The data collection instruments were primary semistructured interviews, secondary semistructured interviews, and documents. The inductive analysis model was used to analyze the collected data from the semistructured interviews. The typological analysis model was used to analyze the documents. The key findings of this study revealed that elementary teachers experience internal and external barriers in teaching computer science. Participants understood the importance of teaching computer science to their students, prompting them to persevere and develop new strategies in their pedagogical approaches. Interviewees also developed strategies to limit the impact of the barriers they experienced teaching elementary computer science.

Keywords: computer science, elementary teachers, barriers, strategies

Dedication

This dissertation is dedicated to all of the educators and administrators who decided it was important to give students an opportunity to learn something new. To the larger computer science community moveForward();

Acknowledgements

I thank my family and friends who have seen me through this journey. My wife Jenny has taken on more than I could imagine to raise our children as I focused on completing my study. Her constant love and support helped me see the end. I thank my three children, Kylie, Ellis, and Isaiah who gave me moments to forget the work that needed to be completed and just enjoy time with them. My parents and grandmother who gave everything so that my brother and I learned the value of hard work and humility. Thank you for teaching me to complete everything to the end. It was with all of their collective support, continued love, and sacrifice that I was able to finish this dissertation. This accomplishment belongs to them.

I also want to acknowledge my faculty chair advisor Dr. Kim and committee members Dr. Miller and Dr. Cavanagh who provided feedback and guidance throughout this endeavor. Lastly, much is owed to Jenny Martel my editor. Thank you for going beyond APA style guides and helping me submit the best possible work.

Thank you to the leaders who have supported me through the completion of this dissertation. Renee, Dr. Yee, Dr. Perez, Dunlap, Dr. Sanchez, Caryn, my ILE team, and Lynn, who all gave me opportunities, support, and guidance throughout this journey.

Table of Contents

List of Tables

Chapter 1: Introduction

In the United States and globally there is rapid change toward a more digital economy and workforce. The U.S. Bureau of Labor Statistics 2015 growth forecast ranked computer and mathematical occupations fourth of all careers, with an expected 13.1% increase (Hogan & Roberts, 2015). This is double the average for all others, adding more than half a million new jobs in these fields by 2024. Concurrently, increases in automation are expected to result in up to a 6.7% loss in existing jobs, especially in manufacturing (Hogan & Roberts, 2015). Computing knowledge is becoming an essential skill across all industries and teachers must provide understanding and experience in computer science (Ryoo, Goode, & Margolis, 2015). Consequently, to prepare students for college and careers in a rapidly changing digital world, schools must work to provide computer science and computational thinking as a part of instruction.

Technology can dramatically impact education and the learning experience when teachers create fundamentally different teaching and learning environments through its use. However, teachers face a challenge in implementing instructional technology in meaningful ways to impact student learning to reflect their current and future worlds (Giannakos, Pappas, Jaccheri, & Sampson, 2017; Peters & Araya, 2011). In preparation for a digitally connected world, schools are teaching digital literacy skills to help build awareness and develop competency with technology. However, digital literacy is distinct from computer science as the latter provides a creative means for self-expression and creative problem-solving opportunities. If digital literacy is about how students read technology, computer science is how students write using technology, using computer science to create and produce programs, websites, and software. Unfortunately, many students are not receiving the opportunity to learn this essential computing literacy.

A large percentage of current high school students are not prepared to take the AP computer science courses as they have not received prior learning experiences. A comprehensive review of the 2016 AP Computer Science A Test results showed that only 0.5% of 1.9 million California high school students took the exam (College Board, 2016). Furthermore, discrepancies exist among student subgroups, as African American students comprised only 1% of the test-takers, and Hispanic students only comprised 15% of test-takers. Gender issues are also prevalent as only 27% of AP Computer Science A test-takers were female students (College Board, 2016). To address these results, students require access to computer science education before they enter high school. This necessitates computer science learning experiences to happen at elementary levels with trained staff. However, to support elementary teachers to meaningfully integrate computer science in their instructional day, they must receive the necessary supports to build their efficacy (Bender, Schaper, Caspersen, Margaritis, & Hubwieser, 2016). Providing appropriate scaffolding requires an understanding of the barriers elementary teachers experience teaching computer science.

Building a pipeline of students through a K–12 pathway for computer science is critical to improving access, exposure, and representation. Learning experiences in elementary grades are particularly important, as students who build confidence and competency in science, math, and technology are far more likely to build up confidence and persevere in future coursework (Krauss & Prottsman, 2017). Young students have positive attitudes about computer science but possess limited knowledge and experience with the content (Krauss & Prottsman, 2017). Early and consistent exposure is especially important to boost girls' confidence and interest in computer science as their affinity towards the subject is a critical factor to pursuing future careers (Hur, Andrzejewski, & Marghitu, 2017). Thus, teachers in elementary grades face a challenging imperative to teach computer science to younger students.

In addition to having elementary teachers who are trained to teach computer science, students need early access to curriculum through concepts and propositions to gain a coherent view of the subject they are trying to learn (Winch, 2013). A structured approach to teach computer science content through meaningful progression requires teachers to have some understanding of what and how to teach. If teachers lack the required knowledge to manage and provide access to computer science, students are unable to develop their own development and understanding of computer science concepts. Consequently, all students are not getting access to learning opportunities in computer science and are unable to get the necessary exposure to build their confidence and competency (DeJarnette, 2012; Srisupawong, Koul, Neanchaleay, Murphy, & Francois, 2018). Without an understanding of teachers' experiences, appropriate supports are not available and groups of students fail to receive equitable learning opportunities to learn and participate in computer science.

Background, Context, History, and Conceptual Framework of the Problem

Background, Context, and History

Computing technologies have changed significantly in the last few decades. Computer science was developed years before the release of the first programmable computer. However, early computer scientists expected definitive limits on computing power and the subsequent impact of technology via methods of automation (Krauss & Prottsman, 2017). Then, automation within a scope of possibilities formalized the idea of algorithms. Since then, computer science has evolved with programming being an element of the broader content. However, early computers were only able to follow exact and precise commands of the user program. Papert (1980) introduced students to computer science as a necessary avenue for learning how and why computing technologies worked. As such, knowledge of computer science, the connection between inputs and outputs, and programming knowledge was essential for those interested in learning how to use a computer. Papert compared computers to mud pies as things to think with, tools used to help express ideas. However, the shift in computer science occurred as technology quickly evolved with user-friendly interfaces, hiding the activity happening in the device (Krauss & Prottsman, 2017). Innovative hardware such as mobile computing devices led to a shift in the way computer science was taught as the new focus centered around digital literacy skills and using technology as consumers rather than creators (Margolis, Estrella, Goode, Holme, & Nao, 2012). Consequently, students were unable to consistently access high quality, rigorous computer science education and curriculum to develop computational thinking practices and habits.

Even so, momentum in computer science is building in some of the largest school districts across the country. Los Angeles Unified, Miami-Dade, Chicago, and New York City school districts have committed to the mission of every child learning computer science every year in their schools (Krauss & Prottsman, 2017). The shift in urgency and need is apparent at legislative levels as well with the recent "Computer Science for All" initiative launched by the White House and "CSforCA" in California (Ladner & Israel, 2016). These campaigns are focused on empowering students with computer science skills to thrive and succeed in a digital world. Providing computer science learning opportunities for students reflects a shift in expectations of computer science knowledge as a necessary skill set for jobs of the future.

Some districts have also started to emphasize teaching and learning computational thinking concepts for all students (Marcelino, Pessoa, Vieira, Salvador, & Mendes, 2018). Computational thinking is a growing field in education and requires teachers to possess new pedagogical strategies and develop an understanding of computer science concepts. The conceptual foundation of computational thinking centers around solving problems effectively and efficiently with the assistance of computers empowering students to develop and apply facets of decomposition, abstraction, and algorithm design (Shute, Sun, & Asbell-Clarke, 2017). This way of thinking is not limited to computer science but can be applied across content areas and reusable in different contexts.

To prepare students in computer science, teachers must have the knowledge and ability to teach relevant content (Margolis et al., 2012). However, perceptions among teachers and administrators is that teaching and learning computer science is daunting and technically challenging (Yadav, Gretter, Hambrusch, & Sands, 2017). Specifically, schools have a lack of elementary teachers with the knowledge and ability to provide computer science instruction (Ozturk, Dooley, & Welch, 2018).

Limited opportunity for exposure in lower grades exacerbates issues of equity and access. Computer science as a fundamental literacy requires that all students have opportunities to learn (Ryoo et al., 2015). Computer science degrees are not essential for all students but changing demands in the labor market requires some knowledge of computing as the use of software has exponentially grown with the rise of mobile operating systems (Carter, 2014). Unfortunately, inequities exist as only a quarter of U.S. high schools offer computer science as an elective. Additionally, the majority of students are predominantly White, economically middle class or higher, and male (Margolis et al., 2012).

The lack of teachers teaching computer science in primary grades causing a lack of access at elementary levels is problematic **(**Ozturk et al., 2018; Sentance & Csizmadia, 2017)**.** Furthermore, districts are unable to provide the best possible resources as there is a lack of understanding on elementary teachers' experiences with computer science. Without learning experiences in elementary levels, students are not prepared to enroll and succeed in computer science classes in later years. This inequity leads to a decline in enrollment at secondary levels (Krauss & Prottsman, 2017). Asking students to wait until high school to take a computer science course has historically been ineffective and is one of the primary reasons there are large gaps. Thus, the importance of providing access early on in elementary levels to all students is essential to building a diverse workforce.

The lack of a diverse workforce is striking in technology companies. In the video game industry, 44% of gamers are female, but 97% of programmers are male (Burrows, 2013). Unequal representation exists in large technology companies. For instance, 94% of African American students actively use social media; however, only 1.8% of African American employees in social media organizations are African American (Harkinson, 2015). The necessity for higher representation from African Americans, Hispanics, and women in technology companies indicates a need to provide all students an opportunity to learn computer science in their classrooms. Without a curricular change and well-trained elementary teachers who provide computer science education access at an early level to all students across gender and ethnic backgrounds, these students will be unfairly disadvantaged, professionally and economically.

Conceptual Framework

Constructivism is a theory of learning wherein learners actively construct knowledge through interactions (Creswell, 2013; Richardson, 1997; Stake, 1995; Wild, 2015). Learners have unique constructions based on their preexisting knowledge, learning style, and the way they interpret the learning experience. According to Piaget (1952), constructivism allows each learner to combine sensory data with existing skills to develop new knowledge. Sensory data through active learning is essential for learners to construct new knowledge. This process of learning is a logical component of teaching and learning computer science. Teachers and students use hardware and software to connect the two resources and build new creations, expanding their learning in the process.

Teachers who are able to apply constructivist theory combined with technology integration can design and promote highly engaging instructional models (Judson, 2006; Ogunkola, 2008). In this approach, computing technology in the classroom provides new opportunities for teaching and learning as these resources can provide student-centered learning experiences. However, teachers' attitudes and beliefs about technology may limit levels of integration and accessibility for students (Wild, 2015). Teachers who are reluctant to integrate technology in their classrooms can be gatekeepers to access, particularly for students who lack computing resources outside of school. However, meaningful technology integration can occur within the context of a constructivist framework. Teachers' use of technology, learning, and abilities to adapt and teach with technology can make the education process more effective (Pourhosein-Gilakjani, Mei-Leong, & Nizam-Ismail, 2013).

While computer science education also requires teachers and students to construct knowledge, the current state of computer science curricula, especially at the primary level, is confronted with a special challenge. Students and teachers may lack a robust model on what computer science is and how to appropriately teach the content without a foundational understanding of computing concepts (Margolis et al., 2012). Therefore, teachers have a tremendous responsibility to engage in individual reflection while providing opportunities for active learning and social interaction. Furthermore, to effectively teach California's adopted computer science state standards, teachers must develop their own knowledge to construct a viable model of computer science to support student learning. Consequently, this study will focus on gaining an understanding of elementary teachers' experiences teaching computer science in a large urban district.

Statement of the Problem

There is a lack of understanding of the experiences of elementary teachers who teach computer science to provide them with necessary supports. States are beginning to prioritize computer science education with statewide standards and initiatives to increase computer science exposure in schools (Harmon, 2018). As such, schools and districts are providing professional development for teachers to gain exposure to computer science and teach the content in their classrooms. However, requiring elementary teachers to teach computer science is challenging and providing the most effective professional development requires understanding their experiences (Sentance & Csizmadia, 2017; Srikoom, Hanuscin, & Faikhamta, 2017)**.**

Computer science education at elementary levels is a relatively new subject as most technology lessons emphasize digital literacy skills (Krauss & Prottsman, 2017; Sentance & Csizmadia, 2017). Research in educators' experiences at elementary levels is limited. This problem impacts elementary teachers because districts must have an understanding of their experiences to provide relevant and meaningful supports.

Additional factors contributing to this problem, may include identifying the barriers elementary teachers face teaching computer science and strategies they develop to better teach computer science. Such information may help guide implementation plans, relevant curricula, and possible supports teachers feel they need to successfully teach computer science. Consequently, this qualitative case study will contribute to the body of knowledge needed to address the problem by exploring elementary teachers' experiences teaching computer science in a large urban district.

Addressing the problem may highlight possible supports that can be implemented for more elementary teachers to teach computer science. For administrators and districts to provide the necessary support systems, they must develop a clear understanding of elementary teachers' experiences trying to teach computer science (Ryoo et al., 2015). Identifying themes within their experiences may provide insight into appropriate supports teachers feel they need to feel successful and prepared. The problem is significant because without elementary teachers implementing computer science, many students may not have an opportunity to learn the critical skills they need for the future (Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013).

Purpose of the Study

The purpose of this study was to gain an understanding of the experiences of elementary teachers in a large urban district who teach computer science. Elementary teachers are subject to high pedagogical expectations as they need to teach and have content knowledge of all subject areas. Unlike their peers at secondary levels, content expertise spans a diverse range of subjects while also being specific to the grade level they teach in a year. Consequently, teachers can apply their own experiences as students with most content areas but are unable to use a similar strategy with computer science. In this unique context, educators must balance the role of both teacher and learner as they are learning computer science concepts from student perspectives, while simultaneously applying pedagogical strategies to support student learning.

As digital technologies become more common in aspects of daily life, students must understand how and why technology works so they can analyze and solve new problems (Ni & Guzdial, 2012). Researchers have shown students develop confidence in new subjects when they are engaged in elementary to middle school ages (Krauss & Prottsman, 2017; Ladner & Israel, 2016). Introducing computer science in elementary grades is essential because computing hardware and software are ubiquitous in today's digitally connected world (Krauss & Prottsman, 2017; Ozturk et al., 2018). However, researchers have suggested elementary students learn and interact with technology from a consumer perspective, learning how to use technology rather than understanding how to make technology work (Israel, Pearson, Tapia, Wherfel, & Reese, 2015). Consequently, the divide is exacerbated by the challenge of introducing computer science at elementary levels. To address the gap and develop students' skills in computer science, elementary teachers are essential partners to expanding access and learning opportunities.

Most of the research on computer science education emphasizes secondary teachers' needs and experiences. Research that leads to better understanding of elementary teachers' experiences teaching computer science may provide ways to develop their efficacy in their efforts. Better understanding of their experiences could help provide direction and guidance on how schools may implement high-quality computer science learning opportunities for all students. I examined the experiences of elementary teachers from a school site in a large urban district. I utilized a qualitative case study to collect data from participants and document their experiences (Creswell, 2013; Stake 1995).

Nature of the Study

A qualitative case study design allowed me to use the best vehicle to focus on understanding elementary teachers' experiences teaching computer science in a large urban district. This case study required in-depth examination of participants' experiences as they taught computer science. The design allowed me to provide layers of analysis in a study through multiple sources of data collection and reflection (Creswell, 2013). To accomplish this, I engaged in two phases of interviews as well as a document review. The design allowed me to collect data to provide rich, thick description that addresses the research questions.

The population for this study was from Inland Unified School District, a pseudonym for the large urban school district in California where this study took place. There are approximately 50 schools in the district, 30 of them being K–6 elementary schools. In the past year, 177 elementary teachers received exposure to computer science curricula through professional development or conferences. I purposefully selected 10 teachers to participate in this study.

Teacher participants were recruited from Abraham Lincoln Elementary School, a pseudonym for the elementary school where this study occurred.

Case studies offer diverse approaches to instrumentation and data collection (Creswell, 2013; Stake, 1995; Yin, 2014). Data for this research were collected from elementary teachers at Abraham Lincoln Elementary school who taught computer science during this study. Participant teachers were invited to contribute documents related to computer science education, which included lesson plans or other resources they felt were relevant to this study. Participant teachers were also invited to engage in two phases of interviews. Each interview was recorded and transcribed for data analysis.

I utilized an inductive approach to data analysis. The outcome of inductive analysis is the development of categories that summarize the data and form key themes (Schutt, 2018; Thomas, 2006). I followed a process to conceptualize the data through coding, combined the codes into broader categories or subthemes, and ultimately established the key findings.

Research Questions

This study answered two research questions:

RQ1: What barriers do elementary teachers experience in teaching computer science to students?

RQ2: What strategies do elementary teachers develop in their experience teaching computer science to students?

Rationale, Relevance, and Significance of the Study

Filling the future's needs for effective, productive workers requires the development of creativity, innovation, and knowledge of computing among students (Guzdial, 2014; Yadav et al., 2017). Specifically, elementary teachers must teach computer science to prepare students for the future (Krauss & Prottsman, 2017; Sentance & Csizmadia, 2017). States across the country recognize this need and are working towards building and adopting state standards, recommending curriculum, and most importantly, providing professional development (Harmon, 2018). Developing a deeper understanding of the experiences elementary teachers face in teaching computer science was essential to identifying strategies and supports they feel they may need to teach all students computer science.

Computer science education is beginning to expand in districts across the country. As more districts and schools offer computer science learning opportunities, teachers have the most significant responsibility in teaching the content. There was particular relevance for the participants in this case study since California recently launched a state-backed campaign to give all students access to computer science titled "CSforCA" (Harmon, 2018). Teachers are critically important as they dictate and guide the content taught in their classrooms (Ferguson-Patrick, 2018). Consequently, the mission of the CSforCA initiative requires providing teachers with appropriate levels of professional development and support as they teach computer science content and standards. If teachers are unable to appropriately identify best pedagogical practices, proper materials, and learning objectives, they are unable to provide meaningful computer science education for their students.

Definition of Terms

Understanding terms in a research study are essential to clarify key vocabulary and their definitions. The terms and definitions listed below provide a common understanding and prevent potential misunderstandings (Bloomberg & Volpe, 2012).

Computational artifacts: Anything created by a human using a computational thinking process and a computing device. A computational artifact can be, but is not limited to, a program, image, audio, video, presentation, or web page file (College Board, 2017).

Computational thinking: A human thought process that cultivates the ability to formulate problems so that their solutions can be represented as computational steps or algorithms to be executed by a computer (Lee & Ko, 2011).

Computer science: The study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society (Tucker et al., 2006).

Digital literacy: The general use of computers and software for productivity. Digital literacy is the ability to use information and communication technology for aspects of reading and writing in the digital age (Heitin, 2016).

Unplugged activities: An emphasis on computational thinking through an approach of learning computer science concepts through physical, kinesthetic experiences that can be taught independent of access to computers and the Internet (Krauss & Prottsman, 2017).

Assumptions, Delimitations, and Limitations

Two main assumptions informed this study. First, I assumed that teachers had some understanding of computer science and integrated the content into their instructional time. Secondly, I assumed teachers were honest in their responses to the research questions. This was because participation in the research was entirely voluntary, and participants were free to withdraw at any moment.

Delimitations, or boundary choices, were implemented based on the time constraints and available resources. As such, I delimited the number of participants to those who were identified through purposeful sampling as having had exposure to professional development and at a school where computer science was a site priority. The small sample size allowed me to gain an in-depth understanding of the participants' experiences, but a more extensive study over a longer period could have produced results not found in this study. A primary strength in case study research is the thorough and insightful description of the case (Merriam, 2009). Thus, this study provided a thorough and comprehensive view of the experiences elementary teachers faced in teaching computer science. Although the study produced deep insight from the sample, the results may not be transferable to other populations and states. Another delimitation in the case study was focusing on one school in one district in California. This research setting was the most accessible to me and allowed for the best use of resources in gathering data. The final delimitation was the selection of a school site where computer science was a priority for the staff. Teachers from other schools in the district may prioritize computer science education in their classrooms, but they may not have had the same level of peer support if they were the lone advocate on campus.

Researchers must identify limitations of the research design. First, limitations of this qualitative case study included the limited amount of time to conduct research, my focus on a single district, and the self-reporting by participants of their experiences and needs. The limited time to conduct this research was the first limitation in this study. The shortened time allowed me to complete this study, but emphasized one point in time as participants' experiences could have changed by the end of the year. I studied one school district because computer science implementation may vary greatly across the region, and research across multiple districts and in other states may present unique perspectives specific to the student populations and demographics in different parts of the country. Lastly, participants self-reported their experiences during the interviews and provided documents they felt were relevant to this study. The limitations of the study presented certain benefits, as I engaged in purposeful sampling and engaged in an in-depth examination of elementary teachers' experiences in the district.

Recognizing the research limitations and delimitations, I was mindful of bias that could occur in the research process. I ensured confidentiality and transparency with all participants throughout the study. Furthermore, I attempted to maintain objectivity and reduce bias in data collection and analysis by member checking and acknowledging ethical issues throughout the study.

Summary

Computer science is not limited to studying computing systems such as desktops and laptops. The broad adoption of technology in everyday life requires access to both digital hardware and software. However, an understanding of these technologies is critical to developing awareness of the impact of global connectivity in our digital age. Furthermore, the science of computing is different than technological concepts such as digital literacy and educational technology. The distinction in computer science is the understanding of both how and why computing technologies function. Thus, the principles, concepts, and practices within computer science are essential to students for both college and career readiness. Beyond the development of technical skills and abilities, computer science instruction can also promote and foster creativity, inclusive mindsets, and problem-solving skills beyond basic technology usage.

The emphasis on understanding how the technology works by building conceptual knowledge through computational artifacts signals a contrast in computer science education from a basic digital literacy curriculum. Specifically, in elementary school, foundational concepts and practices in computer science can help prepare students to navigate digital technologies while creating a pipeline for secondary levels. Preparing students for college and careers by cultivating creative and critical problem solving is essential to global competitiveness.

This qualitative case study was designed to gain a deep understanding of elementary teachers' experiences teaching computer science in a large urban district. By better understanding elementary teachers' experiences, districts can provide supports teachers need to successfully teach computer science to elementary students. Chapter 2 presents an overview of the current literature regarding computer science education, methodological issues, and critique of previous research. Chapter 3 focuses on research questions, case study purpose and design, the process for sampling, data collection, and data analysis. Chapter 4 comprises the reports and results of the collected data. Finally, Chapter 5 presents the conclusions, summary of findings, and recommendations for future research.

Chapter 2: Review of the Literature

Search Strategy

To most effectively explore issues surrounding computer science education and professional development, a subset of literature was selected based on relevance to the recent growth of computer science education, types of professional development and training, and the positive and negative impact on teachers who receive training. The search focused on significant findings within K–12 education that centered on computer science, learner experience, teacher efficacy, and professional development. The literature search revealed teachers face multiple barriers in teaching computer science. Most of the available literature focuses on secondary schools as computer science is typically taught by single subject teachers. As such, more research is needed in regard to elementary school teachers teaching computer science and the possible supports they need. I conducted the search using ERIC, ProQuest, Wiley Online Library, Google Scholar, and other articles accessed through Concordia's library resource. Keywords included *computer science, computer science education, STEM, integration, content knowledge, pedagogical content knowledge, professional development, digital literacy, pedagogy, constructivism, constructivist, instruction, methodology, elementary, secondary, SAMR, TPACK, coding, programming, teacher training, teacher preparation, technology, instructional technology,* and *educational technology.*

Introduction

Computer science is quickly becoming a necessary part of K–12 education. Until recently, computer science was a specialized course in some schools and a mostly unknown entity. However, with the recent "Computer Science for All" initiative launched by the White

House in 2016, schools, teachers, and administrators are recognizing the value in teaching computer science (Krauss & Prottsman, 2017). Despite the high level of interest and increased awareness, schools prioritize subjects that are included in required testing, limiting computer science as an available option for subgroups of students (Margolis et al., 2012). However, schools must leverage their limited resources to promote computer science education for all students as rapid advancements in technology across industries demands computer science literacy as a critical skill. Technology impacts every aspect of daily life and software and hardware interactions continue to increase exponentially in school, work, and recreation (Teo & Zhou, 2017). Even though computer use in classrooms is common for everyday teaching and learning, students often take passive roles as users of hardware and software (Carter, 2014; Peters & Araya, 2011). However, computer science education is distinct from digital literacy in that students develop computational and logical thinking skills to prepare them for the future (Lee & Ko, 2011). Thus, learning computer science provides students an opportunity to learn a new form of literacy, enabling them to write their stories and create with technology.

Historically, computer science has been inaccessible to most K–12 students. In California, home of Silicon Valley, 65% of high schools offer no computing classes (Level Playing Field Institute, 2015). At the K–8 level, the situation is glaring as many students do not receive the opportunity to learn this new form of computing literacy. For some minority students, equitable access is even more limited and problematic (Wang, Hong, Ravitz, & Hejazi Moghadam, 2016). While 60% of California’s student population is African American or Hispanic, these students comprise only 16% of students taking AP Computer Science A exams (College Board, 2016). These statistics transfer to discrepancies in the industry as those same groups make up 15% of the technology workforce (Equal Employment Opportunity Commission, 2016). Even with limited access to high-quality computer science education, students recognize the need to learn how technology works. Over 93% of students in small towns or rural districts indicated they will have a job in the future that requires knowledge of computer science (Wang et al., 2016). Recognition and demand for computer science by parents and students have also grown, requiring administrators and teachers to address this deficit in computer science education.

The general public recognizes the need for computer science education and Americans believe computer science is as essential to learning as reading, writing, and math (Krauss & Prottsman, 2017; Sengupta et al., 2013). Furthermore, 91% of parents want computer science education as part of daily instruction as they see value in their children learning and understanding how technology works (Wang et al., 2016). There is now a necessity in schools to have teachers trained to teach computer science, increasing access for all students as a core subject in addition to specialty courses (Israel et al., 2015). Careful implementation requires a specific focus on providing computer science learning opportunities as DeJarnette (2012) found academic and official curricular support did not systematically translate into reality. Computer science curriculum may not be enough to guide teachers who teach decide to teach the concepts without having a clear understanding of the content (Ozturk et al., 2018). Developing an understanding of their teachers' experiences is critical to providing adequate and effective support systems.

Curricular alignment and early exposure are essential components to expand access and improve computer science education. As such, elementary teachers need support and professional development to find ways to integrate computer science instruction into the day. Thus, the purpose of this study is to gain an understanding of the experiences of elementary teachers in a large urban district who teach computer science. Developing an understanding of teachers' experiences may reveal challenges teachers face teaching computer science. This may provide insight into appropriate supports district and school administrators can provide to develop elementary teachers' competencies and confidence in teaching computer science. The literature review presents the conceptual framework and a brief overview and history of computer science education in K–12 schools. Then, the literature review discusses current research into teachers' experiences and challenges they face with technology and computer science. I explore the strategies teachers must develop to successfully learn and teach computer science combined with pedagogical content knowledge. The review ends with recommendations on specific strategies teachers can build to enhance teaching and learning to provide high-quality computer science education for all students.

Conceptual Framework

The push for computer science education has increased demand for teachers to receive training and provide learning opportunities in their classrooms (Lye & Koh, 2014; Peng, Wang, & Sampson, 2017). Historically, computer science has not been integrated into school, and as such, teachers are now facing new challenges as they receive professional development. Often, teachers are forced to take on a lead learner approach as they are students of computer science content trying to fit their learning with pedagogical experience. As such, constructivism is a fundamental foundation for learning and teaching computer science. In a constructivist model, learners create their understanding, combining what they know and believe with new experiences they live through (Richardson, 1997; Wild, 2015). Teachers must commit to learning what computer science is and how to teach the very concepts they are learning as the content is not covered in preservice programs. As such, I explore the themes and ramifications of elementary teachers' experiences teaching computer science in their classrooms based upon a conceptual framework of the constructivist learning model.

Overview of Constructivism

Constructivism is a theory of learning where learners construct knowledge through past experiences, exploration, and interaction (Ackermann, 2001; Dev, 2016; Turkle, 1984). The importance of doing and developing conceptual understanding through active learning facilitates the learning process, allowing for building and reframing of new and existing knowledge (Papert, 1980; Rogoff & Lave, 1984). In this view, learners bring their own experiences and existing knowledge as a foundation to scaffold and build new learning. Constructivism emphasizes knowing students in depth to effectively integrate what teachers learn about their students and understand the knowledge each learner uniquely holds in class (Gupta & Gupta, 2017). Personalizing learning experiences to engage students requires a shift away from packaged curriculum and one-size-fits-all approaches. A constructivist approach presents teaching opportunities to demonstrate respect and mindfulness of students' diverse backgrounds and knowledge.

Researchers have indicated effective pedagogical practices are enhanced by teachers who purposely and mindfully integrate technology into their classrooms (Ogunkola, 2008). Teachers willing to take risks and adopt emerging technology for classroom teaching and learning often have constructivist compatible pedagogical styles (Allen, Webb, & Matthews, 2016; Judson, 2006). Within this context, teachers' technology implementation through a constructivist lens includes thinking practices aligned to elements of computer science core practices found in the K–12 Computer Science Framework. The core practices represent how learners do computer science while building conceptual knowledge of the critical content areas. Teaching how to access and use appropriate knowledge, creatively and collaboratively problem solve, and understanding inquiry skills to develop new awareness are fundamental to computer science. Core practices are learned and adopted through constructivist learning for both teachers and students as computer science is a new content area for most educators (Pourhosein-Gilakjani et al., 2013). Furthermore, rapid changes and development in technology indicate even in computer science, specific content knowledge may become obsolete. Teachers must not solely learn programming and coding languages but should learn fundamental ideas and skills based on thinking, action, and description (Gómez, 2015; Zendler, McClung, & Klaudt, 2015). They can then equip their students with knowledge and abilities not confined to specific technologies, but rather cross-curricular relevancy in computer science and general education.

Constructivism in This Study

Preparing computer science teachers presents challenges, since constructivist perceptions of the learning process are in direct conflict with science and mathematics, where content knowledge acquisitions take priority over constructivist pedagogy. Plourde and Alawiye (2003) found a fundamental shift towards constructivist teaching and learning is needed to fulfill the responsibility of producing members of society who have the skills required to be capable and productive citizens of the 21st century. Furthermore, constructivist teaching engages and motivates learners, requiring them to take more active roles to apply higher-order thinking skills.

Learners problem-solve, collaborate, and implement shared inquiry practices through social learning (Ferguson-Patrick, 2018; Papan & Sompong, 2012; Wild, 2015). Constructivist learning is essential in preparing teachers to teach computer science as they both develop classroom practices and navigate curriculum and content new to their knowledge and understanding. Papert (1980) compared computers to mud pies as things to think with, mediums to express ideas. Constructivism posits that learning happens best through building things that are tangible and shareable (Ackermann, 2001; Dev, 2016). In computer science, the hardware is not merely a host device, but a raw material limited only by the creativity of its user (Krauss & Prottsman, 2017).

Papert (1980) suggested learners build and construct knowledge by engaging with others through collaborative discussion and shared experiences. Thus, sharing ideas and learning seems essential to building capacity in computer science learners as well. Bell, Maeng, and Binns (2013) discovered collaboration dramatically impacts teachers integrating digital resources and building skills with educational technology. Teachers who were provided regular opportunities to collaborate with peers consistently and substantially used instructional technology to enhance instruction while promoting more in-depth understanding of content. Michalsky (2012) made similar recommendations for shared learning with professional development, as teachers benefit most from learner-centered, peer-collaborative, active-learning environments. Then, to develop effective teachers, learner-centered constructivist learning environments are critical to improving professional efficacy in teaching and learning computer science (Fluck & Dowden, 2013; Page & Margolis, 2017).

Although constructivist teaching provides distinct advantages, Gupta and Gupta (2017) found challenges in successful implementation models. Professional development for constructive teaching models is costly and require extensive time commitments as teachers build on their experiences and create new learning models. Teachers practice and implement the pedagogical strategies they gain in training, reflect on students' feedback, and add new information to their prior knowledge. Computer science professional development presents additional hurdles of developing teachers' knowledge of pedagogical strategies and computing concepts such as abstraction, iteration, algorithmic processing, and computational thinking (Wing, 2006; Yadav et al., 2017).

Review of Research Literature and Methodological Literature

History of Computer Science Education

A clear understanding of computer science education history is needed to contextualize the purpose of the study and the current state of computer science education. The context provides a baseline of understanding in the past, current, and potential future state of this unique educational domain. The hardware we use daily did not always come with a graphical user interface, applications software, and the ability to connect with other devices. The first computer could only follow explicit and precise commands based on what its user programmed it to do, requiring knowledge of computer science and an ability to produce the appropriate inputs generating the needed outputs. Today, much of the activity in computers is hidden from the user as devices run programs written by others (Krauss & Prottsman, 2017). In the early 1970s, Papert (1980) introduced children to computer science in the classroom because a computer user was by necessity, a computer programmer. Piaget (1952) followed constructivist teaching ideal as students learned through inquiry-based learning and interactive environments. Computers were not merely hosts, but a raw material shaped by the creativity and technical ability of their users. As such, computer science involves much more than the use of computer systems; it holds broader concepts such as understanding algorithmic processes carried out through computer programs, or the impact on society with hardware and software designs (Page & Margolis, 2017; Tucker et al., 2006).

Computer science is not digital literacy, digital citizenship, or information technology, the concepts most commonly used to describe computer science education. Instead, computer science builds upon these terms, scaffolding upon them and going further in complexity and depth, with an emphasis on use rather than consumption. Students are not passive consumers of technology, and they are called upon to understand how and why computing technologies work, building upon their conceptual knowledge to create computational artifacts. Margolis et al. (2012) found computer science education in schools was inequitable as subgroups of students, specifically African American, Hispanic, and female subgroups only learned rudimentary digital literacy skills in their classes. This creates a disparity in learning opportunities where high quality, sustainable computer science education becomes inaccessible to schools with high numbers of low-income students of color.

Computer science learning opportunities can also be limited to students who have resources and attend schools with multiple course offerings. Margolis et al. (2012) found having computer science classes does not necessarily guarantee access to high-quality learning opportunities for all students. Even when computer science courses are offered, advanced conceptual understandings of technology are often replaced by digital literacy skills instead of rigorous computer science curriculum. In the rare occasion an advanced computer science course is offered, the curriculum is not sustainable as the curriculum emphasizes specific tools or programming languages which most students find disconnected and too complex (Buzzetto-more et al., 2017; Ukoha, & Rustagi, 2010; Vakil, 2018). Furthermore, schools are unable to find quality teachers for the courses as there is a lack of computer science teacher certification programs, limited opportunity to improve pedagogical skills, and opportunities to learn content.

Computer science teachers face a tremendous challenge because they must balance learning new concepts, develop pedagogical skills, and persevere to learn and teach new curricula. As technology continually evolves, computer science resources can quickly be antiquated as faster hardware and new software become available to users. An increasingly vocal call and demand for the inclusion of computing curricula in schools further places additional expectations on teachers to learn and develop strategies for computer science education. Guzdial (2014) found teachers conducting Internet searches with generic phrases to locate computer science teaching and learning resources. However, without the proper background knowledge, the search results generated unproductive resources and teachers were unable to gauge the usefulness of the content. Consequently, the need for professional development is increasingly critical in preparing teachers.

Professional Development

One of the most impactful and influential resources for increasing access to high-quality computer science learning opportunities are classroom teachers. Blazar and Kraft (2017) found qualified teachers have a significant impact on raising student achievement regardless of circumstance and challenges. However, teachers entering or in the profession are unlikely to have experienced the extensive use of instructional technology and computer science curriculum. Even those with some personal competence in technology are not likely to translate their ability to incorporate computer science in their classrooms (Fluck & Dowden, 2013; Love & Strimel, 2016). As such, a significant challenge within computer science education is preparing and supporting teachers who are expected to teach the content in their classrooms (Ryoo et al., 2015). Professional development opportunities are limited, and few teacher preparation programs offer or require computer science courses as part of the credentialing program. Consequently, even the most well-intentioned and talented teachers have limited effectiveness due to their access to the knowledge and resources required to provide robust learning opportunities. Furthermore, developing best practices and curriculum are limited when there are not many other teachers in the district who are also teaching computer science. Teachers are unable to build a network of support and community, having limited avenues to find the necessary resources to develop their practice (Page & Margolis, 2017). Most high schools have no computer science department; instead computer science teachers are a part of business or math departments. Unsurprisingly, these teachers lack collegial support in planning and teaching computer science (Margolis et al., 2012).

Professional development is most likely to create lasting change in practice when teachers experience constructivist learning. Michalsky (2012) recommended teachers engage in learner-centered, peer-collaborative, active-learning environments. Professional development within these parameters challenges teachers to actively and autonomously undertake learning processes. As such, transformation in practice and mindset become simultaneously impacted by an increase in teachers' competence and comfort with instructional technology (Aslan & Zhu, 2017). Teachers who successfully develop self-efficacy to use technology and teach digital concepts effectively are more likely to take risks and manage challenges teaching new content (Latham & Carr, 2012; Wilkerson-Jerde, Wagh, & Wilensky, 2015).

Professional development opportunities can help move education into the digital age of computer science, but training in itself is not enough. Continuous professional development within school teams and building professional learning communities are essential to encourage learning (Cutts, Robertson, Donaldson, & O'Donnell, 2017; Twining, Raffaghelli, Albion, & Knezek, 2013). Furthermore, content as the sole focus objective in training is not enough for efficient, professional development. Building practice-based learning networks is a critical element in promoting teacher efficacy with computer science and computational thinking (Patrick, Elliot, Hulme, & McPhee, 2010; Twining et al., 2013). Additional supporting elements outside of professional development further help build and establish teachers' confidence and competency, creating high-quality learning opportunities with computer science concepts.

Teacher Efficacy

Teaching is more meaningful when educators shift from a tacit knowledge of instruction to a more explicit one with increased efficacy. Educators must balance curricular and pedagogical goals to meet the needs of their students. In computer science, educators typically do not have the same level of content comfort, knowledge, and background as they would with content aligned with their teaching credentials. Developing knowledge and ability to support computer science learning is difficult because teachers are balancing content acquisition and pedagogy (Ladner & Israel, 2016; Wilkerson-Jerde et al., 2015). However, teachers who successfully balance both can provide exceedingly relevant and meaningful learning opportunities. Computer science allows teachers to offer responsive digital learning opportunities that connect with students' knowledge and needs (Margolis et al., 2012). To effectively engage in the content, teachers must eventually develop some knowledge on how to build and design computational artifacts with their chosen program. However, this challenge is often daunting and challenging as teachers feel they must have this knowledge prior to teaching their students.

Teachers may initially face tremendously challenging roles in supporting computer science classrooms as they must also focus on computational modeling. Creating computational models and responsive learning experiences requires an understanding of software, progress in crafting models, and building connections with knowledge (Peng et al., 2017; Wilkerson-Jerde et al., 2015). Although there are multiple resources to create computational models, most teachers do not have the computational and technological competency or knowledge to identify the most appropriate resources. Furthermore, teachers also face challenges in recognizing and identifying what productive student responses look like within the environments. Teachers need to develop best practices, but lack the required knowledge within computer science content and pedagogical approaches to make appropriate decisions. Consequently, building teacher efficacy through professional development and ongoing support is critical to establishing successful instructional programs.

Aslan and Zhu (2017) found integrating instructional technology into teacher training programs played a tremendous role in teachers' integration of technology into their teaching practices. Teachers who learn to use technology and begin to build a favorable judgment towards their abilities start developing high competencies that manifest in a tendency to use technology more extensively with their students. However, most computer science teachers are current practitioners who are adding to their current duties. These teachers have credentials in other subject areas and need established supports—ideally multidisciplinary teams to collaborate and integrate computer science concepts (Twining et al., 2013).

One strategy for developing teacher efficacy is creating and supporting communities of practice. Teachers who meet regularly to collaborate and discuss content and pedagogical needs and challenges find an increased level of confidence (Yadav et al., 2017). Furthermore, building vertical levels of support with mentors or master teachers may also help develop teacher efficacy. In the United Kingdom, teachers receive professional development through Computing At School (CAS) Hubs. Teachers and researchers come together with a "specific aim of providing (at least) one idea that can be taken and tried in the classroom" (CAS, 2013, p. 5). Offering smaller piecemeal approaches and strategies to computer science education allows teachers to implement incrementally without feeling overwhelmed by the amount of content they need to learn and teach. Furthermore, teachers connect with master teachers who help support and meet the needs of their peers (Sentance, Humphreys, & Dorling, 2014). Creating in-person relationships and coaching support is critical in building up newer teachers' understanding of computer science content and pedagogical best practices.

Building teacher efficacy through online training and meetings presents challenges and opportunities. In situations where in-person meetings and support systems are not feasible, building online communities of learning can make significant progress in teacher competencies. Hepp (2015) found online collaborative work allowed teachers to combat feelings of isolation as they worked in teams, sharing their classwork with others. Online systems allow teachers to document and share their innovations, challenges, and strategies. Blending learning models with a significant component of online interaction requires collaboration between content experts and online learning experts (Papanikolaou, Makri, & Roussos, 2017). Online communities also expand learning opportunities to larger populations and open up possible entry points for teachers in schools without any other computer science teachers (Park, Johnson, Vath, Kubitskey, & Fishman, 2013). Facilitators face a tremendous challenge both in having the appropriate content knowledge and also being able to facilitate discourse between teachers. In initial stages, this may require a content specialist without knowledge of pedagogy, creating a team-teaching model.

Computer science should be viewed as an opportunity rather than a threat, with great potential to change and transform education. Intellectual activity associated with programming can improve teacher digital competence, update methodologies, and improve student learning. These new skills are linked to creative activities, innovation, communication, and collaboration and require teachers to reconsider what educational technology to use in the classroom (Buckler, Koperski, & Loveland, 2018; Buss, Wetzel, Foulger, & Lindsey, 2015). Teacher efficacy significantly increases with changes in mindset as they create new epistemological beliefs (Goldsmith, Doerr, & Lewis 2014). Providing guidance and opportunities for teachers to meaningfully create new perceptions is significant since beliefs about content, teaching, and learning impact perceptions and pedagogical approaches (Bender et al., 2016). Restructuring or adapting current views of expectations and learning outcomes requires renewed explorations into technological tools. Teachers can create learning environments where the experience of experimenting and making mistakes is a benefit rather than a hindrance. This change management can further drive innovation and encourage learners to use errors as opportunities to learn and progress.

View of Computer Scientists

Issues of self-perception and computer science have been problematic for students. A lack of confidence to learn computer science can lead to an exclusion of student subgroups' participation in learning opportunities at their schools. Strikingly, African American students, regardless of income reported having less opportunities to learn computer science at school (Margolis et al., 2012). White and Asian students were more likely to have a computer at home with consistent and reliable access to the Internet. Beyond access to curriculum in school, African American and Hispanic students were less likely to use computers in school or have access in their homes (Margolis et al., 2012). The limited exposure to technology especially impacted students who also found disparities in relationships with role models and their own abilities in computer science. African American and Hispanic students were less likely than White and Asian students to have adult role models in their lives who work with computers or technology (Gershenson, Holt, & Papageorge, 2016). Consequently, these students found limited exposure to computer technology and demonstrated lower levels of confidence in their ability to learn computer science. This dangerous misunderstanding can be exacerbated when teachers share similar perceptions of themselves and their students.

Teaching computer science requires a different knowledge base than other content areas. Teachers need to recognize that learning to program and code is an element of computer science education rather than the core, and they should rather emphasize computational thinking and creativity using computer science concepts (Buitrago et al., 2017; Falkner & Vivian, 2015).

Having teachers reflect on their perceptions of learning and teaching can directly influence their decision-making (Srikoom et al., 2017). Consequently, teachers' perceptions of themselves as computer scientists and their view on students' abilities to learn computer science are critical elements for building sustainable learning opportunities (Tang, Baer, & Kaufman, 2015). Developing teachers' confidence in their ability to understand and solve complex problems with computational thinking may provide opportunities to expand students' access to learning computer science (Grover & Pea, 2013; Lye & Koh, 2014). Thus, significant inequities in access to computer science education requires a concentrated effort in developing and supporting teachers who are able to democratize access to computing. However, equally important in bridging the digital divide is the actual learning experiences provided for students.

Computer Science Learning Experiences

Experiences in computer science learning are critical to consider for successful teaching and learning to occur in classrooms. Understanding learner needs and challenges may help teachers develop strategies and reflections to provide the best possible learning experiences. Consequently, student learning experiences can be ineffective and misaligned due to learner experiences from teacher preparation programs. In regards to instructional technology, the primary focus of teacher preparation programs is in basic digital literacy and offers insufficient learning experiences in computing skills (Papanikolaou et al., 2017). Cetin (2017) found teachers given educational technology experiences in preparation courses were more likely to implement technology into their teaching. However, the integration did not necessarily result in efficient use of technology in their classrooms. As learners, teachers typically do not acquire appropriate and meaningful knowledge to naturally synthesize their digital literacy with pedagogy and understanding of content. For teachers interested in obtaining additional training and certification for computer science, there are limited number of programs in the United States that currently offer computer science certification as the primary license or credential (Yadav et al., 2017). Unfortunately, the limited programs that do offer computer science as part of their curriculum have, "no tangible relationship to what is needed to teach in a computer science classroom" (Gal-Ezer & Stephensen, 2010, p. 63). Consequently, computer science is incorrectly attached to other subject areas such as Technology Education, Education Technology, Instruction or Industrial Technology, Networking, Management Information Systems, or other subjects' regions that use computers to support learning (Khoury, 2007; Yadav et al., 2017).

The inconsistency of learning experiences in preparation programs and professional development creates conflicts between what learners experience and are expected to teach. Thus, as teachers are unable to experience useful computer science pedagogical practices and curriculum from learner perspectives, they struggle to offer engaging learning opportunities for their students. Skoretz and Childress (2013) found an increase in technology integration, but limited learning experiences, to develop and build context and competencies for problem-solving, critical thinking, reasoning, and core concepts in computer science education. Teachers used computers to introduce, reinforce, or reteach ideas rather than leveraging them as dynamic learning tools. Thus, teachers need to experience computer science from student perspectives so that they can recreate those learning experiences in their classrooms. They must successfully navigate dual roles as teachers and students, presenting a challenging shift in thinking and preparation. Cetin (2017) found computers had a central role in enhancing teaching and learning but were not enough to change the classroom environment. Thus, adapting learning experiences in computer science is imperative to attracting and sustaining student interest and engagement.

Constructivist strategies and approaches significantly impact and influence learner experiences in computer science (Ackermann, 2001; Piaget, 1952; Wilkerson-Jerde et al., 2015). Environments inspired by constructivist theories allow learners to navigate within a learning space while reorganizing prior knowledge and generating new ideas. Abstract and challenging computer science concepts are understood as learners engage in programming and find relationships of critical phenomena. Learners must also uncover opportunities for discourse as research shows effective instructors attend to and respond to students' thinking by guiding discussions and establishing disciplinary norms (Giannakos et al., 2017; Wilkerson-Jerde et al., 2015). Capacho (2016) found learning environments that allow for exploration of preconceptions and interactions within a virtual space help create new knowledge for students. Furthermore, a combination of learning networks, a teacher's guide, peer collaboration, and learning communities in computer science further help the creation of new and validated knowledge (Ryoo et al., 2015).

Virtual learning and online collaboration are possible through digital resources and the Internet. Collaborative learning through online methodologies allows learners to communicate and interact with social and cultural contexts different from which they belong (Dabbagh & Kitsantas, 2012). Capacho (2016) explained collaborative learning is an approach that allows learners to be aware of the process of acquiring their cognitive structures as they compare and contrast their learning. Teachers who navigate coursework and collaboration tools through online learning can then use similar tools to craft learning experiences for their students through blended and flipped learning models.

When students are not adequately introduced to technology, they can display physical and emotional responses. Sanalan (2016) concluded computer anxiety produced biological and behavioral outputs such as sweaty palms, dizziness, and shortness of breath. Teachers who experience similar fears with technology are unlikely to respond well to professional development and build competency. Furthermore, they are more likely to "communicate their psychological, behavioral or cognitive characteristics with their students" (Sanalan, 2016, p. 222). These symptoms can negatively impact students' responses to technology and particularly computer science as the rigor exceeds digital literacy skills. As such, teachers need safe environments to explore and learn with enough scaffolding and supports to ease their anxiety. Learner-centered environments could lead to sustainable practices as teacher efficacy significantly increases with changes in mindset. New epistemological beliefs present opportunities for teachers to reflect on their practice and form new pedagogical strategies to implement in their classrooms (Goldsmith et al., 2014).

Review of Methodological Issues

Currently, most research in computer science education in public school settings has heavily focused on secondary teachers (Montoya, 2017; Yadav et., 2017). As states move towards adopting standards, primary teachers will face challenges in teaching computer science concepts (Sentance & Csizmadia, 2017). Thus, there is a need to understand elementary teachers' experiences teaching computer science to better identify possible support systems. Attempts at analyzing beliefs and motivational orientations in computer science teacher education and impact of curricula are narrow due to the limited availability of research (Bender et al., 2016; Sentance & Csizmadia, 2017). Computer science professional development for elementary teachers is a sparsely investigated field with limited valid measurement instruments to determine the occurrence of beliefs and motivational orientations (Bender et al., 2016). As such, comprehensive interview studies with participants and analysis of documents in computer science are strategic approaches to identifying possible baseline analyses (Sentance & Csizmadia, 2017; Yadav et al., 2017). Furthermore, most existing measurement approaches stem from self-assessment and small convenience samples indicating a need to clarify existing uncertainties.

Zendler et al. (2015) found concerns of empirical analyses for central concepts to be covered at specific grade levels, noticeably in elementary instruction. The proper combination of social interaction and instructional methods to teach fundamental concepts is difficult, requiring extensive observations and appropriate coding (Creswell, 2013). Analysis of essential concepts with teachers' development created "regressed experts," a new hybrid state of teaching with disconnected practice and knowledge. Teachers acting as "regressed experts" combined elements from both novices and experts with limited strengths and inconsistent knowledge (Liberman, Kolikant, & Beeri, 2012, p. 257). Discovery of this new teaching state presented further challenges as researchers have yet to sufficiently distinguish the capacity of teachers attempting to teach and learn computer science. Teachers also live and work in unique situations based on multiple factors which may include site and district funding, student populations, available resources for professional development, and availability of classroom funding for hardware and software. Even within the same local educational agency, the variables may differ based on grade level and school location. Consequently, teachers in the same community may need to navigate different circumstances, further creating diverse responses to challenges and barriers in computer science education.

Surveys, enrollment data in current courses, interviews, and focus groups present a large part of the information we currently have within computer science education (Wang et al., 2016). Although interviews can provide comprehensive data, a limitation in qualitative research is the limited number of participants. Considering the small number of computer science teachers, this is more glaring at elementary levels as there is not a significant population to draw from to further shift research. Thus, the small number of participants limits generality to the broader community of computer science teachers (Lye & Koh, 2014; Ni & Guzdial, 2012; Yadav et al., 2017). To gather granular information in an attempt to understand unique differences and commonalities, Merriam and Tisdell (2016) recommended a case study to investigate within a bounded system, uncovering in-depth description and analysis. Similarly, Creswell (2013) declared that a good case study is an opportunity to present an "in-depth understanding of the case" (p. 98).

Synthesis of Research Findings

Previous research in computer science education is limited in elementary contexts as the bulk of information stems from secondary experiences (Ozturk et al., 2018**).** Subtopics discovered in the literature are the history of computer science education, professional development, teacher efficacy, view of computer scientists, and computer science learning experiences (Gal-Ezer & Stephenson, 2010; Master, Cheryan, & Meltzoff, 2016; Mouza, Marzocchi, Pan, & Pollock, 2016). However, the majority of data are in high school and middle school environments.

Computer science education has historically been limited to specific subgroups of students (Montoya, 2017; Wang et al., 2016). Certain students of color and socioeconomic status, as well as female students, have not had equitable opportunities to learn computer science (Ashcraft & Breitzman, 2012; Buzzetto-more et al., 2010; Master et al., 2016; Mouza et al., 2016). Montoya (2017) found a need to recruit, train, and retain a diverse computer science teaching workforce to positively impact low-income and Hispanic students. As such, professional development should include instructional practices and recommendations to promote inclusivity and access to all students.

Professional development as the sole pillar of support is not enough to sustain computer science education. Teachers need comfortable learning environments to explore content and concepts in learner-centered environments. Furthermore, support systems of communities of practice are essential in developing teacher efficacy and competencies. Teaching computer science may feel foreign, and collaboration with peers to plan and implement their new knowledge is critical to developing a willingness to teach computer science to students. Historically, engaging and meaningful professional development opportunities were limited, and teachers were unable to learn relevant computer science concepts and content to become knowledgeable enough to build confidence (Montoya, 2017; Ni & Guzdial, 2012; Yadav et al., 2017). Teachers may find themselves in situations where students understand more content than they do, requiring an ability to model lead learning.

Teachers will likely find successful strategies as well as challenging situations dependent on their level of knowledge as well as their students. Teachers new to computer science face tremendously unique teaching opportunities which may be supported through coaching and community collaboration. Teachers can also benefit from opportunities for reflection as they face these new challenges, allowing them to build capacity and further their learning and refining of pedagogical approaches within computer science. Teachers are better suited to develop their efficacy in sustainable ways when they reflect and learn how to integrate instructional technology into classroom practice through informal discussions and conversations with peers (Skoretz & Childress, 2013; Windschitl & Sahl, 2002).

Previous researchers of computer science professional development have suggested teachers and students face multiple challenges in teaching, learning, and integrating computer science as part of their school experience (Buzzetto-More et al., 2010; Lachney, 2017). Furthermore, issues of access and equity are apparent for underrepresented groups, presenting further challenges and barriers to effective teaching and learning opportunities (Lye & Koh, 2014; Ni & Guzdial, 2012). Still, systemic change in furthering computer science education is possible and at the forefront of the challenge is successfully building efficacy in teachers' understanding and pedagogy with learner-centered professional development and sustainable support systems (Papan & Sompong, 2012; Zendler et al., 2015).

Critique of Previous Research

Currently, there is not enough information from elementary teachers' experiences to guide the development of K–6 computer science curriculum options and sustainable support systems for teachers and students. The majority of research on computer science curriculum and teaching experiences are based on secondary teachers' experiences and fail to address the unique challenges faced by elementary teachers (Sentance & Csizmadia, 2017). Computer science education provides an opportunity to develop students and prepare them to compete in a digitally-connected economy. Globally, the emphasis on expanding computer science education is being approached differently in China, India, Europe, and the United States (Guzdial, 2014; Yadav et al., 2017; Zendler et al., 2015). Empirical findings did not support the idea that computer science educators differ in their evaluations of central content concepts of computer science and central process concepts (Zendler et al., 2015). Although the research provided information on drafting curriculum for computer science, the sample utilized computer science professors who already have a deep understanding of the content.

There are concerns with equitable systematic accessibility to computer science education curricula that are currently not established to meet the needs of teachers and all students. Concerns with equity remain as participation in AP computer science is not spread out among student subgroups. In particular, African American and Hispanic students are much lower than other subgroups as they represented less than 15% of all test takers (Wang et al., 2016). However, between 2012 and 2014, there was a 50% increase in the number of students who took AP computer science. Researchers utilized surveys with a large sample to explore perceptions of access and barriers to computer science education (Wang et al., 2016). However, the researchers did not include documents in their study. The addition of documents can add significant insight into a study, as lesson plans and teachers' meeting notes may produce information about discrepancies in computer science learning opportunities.

Participant samples in recent computer science education research currently skew heavily towards understanding high school teachers' experiences and fail to include elementary teachers' experiences (Margolis et al., 2012; Ryoo et al., 2015; Yadav et al., 2017). Thus, understanding elementary teachers' perceptions requires in-depth interviews to understand and develop rich descriptions of experiences (Creswell, 2013). Professional development for elementary teachers may increase learning opportunities for students as teachers can develop pedagogical practices and shifts to teach computer science concepts. Furthermore, local educational agencies must train and support teachers who have the knowledge and confidence to teach computer science. For example, 75% of principals and 74% of superintendents stated they did not offer computer science because they lacked highly qualified teachers who had the necessary knowledge and skills. Additionally, 79.3% of high school principals reported that finding a qualified teacher was the most significant deterrent to offering a section at their sites (Wang et al., 2016). Recognizing teachers' experiences in teaching computer science concepts is essential to identifying strategic supports teachers feel they need to be successful.

In conclusion, more in-depth analysis and research into teachers' experiences can assist schools and local educational agencies identify best practices in supporting elementary educators who teach computer science concepts. Increasing access for students by expanding computer science teaching and learning will require considerations for professional development topics such as building teacher efficacy and creating sustainable and engaging learning environments. Support staff and instructional coaches can also play critical roles in assisting teachers, but more research is needed to identify how to best utilize such roles to improve current and future programs.

I utilized a case study with the intent to identify relationships and common elements of barriers elementary teachers experienced trying to teach computer science and the strategies they used to overcome those barriers. This case study may be helpful for local educational agencies planning professional development for elementary teachers, as well as for schools of education creating classes for preservice elementary teachers. Computer science education and the impact on elementary teachers is almost nonexistent as the majority of previous studies have focused on middle and high school teachers. This research is necessary as the literature review demonstrated a heavy focus on secondary teachers and their experiences teaching computer science.

Chapter 2 Summary

Computer science education and the professional development for it are emerging fields within local educational agencies across California. An increased emphasis on creating and adopting state computer science standards and global competitiveness is ramping up stakeholder advocacy for instructional time, curricula, teacher training, and course offerings. Schools and local educational agencies are making computer science education a critical part of instruction and learning. Teachers face challenges of learning computer science concepts, while identifying best practices to improve pedagogy. The literature review indicated the majority of studies focused on secondary teachers and their experiences. Computer science courses traditionally are offered in high school, which is why research is skewed towards upper levels of school; however, an increase in recognizing the need to prepare all students for the future is pushing down demand for computer science education to middle and elementary levels.

Beyond local school board priorities, states are beginning to emphasize the need for computer science education. California completed an adoption timeline for the state’s first computer science standards with the Instructional Quality Commission and State Board of Education. The plan highlighted the key dates to adopt standards and recommend an implementation plan for California and the local educational agencies in the state (California State Board of Education, 2018). Consequently, demand for computer science learning may increase at secondary levels, requiring foundation-building at elementary schools. For this reason, I will focus on elementary teachers and their experiences teaching computer science in a large urban local educational agency. A comprehensive review of the literature and previous research indicated a need for further exploration of elementary teachers’ experiences that was covered in this study. Thus, this study was designed to answer two research questions:

RQ1: What barriers do elementary teachers experience in teaching computer science to students?

RQ2: What strategies do elementary teachers develop in their experience teaching computer science to students?

Chapter 3: Methodology

In this study, I utilized a qualitative case study to gain an understanding of the experiences of elementary teachers in a large local educational agency who teach computer science. As elementary teachers prepare students to compete in a digital economy, a better understanding of their experiences was necessary. Expanding computer science education requires new opportunities for professional development and teacher preparation, which can be guided by teachers' experiences. Furthermore, literature on research in computer science education indicated a heavy focus on secondary teachers' experiences. As such, qualitative studies, particularly at K–6 levels may help form best practices for elementary teachers as they attempt to teach computer science concepts. Yin (2014) recommended a case study design when the focus of the study is to answer "how" and "why" questions.

Research is a scientific process that requires a methodology. The methodology is the foundation for the steps in research to gather and analyze data (Creswell, 2013). Data from this research helped form an understanding of elementary teachers' experiences teaching computer science. In particular, the data produced findings that revealed recommended supports that elementary teachers felt they needed to address the unique challenges they faced with computer science education. This chapter explains the components of the research methodology, design, and justification for the chosen methods.

Chapter 3 begins with the research questions, followed by a description of the purpose and design of the study. The research population, sampling method, instrumentation, and data collection procedures for the study are explained. I then discuss the data analysis procedures, limitations of design, validation procedures, and expected findings. The chapter concludes with the ethical issues of the study and a summary. Each of the components in the chapter contributes to the framework of the study to better understand elementary teachers' experiences teaching computer science.

Research Questions

To better understand elementary teachers' experiences in teaching computer science, research questions were designed to identify barriers in teaching computer science and the strategies developed to overcome those barriers. The research questions were as follows:

RQ1: What barriers do elementary teachers experience in teaching computer science to students?

RQ2: What strategies do elementary teachers develop in their experience teaching computer science to students?

Purpose and Design of the Study

Qualitative research through the case study approach allows for an in-depth exploration of perspectives and experiences (Creswell, 2013). Yin (2014) recommended case studies as a tool for investigation within a real-world context. In the case study approach, the case can be understood as an entity, that is, the teachers participating in this study. The data collected in this qualitative study included two interviews with teachers and collected documents.

As the purpose of this study was to gain an understanding of the experiences of elementary teachers in a large urban local educational agency who teach computer science, I utilized a qualitative design because it is inductive and allowed me to uncover themes and understand the perspectives of individuals living through the experience (Creswell, 2013; Stake, 1995). Creswell (2013) noted the emergent nature of qualitative research as a primary strength in helping identify phenomenon. I did not pursue a quantitative approach because quantitative research methods have a limited ability to probe answers as data is in numeric form (Creswell, 2013; Stake, 1995). Quantitative data could indicate hours teachers logged into a resource or number of units they completed, but such data fails to answer "why" and "how" questions. Surveys gathering feedback from participants was another quantitative option, but the approach could limit the breadth and depth of inquiry in this study.

Case studies as a research strategy are distinctive in helping researchers understand complex social phenomena of real-life events (Yin, 2014). With the recent adoption of California state standards for computer science education, demand for elementary teaching and learning in computer science will likely impact educators all across the state. Thus, a case study research design was appropriate for this study as teacher experiences in elementary schools have not been reported by multiple researchers.

I focused on teachers' experiences from one school in a single local educational agency, utilizing rich, thick description of the group. Thus, this study was a qualitative case study, as the approach facilitated a deep understanding of teachers' experiences from the site. Focusing on a single school allowed for depth of understanding in the experiences of the study participants (Stake, 1995). Due to the recent emphasis on computer science education and velocity of adoption across the state, the case study may not allow for building theory. However, findings may help schools with implementation plans and provide recommendations for professional development and support systems at elementary levels.

Research Population and Sampling Method

Qualitative researchers must understand and anticipate pivots during their study. Marshall and Rossman (2011) explained that qualitative research allows for evolutions in design during the research. Thus, creating a base of study through the selection of participants and site will be critical elements to beginning exploration and investigation.

Population

This study took place in a large unified school district located in California. Pseudonyms were used for Inland Unified School District (IUSD), as well as the participating schools and participant names. IUSD is a large K–12 urban unified school district servicing a large swath of Inland County. There are approximately 50 schools in the district, 30 of them being K–6 elementary schools. The schools serve 43,000 students and employ approximately 1,900 educators. Priority for computer science education was declared by the school board last year. The board's target population included all K–6 teachers from the elementary schools in IUSD teaching computer science. As such, elementary teachers from a single school were purposefully invited for this study (see Appendix A).

Sample

Purposeful sampling was used in this study. Marshall and Rossman (2011) recommended utilizing purposeful sampling when the researcher requires access to a specific subsection of participants whose insight is required for the study. Although purposeful sampling may leave gaps in information (Creswell, 2013), the rich data provided by the selected participants can provide ample content for the study.

Participants in this study were elementary teachers from Abraham Lincoln Elementary School in IUSD. The characteristics of the sample included their years of experience, state-approved credentials, grade levels, and potentially, Google Educator Level 1 and Level 2 certifications. IUSD is a Google for Education school district with a comprehensive roll-out of Chromebooks and the suite of Google apps. Thus, teachers can apply for Google Educator certifications that demonstrate their knowledge and usage of technology. Focus on this study emphasizes understanding the experiences teachers have teaching computer science as instructor anxiety was one of the largest identified barriers to providing students a meaningful introduction to computer science (Krauss & Prottsman, 2017).

Participant selection was based on access granted by the principal and participants' previous experience with computer science professional development (see Appendix B). Purposeful sampling allowed for efficient use of limited resources as participating individuals were knowledgeable about the phenomenon of interest, in this study, their experiences teaching computer science (Creswell, 2013; Patton, 2015). Thus, teachers were recruited from a population of those who have received one year of exposure to computer science curricula either through professional development or conferences. IUSD utilizes an online professional development hub that allows teachers to register and sign up for events and trainings. I recruited teachers from Abraham Lincoln Elementary School who attended events tagged with keywords "computer science." I selected Abraham Lincoln Elementary School, because this school had the highest number of elementary teachers who received exposure to computer science. I purposefully selected 10 teachers in this case study to participate in interviews and document collection. The sample size was large enough to obtain feedback for most perceptions, since Guest, Bunce, and Johnson (2006) proposed saturation occurs with over 10 participants.

Instrumentation

Case studies offer diverse approaches to instrumentation and data collection, with a wide variety of available evidence (Creswell, 2013). This research study consisted of multiple sources of data through two sets of semistructured interviews with each participant as well as collected documents. I used the instruments to triangulate the data, providing validity to the findings (Creswell, 2013; Stake, 1995).

Documents

In a case study, documents can take a variety of forms (Creswell, 2013). For this study, I gathered and reviewed public articles, as well as district and school documents related to computer science initiatives and policies. I also requested documents from the participants in the form of lesson plans, meeting notes, and agendas related to computer science. Yin (2014) noted documents can corroborate and support evidence from other sources, potentially providing insights that cannot be obtained through interviews or observations. Even though they may be written for purposes and audiences outside of the case study being conducted, documents can provide insight into the study (Yin, 2014). Document review also provided historical context via background information on the current status of elementary teachers' experiences in teaching computer science. Thus, documents collected in this study provided data that may not have been observed (Stake, 1995).

Semistructured Interviews

Interviews provide valuable insight into participants' experiences. Interviews can take the form of highly-structured, semistructured, or unstructured protocols (Merriam, 2009). Interviews can also uncover and portray multiple views of the case. According to Stake (1995), gathering and analyzing participants' views and experiences are essential as "the interview is the main road to multiple realities" (p. 64). Additional benefits in utilizing interviews include allowing participants to have an opportunity to share events and insights that are not directly observable (Creswell, 2013). Lastly, interviews are interactive and the collaborative nature presents opportunities to probe for complete and clear answers. In this study, I conducted two interviews to gain a thorough and comprehensive insight into teachers' experiences.

The interviews were semistructured, allowing participants to expand and elaborate on their responses (see Appendix C). Structured interviews do not provide as much flexibility for probing questions and unexpected conversation (Creswell, 2013). Thus, I used semistructured interviews in both phases as opportunities to explore unexpected topics unrelated to the scripted questions that arose during the interview process (Gill, Stewart, Treasure, & Chadwick, 2008).

Secondary Semistructured Interviews

Yin (2014) suggested two levels of interviewing to build rapport, connecting with participants and collecting data for research. Conducting two levels of interviews with the participants allowed me to better connect with participants, build relationships and receive authentic responses to the questions (Yin, 2014). Establishing a relationship through multiple interviews allows participants to feel their opinions are valued and their responses are recognized (Kvale & Brinkmann, 2009).

I scheduled and met with interview participants in two phases to gather additional evidence. The 2019 school year was the first year teachers in California could teach computer science with state-adopted standards. I conducted the initial interview, analyzed the data, and then followed up for the second round of interviews (see Appendix D). Considering the significant demand for elementary teachers to teach all subjects, interview responses within two phases allowed the participants to experience additional real-life scenarios in their classrooms. The time between interviews allowed for increased opportunities for interactions and social situations in the classroom, which impacted some teachers' perceptions (Creswell, 2013).

Data Collection

Data for this research was collected from elementary teachers who were teaching computer science. Significant progress made in the adoption of standards and the subsequent urgency for computer science education aligned well with this case study, as the circumstances allowed for research in which some phenomenon was presently happening (Yin, 2014). Bernard (1995) offered four goals of qualitative research: exploration, description, comparison, and testing models. To that end, qualitative data can be collected in various ways. In this case study, the tools were documents and two interviews.

Following Institutional Review Board (IRB) approval from Concordia and the school district, I informed the principal and contacted each of the teachers to schedule a meeting to go over the background of the research and provide an opportunity to ask questions about the study. During the initial meeting, participants received informed consent forms (see Appendix E) and I scheduled interview appointments with participants who agreed to participate in the study. The interviews were scheduled based on time and location preferences set forth by the participants.

In addition to collection data through two phases of interviews, I also asked participants to share relevant documents for this study.

The data reflected the emic perspective of participants. Merriam (2009) explained the emic perspective represents the internal language of a group. Yin (2010) stated, “an emic perspective attempts to capture participants’ indigenous meanings of real-world events” (p. 11). In this case study, the scope of culture was based on a small group of individuals who shared the common characteristic of teaching computer science in K–6 grades.

Documents

For this study, I collected multiple document sources. On a site level, I requested teacher-created documents such as curriculum guides, lesson plans, and meeting notes (see Appendix A). I sent out a reminder via email requesting the documents prior to the first interview. Teachers were able to provide documents they wanted to share at the initial interview meeting. At the conclusion of the first interview, I reminded participants they could provide additional documents in the second interview. Teachers were able to submit documents at each of the scheduled interviews.

Mail between school sites and other departments are sent and delivered internally through the district as a team collects and delivers the envelopes. Participants received a large envelope they could use to send and receive district mail. Teachers were able to share documents in the form of lesson plans, curriculum guides, meeting notes, and other artifacts that they felt would benefit the study. Teachers were able to send any documents they felt would be helpful through district mail when they felt the content was relevant to the study.

Lastly, I searched for and collected school and district records specific to professional development, programs, and initiatives available through public domain. These documents were available on the district and school websites as well as online publications and other websites. I utilized a strategic Google search features using keywords I used for the literature review.

Semistructured Interviews

I met with each participant to gain a deeper understanding of teachers' experiences regarding computer science education. Each interview took 45 minutes to one hour. All interviews were arranged to accommodate participants' schedules and were conducted at locations they selected. Participants were provided the list of interview questions before the meetings, allowing them to reflect and consider their responses (Gibson & Brown, 2009). These steps were essential to developing and maintaining a comfortable and relaxed climate for the participant.

I scheduled a meeting with each participant, making sure to accommodate their schedule and preferred location. Some participants felt more comfortable meeting outside of their school site, and it was important to meet such requests (Kvale & Brinkmann, 2009). I asked each participant 12 questions (see Appendix C). The first six questions allowed me to gain a better understanding of each participant's background in education and possible motivations for teaching computer science. The seventh and eighth questions gave teachers an opportunity to share their experiences teaching computer science. Questions 9 and 10 encouraged participants to share any discoveries with challenges they faced and instructional strategies they developed in response to those challenges. The final two questions allowed participants to include additional insight into their experiences teaching computer science that may not have been uncovered by the initial questions.

I recorded all interviews via a password-protected mobile recording device while taking notes when the interviewee was talking. The audio interview files were uploaded to a private Google Drive account. The files were also downloaded to a password-protected external hard drive to secure multiple duplicates of the original content. Audio files were transcribed utilizing Google's Voice Typing Tool with transcripts being converted to Google Docs for data analysis. Immediately following transcription, I followed IRB policy and deleted recordings on the Google Drive account and external hard drive. I am keeping the transcriptions on a password-protected external hard drive that is stored in a locked cabinet in my office. These documents will be deleted three years after this study's completion.

Secondary Semistructured Interviews

I scheduled the secondary semistructured interviews after the first phase of interviews was completed with all participants. I followed the same protocol as the first phase of interviews, and scheduled each interview to accommodate times and locations based on participants' preferences. The questions during the secondary semistructured interviews allowed teachers to share how their experiences may have changed during the study. Questions for the secondary semistructured interviews were generated based on the data analysis findings from the first interview (Hatch, 2002). I followed the same data collection protocol from the first interview process and utilized the password-protected mobile device for voice recordings. I followed IRB policy and destroyed the recordings as soon as the data was transcribed. I will keep the password protected transcriptions from the secondary interviews and will delete them 3 years after this study.

Each piece of collected information from the multiple data sources acted as a smaller piece of the larger puzzle. Smaller elements contributed information and findings, helping build an understanding of the phenomenon. Ultimately, combining the multiple pieces helped build coherence and convergence within the data to generate a better understanding of the phenomenon.

Data Analysis Procedures

Data analysis procedures focused on interpreting and identifying meaning from the rich data collected in the case study. The case strived to capture the richness of the study with a foundation from stories and experiences that surfaced from collected data (Grandy, 2010). Conducting data collection and analysis through the multiple phases allowed me to consistently keep a lens on common findings as well as outlier findings. Furthermore, utilizing the multiple sources of data maintained a chain of evidence (Yin, 2014). Gillham (2000) recommended a similar approach, emphasizing the need to use multiple sources of evidence because "all evidence is of some use to the case study researcher: nothing is turned away" (p. 20).

A process of discovery reveals categories in the data, patterns, and relationships. Qualitative data analysis is inductive and malleable as there are no predefined measures for capturing the phenomenon (Schutt, 2018). Data for this study were analyzed through an inductive approach. Transcripts from the interviews, collected documents, and researcher notes provided context into the subthemes. I maintained a record of noticeable data and collapsed similar findings to reduce overlap in the data. The collected codes were named and collapsed into a key number of subthemes. The subthemes were grouped based on their relationship with the data to identify the main themes. The inductive analysis approach emphasized searching for patterns in qualitative data (Stake, 1995). Working back and forth between themes and the collected data allowed me to generate knowledge and increase understanding of the data during the analysis process (Creswell, 2013). I was able to find relevant meaning in the collected data by going through the inductive analysis approach.

Primary Semistructured Interviews

Interviews were transcribed and analyzed to find research findings from the significant themes in the raw data (Thomas, 2006). Using an inductive analysis approach, I categorized the collected codes through frames of analysis and identified subthemes. This process allowed me to interpret the data and develop naturalistic generalizations and establish common themes (Hatch, 2002). The established subthemes were reduced to the main themes based on elementary teachers' experiences teaching computer science. Throughout the analysis process, I was mindful about keeping a lens on common findings while looking for outlier findings in the data.

Prior to reading and analyzing the transcripts, I prepared and managed all data files. All text was modified into a common format with consistency in file type, font size, margins, and specific highlighted segments. Once all transcripts were properly formatted, I started my data analysis.

I conducted an initial reading of all transcripts to become familiar with the content (Saldaña, 2015). Close reading allowed me to gain a better understanding of possible details in the text data. I wrote down notes about my impressions and carefully focused on participants' responses**.** I established three frames of analysis: (a) experiences teaching computer science, (b) factors impacting teaching and learning computer science, (c) motivations for teaching computer science. I separated the participants' responses by color based on my frames of analysis which further helped with organization of the data. I reviewed participants' responses and highlighted key ideas and recurring terms. Meaningful segments included words, phrases, sentences, or sections from the transcripts. These data were used to identify the relevant text segments that were charted on a large whiteboard as well as the digital version within a Google Doc. The color codes provided additional visual cues during data analysis. Individual data on the whiteboard were referenced with related responses and interview identification numbers. This helped reveal overlap between participants' comments and also helped identify word and phrase repetitions to create relevant codes.

I reviewed the codes and looked for redundancy and relationships between the collected data. I combined relevant codes and grouped them together into the most relevant categories (Creswell, 2013; Thomas, 2006). Next, I developed an initial description of the meaning of each category. Throughout the analysis process, I continued to review the data to create, combine, and reduce categories by identifying relationships and possible overlap.

I also considered alternative understandings throughout the data analysis process before writing down the findings (Marshall & Rossman, 2011). Outlier information was listed on the whiteboard as those data findings emerged in the interviews and were considered as possible additional subthemes. Finally, I interpreted the meaning of the data, considered alternative findings, and then wrote down the findings (Marshall & Rossman, 2011).

Secondary Semistructured Interviews

Data analysis in the first round of interviews uncovered potential themes in this study. I reflected on the data and formed additional questions to gain a deeper understanding of data and participants' perspectives (Hatch, 2002). I utilized the same data analysis process for the second phase of interviews. I identified the frames of analysis, relevant codes, and searched for relationships among the data. I continued my analysis to interpret the collected data, analyzed the codes, and identified the themes. I was able to confirm the significance of the themes identified during my analysis with data supporting my findings. Throughout my research, I followed a standard protocol of data analysis for both interviews to help maintain validity in this study.

Documents

I utilized the typological analysis recommended by Hatch (2002) to analyze the collected documents. I gathered and analyzed multiple sources of data throughout the study. All provided material included team meeting notes, lesson plans, agendas, and public domain articles. In preparation for data analysis, the collected documents were printed. I set four typologies: (a) computer science pedagogy, (b) computer science goals, (c) approaches to computer science, and (d) interdisciplinary connections. I assigned each typology a color and started the first review of the documents to gain a comprehensive understanding of all the elements. I reread the data and began highlighting and coding any information. I looked for patterns, relationships, and possible themes that were revealed to the typologies. I made sure to highlight and note critical items related to the research questions. I was able to write an automated program using the script editor within Google Docs (see Appendix F). This allowed me to use loops and functions written in JavaScript to code and highlight essential items on a digital summary sheet. Using a computer program allowed me to collect and create a visual picture of codes and themes (Creswell, 2013). I created a matrix and searched for patterns between the codes. The findings revealed emergent themes and I referred back to the documents. I identified data examples to support the emergent themes and wrote down one-sentence generalizations. I went through the collected data for a final review and selected data excerpts that supported the written generalizations.

Limitations of the Research Design

Researchers must be transparent about limitations in their study as an individual case study would not produce an all-encompassing conclusion (Creswell, 2013; Marshall & Rossman, 2011). Furthermore, awareness and acknowledgment of limitations in the study will help guide measures to maintain reliability and validity. In this study, I identified three limitations: (a) limited time to conduct the study, (b) research centered on one school district, (c) and self-reporting of participants' experiences and needs in teaching computer science. This case study emphasized the experience of a group of teachers in one part of the country who are implementing computer science education at elementary levels. The conclusions may not entirely be transferable due to the large number of elementary school teachers, the various range of needs and challenges in different grade levels, and administrative priorities within individual districts.

Fully understanding the limitations of the research design allowed me to highlight the impacts of this case study. I was able to gather participants through purposeful sampling, producing meaningful, information-rich data (Patton, 2015). Furthermore, the intimate scope of the study allowed me to build connections with participants and gain a deeper understanding of needs and challenges these elementary teachers faced with computer science education.

Delimitations of the Research Design

I created intentional boundaries in this case study. The choices I made helped ensure the research was completed within a reasonable time and was sensitive to the needs of the participants as well as available resources. In this study, I identified three delimitations: (a) the sampling of participants who had some exposure to computer science, (b) research centered on one school site, (c) and the selection of a school prioritizing computer science education.

The sampling of participants who had some exposure to computer science allowed for purposeful sampling as teachers without exposure were not considered for participation in the study. I also selected the questions for the interviews. Consequently, I was able to focus on a manageable sample size allowing me enough time to go through all data collection and analysis. Finally, confining the study to one school helped me focus on developing deeper understandings with each participant in a consistent setting.

Validation

This case study investigated the challenges participants experienced teaching computer science and the strategies they developed to overcome challenges. To improve the trustworthiness of qualitative research, Yin (2014) and Merriam (2009) recommended various constructs and strategies. In this case study, I maintained validity by emphasizing constructs of credibility and transferability.

Creswell (2013) defined validation as “the closeness of the researcher to participants in the study” (p. 250). Throughout the case study, I was in constant contact with participants. I maintained validity in multiple ways including: (a) triangulating multiple data sources; (b) maintaining careful attention to conflicting information; (c) producing thick, rich description (Bernard, 1995; Creswell, 2013). Furthermore, Seidman (2006) recommends an approach to establishing validity by checking participant consistency through multiple interviews. This was accomplished in the two data collection phases with the primary and secondary semistructured interviews. Yin (2010) recommended establishing construct validity through the use of multiple types of data, creating a chain of evidence, and having participants review transcripts. All three strategies were applied during the data collection and analysis phases.

Credibility

Credibility is dependent on the researcher's ability to investigate and analyze realities constructed by participants in the case study (Merriam, 2009). Recognizing the importance of matching participants' perceptions with my portrayal was an embedded consideration throughout the study (Bloomberg & Volpe, 2012). I utilized multiple strategies to ensure credibility in this study through: (a) purposeful reflection and monitoring of researcher bias by continually re-evaluating impressions of participants and challenging pre-existing assumptions, (b) clearly defining the research process, (c) triangulating multiple sources of data, and (d) conducting member checking by paraphrasing interview responses back to participants (Bloomberg & Volpe, 2012). The strategies helped support the credibility of this study to ensure the research findings matched reality.

Transferability

Transferability is possible when the case study produces rich, thick description of all components and findings of the research. Creswell (2013) highlighted the benefit of detailed descriptions, allowing researchers to transfer information to other settings and determining if the findings can be transferred. This study provided a thorough description of the context of the study, participants' barriers to teaching computer science, and the strategies they developed to overcome challenges. Direct quotes from participants provided an authentic view of their experiences and allow a future researcher to apply similar protocols when conducting a study in other settings. Thus, using thick and rich descriptions of the research process will assist future studies if other researchers feel secure enough with the transferability of the case study.

Expected Findings

Comprehensive data were collected during this case study. I analyzed the data to understand elementary teachers' experiences teaching computer science. Thus, the data revealed possible opportunities to identify entry points for various types of support that elementary teachers may need to teach computer science. I expected to find a level of professional development that was significantly exceeded by a need for content. In this context, teachers had to assume the role of a learner and student as they developed enough knowledge of computer science concepts to be able to implement and teach effectively. Furthermore, I expected to find external factors such as pressure from parents to teach computer science coexisting with potential conflicts from administrators' priorities. The results of this case study may add to the existing literature on the experiences and challenges of computer science teachers while adding the unique perspectives of elementary teachers.

Ethical Issues

I exercised the proper recommended practices and protocols for addressing possible ethical issues. I followed the ethical principles recommended by The American Psychological

Association (APA, 2010) of beneficence and nonmaleficence, fidelity and responsibility, integrity, and respect. Furthermore, I followed the standards set by Concordia University's Institutional Review Board (IRB) and the school district's approval board to further mitigate potential ethical issues. Pseudonyms were used for the participants, further maintaining confidentiality. Full disclosure of my position statement, consideration of ethical issues, and transparency of procedures and analysis additionally helped support credibility of collected data and the subsequent conclusions drawn through data analysis.

Conflict of Interest Assessment

Throughout this study, I carefully and explicitly identified strategies and measures to address conflicts and maintain absolute transparency with participants. Participants signed consent forms (see Appendix E) and received a full explanation of the background and requirements of the study. Participants also had complete flexibility and autonomy in their participation in the study and were allowed to opt out at any time. Furthermore, participants received a full report of the collected data and a draft of the final dissertation before it was submitted. Consequently, I did not anticipate a conflict of interest in conducting or reporting the results of the research.

Researcher's Position

At the time of this study, I was an employee of the case study school district. I did not hold a site position and thus was not be able to give directives to teachers on what they can or should teach in their classrooms. I held no authoritative influence on participants in the study. Furthermore, I started with the same knowledge of computer science concepts as the participants but enrolled in a supplementary credentialing program to better understand and develop content knowledge.

Ethical Issues in the Study

I received approval from Concordia University's Internal Review Board. I also provided participants with complete and transparent information about the case study and allowed them to leave at any time. Participants were given a full understanding of expectations for participating in the study, the reasons for this research, and the process I used to conduct the study (Creswell, 2013). Furthermore, all voice recordings were destroyed once they were transcribed, and the data were analyzed. At no time were participants put in a position of harm or wrongdoing, and pseudonyms were used to further protect their identities.

Chapter 3 Summary

The release of statewide computer science standards and increased adoption of computer science across elementary schools has created unique challenges for educators. Teachers with multiple-subject credentials are attempting to teach an entirely new concept with limited support resources. Furthermore, the majority of research on teachers' needs have been limited to secondary experiences. Significant differences in teaching and learning from elementary to secondary perspectives necessitated an urgent need for further research into elementary teachers' experiences. The research questions were designed to understand elementary teachers' experiences teaching computer science in a large urban district. I gathered data through three phases and identified themes through coding and data analysis strategies shared in this chapter. The protocols allowed me to triangulate data to ensure validity and form a deeper understanding of elementary teachers' experiences with computer science education.

Chapter 4: Data Analysis and Results

This qualitative case study was designed to identify the experiences of elementary teachers teaching computer science in a large urban school district. The case study enabled me to explore and develop an understanding of the experiences elementary teachers had teaching computer science (Hatch, 2002). Ten elementary teachers participated in this study, providing input of their experiences teaching computer science through interviews and documents. This study addressed two research questions:

RQ1: What barriers do elementary teachers experience in teaching computer science to students?

RQ2: What strategies do elementary teachers develop in their experience teaching computer science to students?

In this chapter I present a complete description of the sample that was used for this qualitative case study as well as the research methodology. A complete description of each participant provides context into their backgrounds. The chapter continues with the research methodology and analysis of data. Data for this study were collected from multiple interviews and documents. I present the results of the data and end the chapter with a summary of the findings.

Description of the Sample

I invited 12 elementary teachers teaching computer science from Abraham Lincoln Elementary School to participate in this qualitative case study. The 12 elementary teachers were selected because they received exposure to computer science professional development the previous school year. Of the 12 teachers, 10 decided to participate in the study. All of the teachers hold multiple subject credentials, an authorization to teach all subjects in an elementary setting. The teachers in the study also received professional development to teach computer science. Two teachers currently hold supplemental authorizations and two teachers are working on obtaining a supplementary authorization. Total years teachers had been teaching ranged from four to 26 years. All participants reported going through computer science professional development either through conferences or district- provided training sessions.

Description of Participants

This section provides a comprehensive description of the participants in this study. Table 1 presents an overview of all of the participants, their assigned pseudonyms, their current grade level, number of years they have been teaching, and information about additional credentials they possess or are working on adding to their licensures. All participants possess a multiple subject credential.

Jefferson

Jefferson has been teaching for 21 years. All 21 years have been in IUSD, with 20 years in one school. This year he transferred schools to Lincoln and has always tried to integrate technology in the classroom. In the last few years, he was working on incorporating personalized learning elements at his previous school, acting as the grade level lead and site technology mentor. He believes students should have a voice and choice in the way they learn and he utilizes a form to gauge their learning preferences. He works to adapt instruction based on students' preferred learning methods and utilizes station rotations as much as possible. He does not believe in a full day of direct instruction as that would inhibit him from providing personalized learning experiences. Jefferson currently consults for a few educational technology companies and shares his experiences with other educators who use those products. He also teaches a class on instructional technology at the local community college.

Table 1

Overview of Participants

Pseudonym	Grade Level	Years of Experience	Supplementary Authorization
Jefferson	4th	21	
Rachel	2nd	14	Mathematics – In Progress
Lisa	6th	4	Mathematics and Science – In Progress
Deborah	6th	26	Computer Science
Ellison	4th	20	Computer Science
Maria	3rd	18	Administrative Services
Heather	3rd	7	
Claudette	5th	20	
Pamela	1st	12	
Keisha	Kindergarten	21	Language Arts

Rachel

Rachel became a teacher because she wanted to help all students succeed, particularly in math. She also wanted to become an educator because she felt there are not enough Asian American women who are teachers. Rachel's area of expertise is in elementary mathematics and she has been teaching for 14 years. She also spent three years working as teacher on special assignment, focusing on professional development in elementary mathematics. Two years ago, Rachel returned to a school site and is currently teaching second grade. She is also in a program at the local university to obtain a supplementary authorization in mathematics. Once she completes the program, she will begin looking for opportunities at a middle or high school in the district.

Lisa

Lisa decided to pursue teaching as a second career. She previously worked as a project manager for a geographic information system software provider. In her first career, she worked closely with technology and brings forth her experience as a classroom teacher, looking to find meaningful ways to integrate instructional technology. This is her fourth year in the classroom and she holds a multiple subject credential. She is also working on obtaining her supplementary authorizations in science and math. Once she completes the program she will be able to teach middle school and high school science and math courses but she is not sure when she would leave her elementary setting. Her current reason for pursuing the supplementary authorizations is to improve her background knowledge and pedagogical strategies for teaching those subjects in elementary school.

Deborah

Deborah is a veteran teacher who is a couple years away from retirement. She has taught for 26 years but is teaching for the first time in IUSD. In her former district, middle school consisted of students in Grades 6 through 8. This is her first year teaching at an elementary school. She is one of the few teachers who holds a supplementary authorization and bachelor’s degree in computer science. Deborah considers herself to be on the cutting edge of technology and is always researching and exploring new devices and software for personal and professional use. She enjoys finding ways to integrate technology and teaching computer science in a middle school model. This is her first year exploring how to teach computer science in an elementary setting with challenges stemming from her recent change.

Ellison

Ellison always wanted to be a teacher. She knew from an early age that she wanted to work with students and started her career very early on. She has taught a wide range of grade levels at elementary levels and has spent all 20 years of her career in IUSD. This year, she received an offer to work as a teacher on special assignment, focusing on elementary mathematics. The district adopted new math curriculum for elementary grades and decided to offer support by having two teachers on special assignment who design and deliver professional development. Ellison is one of the two teachers who was selected for this position. In addition to coteaching her classroom with the other teachers on special assignment, she is receiving training to facilitate math trainings. She holds a multiple subject credential and recently completed the coursework for a supplementary authorization in computer science. The supplementary authorization allows her to teach computer science courses in middle school and high school. She has no prior experience teaching at secondary levels but now has the option to do so with the additional credential.

Maria

Maria has been teaching for 18 years. She has spent her entire career in IUSD and is currently teaching third grade. She particularly enjoys lower grades but is flexible enough to move across grade spans. Her ability to adapt has allowed her to teach from kindergarten to sixth grade in her 18 years. Education is a passion of her and she feels a constant need to learn new things professionally and personally. Maria is working on obtaining her administrative credential and is also pursuing a doctorate in education. She feels both of those programs are a part of her need to always feel she is growing and learning. Maria is extremely well-respected in the district as she has worked at multiple schools and served on numerous committees. Most recently, she participated in the language arts curriculum adoption committee and is involved in this year's mathematics curriculum adoption committee. She is also her school site's technology coach which is one of the many available adjunct duties.

Heather

Heather is relatively new to the district. She has taught for five years in IUSD, but has seven years of total experience. Her previous time was at a private school where the demands and responsibilities were noticeably different from public schools. Heather also has children who currently attend schools in IUSD. Heather believes her students must develop skills to be socially engaged contributors. She feels education should emphasize content and technical standards as well as skills to collaborate and communicate with others. Heather believes students learn these skills very early and she encourages students to understand and empathize with their peers.

Claudette

Claudette has spent over 20 years teaching in the district. She has only taught fourth and sixth grade classes but is teaching fifth grade for the first time. For one year she left the classroom and worked as a staff development specialist based at the district office. Before teaching computer science she had no experience but decided it was important to provide the opportunity for her students. She believes teachers should be willing to take risks and try new things in the classroom. Claudette loves creating magical learning moments and establishing community which is why she has spent so many years at the same school site. Her classroom is unique as she designed the environment to look like a living room. Walls are painted bright colors and two large couches take center place in the room. She feels her room might be one of the only places some of her students have access to a consistent environment, so she wanted the room to feel welcoming and warm.

Pamela

Pamela loves teaching younger students. Her first love is kindergarten and she considers that grade to be her area of expertise. However, she has recently taught first and second grade because she wants to know what students are expected to understand at those levels. She feels she will be able to better prepare kindergarten students and help them build a foundation if she has knowledge of standards and curriculum in the next couple grade levels. Pamela has spent seven years in IUSD with a total of 12 years of experience. She previously taught at a neighboring district but took five years off when she had children. Her children are older now and they also attend school in IUSD. Pamela emphasizes finding students' strengths and helping them apply themselves based on those strengths. She is a certified Gallup coach, and uses Clifton Strengths to identify students' themes and talents. She feels teachers have a responsibility to adapt to students' needs rather than asking them to change based on the teacher. This perspective also stemmed from her work at her previous school, where the staff focused on implementing personalized learning. Pamela feels creativity and wonder should be fostered in classrooms and she finds great joy when students discover more about their world.

Keisha

Keisha loves working with younger students. She has taught multiple grades during her 21 years in the district but her primary love is kindergarten. Keisha feels building strong foundational skills in kindergarten can help students succeed in later grades. She feels developing students' skills in fluency and reading comprehension are essential and holds a supplemental authorization to teach language arts. At one point she thought about teaching middle school language arts but decided to remain at the elementary level. Keisha wanted to pursue computer science in college because of her curiosity with the computer her father brought home. However, she moved away from studying computer science due to her negative experience in college as the only African American female student in her class. Based on her experience, she wants all students to have opportunities in computer science which is why she strives to teach it at a kindergarten level.

Research Methodology and Analysis

In this qualitative study, I utilized a case study to gain a better understanding of elementary teacher's experiences teaching computer science. The case study was designed to describe the experiences of elementary teachers teaching computer science within the context of real life (Yin, 2003). Two questions guided this study:

RQ1: What barriers do elementary teachers experience in teaching computer science to students?

RQ2: What strategies do elementary teachers develop in their experience teaching computer science to students?

Data collection and analysis included semistuctured interviews, secondary semistructured interviews, and documents. Analyzing the semistructured interviews in this study, I followed the inductive analysis model, moving from the specific data to a search for patterns (Hatch, 2002). I utilized typological analysis to analyze the documents I collected for this study (Hatch, 2002). I allowed participants to read and review interview transcriptions as well as the analysis reports. They were given an opportunity to ask clarifying questions regarding the data and review the identified themes. Participants did not find any need for revisions. Following this process for member checking ensured participants' perspectives and experiences were shared transparently and accurately (Lincoln & Guba, 1985). In the following section, I explain the steps I followed in collecting data for this study.

Data Collection

This study required three phases of data collection. I utilized two interviews as well as documents in the form of lesson plans and meeting notes. I met with all participants and held semistructured interviews. Data from those interviews were analyzed before scheduling the secondary semistructured interview.

Semistructured Interviews

I utilized interviews for this study because they allow researchers to explore participants' experiences (Hatch, 2002). The first set of interviews occurred over a period of a week. Ten participants agreed to participate in the study and meet for interviews. Each interview was scheduled to best accommodate participants' preferences on time and location. The majority of interviews occurred in participants' classrooms except for three. Two of these interviews took place in a coffee shop near the school site after school. The third interview took place in a fast food restaurant on a Saturday. This was the only interview that did not occur during the work week.

In the first set of interviews, I asked each participant 12 questions. Questions 1 through 7 were structured to give participants an opportunity to provide information about their backgrounds in education and possible motivations for teaching computer science. Question 8 was designed to gather information on participants' experiences teaching computer science. Question 9 was designed to encourage participants to share discoveries with challenges they experienced teaching computer science. Question 10 asked participants to share instructional strategies they may have developed in response to their answers from the previous question. Question 11 was designed to allow participants to discuss the types of support they felt they needed based on their experiences teaching computer science. Question 12 allowed participants to provide additional insight into their experiences teaching computer science that may not have been discussed in the previous questions. I recorded all interviews and promptly transcribed them. Each interview was transcribed on a Google Doc which was then converted into a PDF format.

Secondary Semistructured Interviews

Following the first round of semistructured interviews, I moved into the analysis phase and looked through all the transcripts. After analyzing the data, I formed additional questions that I asked each of the participants during the second round of interviews. The additional questions were designed to provide participants an opportunity to clarify and elaborate findings from the initial data analysis results (Hatch, 2002).

I prepared the questions and submitted them to participants prior to their secondary interview appointment. Sending participants the interview questions provided them time to consider their responses. The interviews took place over a period of two weeks at locations and times selected by the participants.

Questions 1 and 2 asked teachers to identify and expand on the challenges they experienced teaching computer science and the supports they felt they needed to address those challenges. Question 3 asked teachers to discuss the impact on computer science curriculum and resources and the way it impacted their experiences. Question 4 revealed teachers' opinions about the interdisciplinary connections in computer science. Question 5 encouraged teachers to expand on their perceptions of administrator support for computer science education. Question 6 asked about teachers to discuss the importance of building awareness in computer science among stakeholders. Question 7 allowed teachers to openly share any information they felt was not included as part of the questions in the interviews. I followed the same process as the first semistructured interviews. I voice recorded each interview and transcribed them on a password-protected Google Doc.

Documents

At the end of the first semistructured interview, I each asked participant to share lesson plans, curriculum guides, and meeting notes related to computer science. I provided each participant a large envelope they could use to mail me documents using district mail services. I followed up on the request via electronic mail and allowed participants to submit documents throughout the study. Some participants chose to scan and send the files via their district Google Drive accounts or via electronic mail.

To gain a better understanding of computer science education beyond classroom perspectives, I utilized other documents. I searched for publicly available documents related to computer science in the district. I found resources on the district website and found articles written about the school district's work in computer science. The additional documents helped me gain a broader understanding of teachers' experiences with computer science.

Data Analysis

I utilized a combination of data analysis procedures to analyze the collected data. I followed the inductive analysis model to analyze the primary and secondary semistructured interviews and typological analysis for the collected documents (Hatch, 2002). The collected documents included lesson plans, meeting notes, curriculum guides, and publicly available documents related to teaching computer science in the district.

Using an inductive approach for qualitative data analysis allowed me to condense extensive and varied text data, establishing clear links, and uncover summary findings. Another benefit of the inductive approach was that research findings emerged from frequent and dominant themes inherent in the raw data (Thomas, 2003).

Semistructured Interviews

To begin the analysis of the primary and secondary semistructured interview data, I read through all transcriptions to check for accuracy. In addition to checking for accuracy, the initial sweep allowed me to get familiar with the data and the information participants shared in the interviews. I emailed the transcriptions to all participants and asked them to check for any errors or information I may have missed. Participants noted there was no need to make corrections to the transcriptions, allowing me to proceed with data analysis.

I began analyzing the data by preparing the raw data files to a common format by converting all text into a Google Doc. Once the text was prepared, I needed to become familiar with the content and search for details in text. I went through multiple, close readings of text which allowed me to consider multiple meanings within participants' responses. This allowed me to gain insight into preliminary dominant themes which helped form frames of analysis: (a) experiences teaching computer science, (b) factors impacting teaching and learning computer science, (c) motivations for teaching computer science. Reviewing the data through the frames of analysis helped identify possible relationships in the research data as I analyzed the data through specific lenses (Hatch, 2002).

Using the frames of analysis, I was able to identify relevant and meaningful data. I coded the data in accordance with the frames of analysis. I followed the inductive analysis process utilizing open-coding to examine and compare the collected data (Saldaña, 2015). I created and assigned codes to text segments that contained meaning such as actions, concepts, opinions, or differences (Saldaña, 2015). I used "lumper" coding to create codes to represent the meaning of phrases used by the participants in specific text segments. I identified 13 clusters of related codes that were analyzed and labeled with a corresponding category. I reduced the data by reflecting on each cluster and identifying any patterns that I may have missed. As I reread through the transcripts, I separated the codes that were relevant to specific subthemes and grouped them accordingly (Hatch, 2002). I identified 10 additional clusters for a total of 23 subthemes. These codes were considered upper level subthemes or general categories which needed to be collapsed into more specific categories (Thomas, 2003).

The next step was to reduce the collected data to find the emergent themes in this study. I needed to filter through and regroup overlapping categories, moving from upper level to lower level categories (Hatch, 2002). I looked for relationships via associations, links, and implications (Thomas, 2003). I continued revising and refining categories by searching for entry points to combine or link codes with similar meanings. I also searched for subtopics and contradictory points of view. This step reduced overlap and redundancy among the categories allowing me to identify the four main themes. The four emergent themes identified were: (a) elementary teachers experience external barriers through organizational structures that impact their ability to teach computer science, (b) elementary teachers experience internal barriers to teaching computer science such as their limited efficacy to teach a new content area they are not familiar with, (c) elementary teachers rely on coping mechanisms for external barriers to reduce the impact of organizational structures, (d) elementary teachers rely on a shift in mindset to address internal barriers.

Documents

I utilized the typological approach to analyze the collected documents. The typological approach was appropriate because the findings revealed distinct categories within the collected data. Participants were provided with envelopes to send their documents and were given the option to send them electronically. The collected data included lesson plans, meeting notes, curriculum guides, and publicly available documents related to computer science in the district.

The typological approach involves both deductive and inductive reasoning (LeCompte & Schensul, 1999). I began analyzing the data by setting typologies: (a) computer science pedagogy, (b) computer science goals, (c) approaches to computer science, and (d) interdisciplinary connections. I assigned each typology a color to quickly identify the differences between the collected data in the documents. I read the data to familiarize myself and reread the data. I coded the data based on the colors of the corresponding typologies. I created a matrix with the collected data and considered the appropriate typology for each entry. I repeated this step to confirm or modify the connections between each piece of data and typologies. I reviewed the matrix and identified patterns that existed between the codes. I identified the main ideas and searched for patterns that existed between the codes. The findings from the last step revealed emergent themes.

I identified three themes within the typologies: (a) analog computer science, (b) building an inclusive computing culture, and (c) focus on equity. I referred back to the collected documents and searched for evidence that supported the themes. I also reviewed the data to look for evidence that did not support the themes. After completing a comprehensive review of the collected data, I wrote one-sentence generalizations for each theme. Lastly, I identified powerful examples that accurately and clearly represented the findings (Hatch, 2002).

Summary of Findings

The findings from this study revealed that elementary teachers experience a combination of exterior and interior challenges to teaching computer science that are significantly different from their secondary counterparts. Some of the barriers can be overcome by participants, but others are systematic and teachers are not able to control those elements. The external barriers were elements that could impact teachers' success teaching computer science but were barriers they could not control (Rotter, 1975). External challenges consisted of barriers to teaching computer science that teachers had no control over such as instructional minutes and district priorities. Teachers also faced internal challenges that were in their locus of control because they felt they could still impact the outcomes (Maehr & Meyer, 1997). Internal barriers included elements teachers could influence such as modifying pedagogical approaches and building self-efficacy.

Participants discussed positive and negative teaching experiences specific to computer science education. Teachers faced challenges in learning the content but also understanding how to properly teach the concepts. Unique to elementary grades, the expectation to teach all subjects was a challenge teachers face in implementing computer science.

Participants found time to be a significant barrier to teaching and learning computer science. Limited minutes in each school day are impacted by new district priorities, site goals, and content areas such as computer science. Teachers described having to prioritize implementation for a new math adoption as a challenge impacting their effectiveness and ability to teach computer science. Time to prepare and plan lessons was reduced by the new math adoption as teachers found themselves learning how to teach and assess using an entirely new curriculum. Time spent to become comfortable with the new math adoption negatively impacted participants' opportunities to teach computer science.

Participants discussed not having enough prior knowledge in what teaching computer science can look like and a lack of their own foundational knowledge. Without prior experiences learning computer science, teachers were unfamiliar with the content and unsure how to accurately gauge their effectiveness. Teachers described a sense of uncertainty in their pedagogical approach to computer science and shifting to a model of allowing their students to take the lead. Teachers believed their students could learn the content faster than them and acknowledged needing to facilitate the lessons rather than providing direct instruction as they do in other content areas.

Despite facing a range of challenges, participants found value in teaching computer science. Teachers found opportunities for students to exhibit signs of feeling empowered while shifting to an active role in using technology in their classrooms. Participants noted an increase in students' problem-solving strategies and skills.

Teachers acknowledged being able to provide positive learning experiences for many of their students, particularly for students who were not as engaged with traditional subjects. Participants noted an increase in engagement and interest when teaching computer science. Participants also shared frustrations in teaching and learning computer science as they experienced multiple barriers to planning, teaching, and implementation models.

The documents revealed the teachers felt computer science learning activities that started with hands-on lessons were prioritized in their pedagogical approach. Teachers shared lesson plans that encouraged students to engage in learning computer science concepts and utilize computational thinking practices prior to demonstrating their learning on a device. Meeting notes revealed teachers most often planned to teach computer science as a part of their math time or as a specialty item during the week. Computer science was not included during the other instructional content areas that elementary teachers provide for their students. Documents related to computer science in the district indicated a clear focus on equity and access for all students, particularly underrepresented minorities and females.

Presentation of the Data and Results

Data from this study included two sets of interviews and collected documents. I used the inductive analysis approach for the primary and secondary interviews and typological analysis for the documents (Hatch, 2002). The data and analysis of the results are presented in this section.

Semistructured Interviews

Data collection and analysis revealed patterns within the frames of analysis. The patterns or subthemes identified during the process revealed the meaning of the data (Hatch, 2002). In total, 23 codes emerged from the collected data (see Table 2).

Code 1: Limited access to devices. Participants noted limited access to devices as a barrier to consistent implementation models. Although computing concepts can be taught with unplugged lessons and activities, computer science coursework eventually requires students to have a consistent level of access to apply their knowledge of computing systems with hardware. As teachers began to increase their levels of implementation, having limited access to devices whenever they were needed presented problems. Kimberly said, “Having more computers would just help you know because they [students] wouldn’t have to share all the time.” Keisha had similar sentiments as she indicated having to share computers with other staff members following a rotation model. She stated, “I wish we could keep the cart but it’s only available on rotation so after a month it got moved to another teacher.” Deborah developed a strategy to share access to devices with her peers,

> It would be really helpful if we could have one device for every student but I get why that might not be possible. One way I kind of worked around it was by making friends with other teachers and borrowing computers from their class.

Teachers who previously taught at other schools with differing levels of access to technology clearly noted the impact of reduced accessibility. Jefferson explained the difference between schools as he transitioned at his new site:

> You know when you're at some schools, you're fortunate since you have the one-to-one devices. So no that isn't an issue for those schools. Here at Lincoln, you know we get the Chromebooks and we were one-to-one for the first few months because lower grades didn't need them. That helped a lot. Once the Chromebooks started to get passed down, that's the cart you know, two or three times a week, that limited the accessibility to it. Even though I have a few Chromebooks in the classroom now, I don't have daily access so that that would be a challenge.

Changes in site policies and procedures in the distribution of hardware was noticeable to teachers who previously had consistent levels of access to resources. Teachers accustomed to reliable access to devices noted the difficulty in new teaching enviornments where access was limited. Teachers felt frustrated even though they understood the reasons why they did not have the same levels of resources in their classrooms. Teachers noted having a shift in mindset as well as identifying new strategies in these situations so they could better support their students.

Table 2

Overview of Themes and Subthemes

Theme Number and Developed Themes	Subthemes
Theme 1 - Elementary teachers experience external barriers through organizational structures that impact their ability to teach computer science.	Limited access to devices Lack of Internet access Digital literacy skills Time during the instructional day Ongoing support District coach Administrator knowledge
Theme 2 - Elementary teachers experience internal barriers to teaching computer science such as their limited efficacy to teach new and unfamiliar content.	Foundational computing knowledge Struggling to collaborate Lack of prior knowledge Building interdisciplinary connections
Theme 3 - Elementary teachers rely on coping mechanisms for external barriers to reduce the impact of organizational structures.	The value of exploration Hands-on learning Conferences Needing time to learn and reflect Utilizing YouTube
Theme 4 - Elementary teachers rely on a shift in mindset to address internal barriers.	Harder to learn easier to teach Building professional learning networks Engaging for students Preparing for the future Shifting from instructor to facilitator Learning it together Students as experts

Code 2: Lack of Internet access. Participants found the lack of reliable and consistent Internet access at home to be a problem in their classrooms as students progressed at different rates. Students who had the resources of a computing device and Internet connectivity were able

to continue learning computer science at home. Many of these students accessed the resources online and were able to progress at a faster rate, getting ahead of some of their peers. Jefferson noted, “I was seeing kids saying ‘I want to do it more’ because they didn't have Wi-Fi at home or they didn't have a device, they were limited just doing it at school.” Ellison found herself having to differentiate her lessons as students with Internet access were able to work at home and get ahead, “For kids who didn’t have that access they fell behind some of their peers so I had to figure out how to keep the high kids challenged while supporting the rest of the class.” Claudette also shared, “Some of the kids were able to fly through because they have a computer at home but for some kids, they were limited to the time we had in class.”

Lisa argued that older students in elementary school should be provided with hotspots if they did not have reliable Internet access. She felt it would help them progress in computer science but also benefit them in transitioning to middle school where all students receive their own computers. Lisa said, “At our middle schools we give kids hotspots so they can get online. I think we should do the same for our younger students, at least fifth and sixth grade because it's preparing them for middle school.”

Code 3: Digital literacy skills. Participants shared how important it was for students to be digitally literate and understand how to navigate a world surrounded by technology. Pamela said, “I think that's kind of the direction that this I guess this group of kids or this era of kids are headed towards. They’re going to a more digital world. I guess they need to be digitally literate.” Deborah said, “we're surrounded by technology and probably every kid is going to need some of these skills.” Maria felt students were born in this new digitally connected reality, “We have these digital natives and have to make sure they’re prepared for the future.”

Teachers noted the importance of digital literacy skills for younger students as a part of computer science education. Students needed to understand how to use their devices to access content and curriculum. In early grades, students benefited from having touchscreen user interfaces as they did not have to learn keyboarding skills to be able to login and use their devices. Keisha said, “Once we got past the logging in and not having to use the keyboard for everything it was much easier to do in class. Having the touchscreen Chromebooks was huge.”

Teachers acknowledged that they initially had a misunderstanding of digital literacy and computer science. Maria said,

> If you asked me a couple years ago I would have said digital literacy and computer science were pretty much the same thing. As long as my students knew how to do things on their computers it felt like they were learning and developing skills.

Ellison said, “Learning how to use a Chromebook or iPad isn’t the same as understanding how those things work. It’s [computer science] learning how a car moves, not how to drive a car.”

Code 4: Time during the instructional day. Finding time for computer science education as part of the instructional day was difficult for teachers as they felt they had very tight schedules. Even after making a commitment to teaching the content, teachers found it difficult, “We committed to 20 minutes daily the first year so that was a challenge to kind of look where am I going trim in my schedule,” said Heather. Keisha noted, “The other challenge for me is finding time in our curriculum. Elementary school teachers have in our district so many things they have to do.”

Teachers felt they were balancing their instructional schedules and navigating difficult decision making of losing time for other priorities when they included computer science. Jefferson explained,

> The first thing was just trying to fit this in. Is it going to replace math time? Or science time? Where am I going to find the time to fit this in and what am I going to take out? You know that schedule was a part of it.

Rachel and Ellison discussed experiencing feelings of having too many expectations in their instructional day without having anything removed. Ellison said, “Time was the biggest challenge. It seems like nothing has ever been taken away from what we have to do it's only added on and so it was challenging to find the time.” Rachel also shared having that feeling of prioritizing priorities within her instructional minutes. She said,

> The challenge isn’t teaching, it’s all these new things we get asked to do….you need a lot of creativity because you only have a certain number of minutes for instruction. You know we just have so much content to cover in every subject area.

Lisa found it challenging to teach computer science in elementary school when she compared her experience to teaching at a middle school. She stated,

> It's so different at this school because I'm used to having a dedicated time period with a new group of kids. I guess that would be one of the benefits of being at a middle school because we had set periods.

The structure of a contained elementary classroom setting was in direct contrast to the flexible period structure of middle schools.

Code 5: Ongoing support. Teachers said that they needed follow up trainings and support opportunities with ongoing support during the school year. They felt continued professional development and resources would help them deepen their understanding and develop their skills in teaching computer science. Rachel and Deborah said that follow up trainings in computer science would help teachers build up their confidence. Rachel stated, "You don't have the same support you had in the training. Just having a refresher training would really help." Deborah said,

> A lot of times we go to a training and we're excited to try it back in the classroom but we get back and it kind of falls on the back burner. Having some opportunities to get a training to get teachers back on track or remind them of the tools and resources would I think help more teachers feel confident about what they're doing.

Heather indicated a desire for shorter follow up trainings for teachers. She said, "I think A refresher course. Like a 45-minute refresher course each year would be really good." Claudette also felt multiple opportunities for ongoing support was essential in helping elementary teachers. She stated, "We need more opportunities to develop our own abilities. If we had more trainings that built on the previous ones, we could learn new skills that we could take back to our classrooms."

Code 6: District coach. Participants felt that a district coach would be an asset in improving their experiences teaching computer science. Teachers felt a supportive district coach would be able to provide support in lesson planning and delivery. Teachers did not feel that they needed a coach who would pull out students to teach the lessons. They noted having a coach who could observe and offer suggestions during their own lessons would be beneficial. Jefferson said that having someone for the first few lessons would help teachers build up the confidence to teach the lessons on their own. He said, “Just having that extra semi expert or expert that can say, ‘You know what, I'll walk you through it and you know we can make this happen.’ I think the first couple times that would be huge.” Rachel and Ellison felt teachers would benefit from the same type of support. Rachel shared, “Maybe someone from the district that can come in and model a lesson. Somebody to walk you through a lesson plan, help you teach the lesson, and go over assessments.” Ellison said,

> For elementary teachers one of the best supports that I can think of just in the spur of the moment is the possibility of providing them with someone to first teach them how to do it so they feel comfortable starting it but also maybe doing it with them.

Code 7: Administrator knowledge. Participants felt site administrators should build their knowledge of computer science to be able to better support staff. Rachel and Maria felt administrators would be able to support teachers teaching computer science by building up their knowledge to know what to look for when they visit classrooms. Rachel said, “They need to learn with us and go through it with us so they understand.” Maria added, “I don’t think they know how to support teachers because they don’t understand what it is and what it should look like.” Ellison stated, “Principals have to be able to walk into a classroom and understand why a lesson is being taught a certain way or what we’re trying to get the kids to understand.”

Teachers stated computer science could become more common in schools if administrators mandated that computer science be taught as part of the instructional day. They felt administrators would be able to prioritize computer science education and encourage elementary teachers to teach computer science if they participated in their own professional development and built up an expectation of teaching the content with their staff. Heather said,

> I honestly think that sometimes site administrators have a disconnect from implementing something and what it looks like in the classroom. They haven't experienced it and so I just think over time as you're out of the classroom you do forget you know all the stress and anxiety that comes with all these things. So I think it's important for site administrators to be like fully immersed.

Keisha said that administrators could make computer science education a mandatory part of students' learning experiences but would need to be prepared to lead their staff. She said,

> Principals should periodically check-in or you know talk to teachers about what they're experiencing with something that's new to everybody. I think that would help and most teachers would feel better. But again, that only works if they attended the training with us or went to one on their own. If they don't go to the training then they don't know what to ask when they check on us.

Code 8: Foundational computing knowledge. Teachers indicated a need to build foundational knowledge in computing to be able to better support their students. Rachel and Keisha expressed a need to understand computer science concepts and build their knowledge on what teaching computer science looks like. Pamela explained how she decided to build her own knowledge because she saw increased engagement with her students when she taught the lessons:

> I just I find myself like three hours on Google reading about things or watching videos and just seeing the impact in my classroom makes me passionate about learning more

> about it. When I know something is engaging to my kids I want to be able to understand how to teach it so I just kept looking up things on coding.

Rachel noted building foundational knowledge was important for teachers, “just knowing computer science ourselves to be competent to teach it.” Ellison obtained a supplementary credential to build her knowledge:

> I just wanted to learn more and understand it in depth. I received some training before and really enjoyed learning how to code and if I want to be able to teach it, I needed to learn it . . . and that’s probably why I went for that credential.

Rachel wanted to learn and go through the entire process of learning how to write code to develop her own computational artifact as a means to build her knowledge:

> I’d like to have the opportunity or the experience to actually build something meaningful to see what the product is. I guess to almost have to start from step one and then come out with like a final product. Something I made using my own code.

Teachers expressed a desire to learn and grow their own knowledge in computing but found it difficult to prioritize and find the time to learn. They wanted to gain an understanding beyond what was provided at conference or professional development.

Code 9: Struggling to collaborate. Elementary teachers struggle to confidently and consistently collaborate with peers on computer science education. Maria shared how important it was to collaborate with her peers but was unable to find opportunities to develop their own capacity. Their collaboration differed from other subjects as the team was able to share successful experiences but not pedagogical strategies, “We talked about if we saw a benefit about making it like a team decision to incorporate some coding activities or computer science or computational thinking activities." Claudette explained, "As a grade level we have not gone into code talk as far as planning and collaboration. I would touch base with them to make sure we're all in similar spots but not strategies."

Lack of consistent collaboration impacted consistency in instructional models as teachers taught the lessons in ways they felt was most appropriate. Pamela noted, "We have grade level collaboration and plan as a team but there isn't enough time to include planning for coding. We teach it but we do it our own way so it isn't consistent." Heather shared a similar experience recalling her teams struggle to collaborate, "It feels a little like it's the blind leading the blind. We look at each other and do our best but we don't know right away if we're planning correctly.

Deborah shared how she felt isolated without collaboration with her peers. She shared, "I mean I was doing it alone, I didn't have anyone to plan with or talk about how my lessons went. It just surprised me how little people around me know about it." She wanted to reflect on her teaching and classroom experiences with her peers but felt unable to share her knowledge without a team dedicated to collaborating on computer science education.

Code 10: Lack of prior knowledge. Participants noted they did not have prior knowledge in computer science as part of their own education, credentialing programs, or college experiences. The lack of prior knowledge impacted teachers' ability to lesson plan and they found themselves learning what teaching computer science could look like as they continued to deliver lessons. Jefferson said, "So there's that initial unknown. I guess the unknown was a big challenge." Claudette expressed feeling lost as she did not have prior knowledge, "So I'm not as lost. . . . That first year was really a struggle."

Teachers expressed having feelings of uncertainty teaching computer science as the concepts were not included in their preservice requirements. Rachel said,

> I didn't learn about this in school when I was a student and I didn't get to do it in college. Even when I got my credential, I don't remember having anything like coding in my classes or student teaching.

Heather said,

> I use technology and know how to use a computer but it wasn't like I knew how to make an app . . . I don't remember learning any of this [computer science] in school. During my credential program we didn't get exposure to this [computer science].

The lack of prior knowledge impacted Maria's' perceptions on who could learn computer science. Maria explained, "I never knew that you could do block coding. I didn't know you could scale it down . . . that was like something that I could be successful at or that I could learn alongside my students."

Code 11: Building interdisciplinary connections. Teachers said that finding interdisciplinary connections was a prominent strategy in successful implementation models. Ellison and Pamela found opportunities to connect computer science with their mathematics instruction. Pamela said, "I did it during math time because I thought well, it teaches them problem-solving and perseverance and builds their stamina. . . . It's woven into everything." Ellison shared how she combined math and computer science as part of small group instruction and a rotation model with math,

> Well I would say I integrated it into math the best I could and I also used our Wednesdays which are our collaboration days. The kids leave early on Wednesdays and

> at the end of the day I would do unplugged lessons from Code.org on those days and then I would do the plugged lessons in rotations for math when I did small group instruction and math centers.

Deborah shared how finding connections started to come naturally to her. She said,

> I basically found ways to integrate it through everything that I did even if I wasn't actually teaching a full computer science class. I think it was because I wasn't afraid of it and I knew what it was capable of, it was always part of what I did.

Other teachers who did not have the capacity to find interdisciplinary connections noted they would benefit from having curriculum in other subjects aligned with computer science. Maria said teachers would be more likely to teach computer science if connections were explicitly established and shared with teachers:

> I really wish that this was embedded into our math framework somehow and maybe with the new standards it will be. If it isn't integrated you still have those teachers who aren't going to do it because it's not an adopted standard or it's not in the curriculum, it's something going above and beyond.

Keisha shared that connections can go beyond mathematics:

> It's really hard at elementary because we have so much to do, if there's a way that we could integrate it with core curriculum and science and social studies that would be a way I think you can get more people to do it . . . I think that's why if we had programs that had components of computer science it would make everyone's job easier. We should look at adoptions that integrate other subjects you know so you're teaching language arts but maybe you're also teaching science or history. If we adopt a new science or math

> curriculum in the future whoever makes that decision needs to make sure it has computer components.

Code 12: The value of exploration. Computer science as a subject in elementary schools is new to teachers and students. Professional development for computer science was significantly less than other content areas and the trainings were limited to single day sessions. Participants found that exploration was important for teachers and their students because they lacked prior experiences and sufficient professional development opportunities in computer science. Teachers found they were able learn and build their understanding of computer science by accessing the curriculum and resources and exploring on their own. Jefferson said, “It was a lot of exploration on my own and maybe even some colleagues. . . . Once I started looking into it, it just seemed like something new and different that I wanted to try in my class.” Rachel discussed her surprise at the accessibility and being able to scaffold the content for younger students, “I didn’t know we could do it with blocks because I thought you had to learn really complex stuff that only technical people know how to do.” Heather shared a similar experience with Rachel as she felt more confident to be able to teach computer science once she explored the content. Heather said, “When I heard computer science, I thought it would be difficult and almost intangible for the kids. It seemed very abstract until I actually tried it. Then I did see, it was extremely the opposite.”

Exploration proved to be valuable for students as well as teachers. Students who are not successful in other content areas were able to demonstrate their abilities in computer science and surprised their peers and teachers. By allowing students to explore and uncover new learning, Maria and Claudette found students with special needs and previous behavior issues becoming successful. Maria shared,

> I have some students with special needs and some of my English learners who are my top students in computer science. It gave them something to be proud of and share with the rest of the class. I think teachers need to just give it a try and see how their students react to it.

Claudette stated, “I will say that all of my students found some success in it. Kids that might be considered behavior problems particularly connected with the lessons.” Heather discussed how students were able to find alternative solutions through exploring the content and collaborating with peers, “I loved to see the creativity that the kids brought in. They found ways to solve the puzzles in ways I didn’t even see and the collaboration between them was incredible.

Code 13: Hands-on learning. The participants said that teaching computer science with unplugged, hands-on activities provided pathways to connect abstract concepts. Pamela said,

> I like to learn by doing, that’s probably my most favorite way because then I can kind of go off and explore so I took the Ozobots and tried doing it with the markers and a light just went off.

Maria discussed how some of the resources were able to build learning connections. She said,

> So the district has a bunch of kits that they have purchased and it blends the physical aspect of coding with the digital aspect. So it’s more hands-on equipment that the students can use to kind of bridge that gap between the abstract idea of coding.

Lisa shared how she found students were better able to build connections and scaffold their understanding in programming by writing down their code before trying to input their programs on the computer. She said,

> We had the drivers and the navigators and so really using that to help them go back and forth coding first on paper and having them do all that work ahead of time. All that thinking and processing ahead of time really helped them understand what we were doing and why.

Code 14: Conferences. Participants said that conferences were a significant gateway into their introduction to learning more about computer science and deciding to teach it in their classrooms. Prior to pursuing professional development or deciding to explore on their own, teachers became interested in learning about computer science during conference sessions. For many of the participants, district provided professional development opportunities in computer science were not available when they first became interested in teaching the content. Jefferson and Rachel said that conferences were their first introduction to computer science. Rachel spoke about her initial exposure to computer science education, "It was the annual convention for computer science I got to attend with the Department of Innovation. . . . I didn't learn everything and wouldn't say I'm a computer genius but I knew it was important." Jefferson said, "I probably joined in at least a handful of short sessions at different things. . . . So I just looked at it and it just made sense. And it also looked very engaging too." Heather also shared how shorter sessions about computer science piqued her interest in the content, "I also went to some mini sessions in the different professional development conferences the district provides like August PD. . . . It seemed engaging for the students and I will do anything to engage my students in a different way."

Ellison decided to obtain a supplementary authorization to teach computer science based on her experiences at conferences. She shared how she was able to build her capacity and knowledge in computer science by attending the same conference in different years. She said,

> I've been to two different Code.org facilitator summits which were interesting experiences. The first summit I had no idea what I was doing and what I was learning but I just knew it was important to be able to know how to teach this to my students. The second summit was last year and at that one I knew so much more and was able to jump in right away.

Code 15: Needing time to learn and reflect. Five teachers said that they needed time to learn to become better computer science teachers. Teachers felt they struggled or had difficulty learning computer science and would benefit from additional time dedicated to their own learning. Keisha felt a desire to learn but did not have the time to dedicate to building her own knowledge, "I need to learn more for myself but I don't have the time." Teachers felt the district should provide resources for teachers to take the time to learn and reflect teaching computer science. Maria said, "We need a day where we get subs and just get some time to learn. We would benefit so much from having uninterrupted time where we don't have other responsibilities outside of getting to learn." Claudette stated she would also benefit from having additional time to learn as she felt she was having trouble learning the content, "I'm struggling as a learner on it so I realize that I'm not the most stellar code person right now, but again I felt it was really important to provide it for our kids."

Participants also said that they needed time to reflect on what they learned as well was their own teaching practices. Lisa said, "When I teach a lesson for the first time I need time to think about what went right and what didn't go right." Heather felt she also lacked time to reflect after teaching computer science lessons. She said, "We are backed up to the minute. When we finish a lesson, we move right along to the next activity. I don't have the luxury to think about happened until the end of the day."

Code 16: Utilizing YouTube. Participants said that YouTube was a valuable resource in developing their knowledge of computer science and helping them learn on their own. The online resource provided teachers an avenue to find on demand content for subject specific questions. Deborah, Heather, and Ellison mentioned how they were able to find answers to some of their questions through YouTube. Deborah said, "I've used Twitter and YouTube to look up some things quickly and get responses from other teachers." Heather shared, "I also found resources online on YouTube and Twitter that other teachers share." "I watched online videos about the Lego Mindstorms and I use social media if I have questions," said Ellison.

Videos that were created for students were also helpful for teachers. Rachel said,

> I actually really liked the videos we show the students because the language was easy to understand and they were made so kids could learn so they were really helpful. I learned from those videos and it made a lot more sense to me.

Teachers were able to utilize YouTube as a learning resource and found answers to their questions or watch videos about various computer science resources. Although the videos are not direct substitutes for professional development, they were an additional tool that scaffolded teachers' knowledge of computer science. Participants were able to use YouTube as a source of instructional computer science content to gain a depth of knowledge specific to their needs.

Code 17: Harder to learn and easier to teach. Participants found teaching computer science to be easier than they initially expected and felt empowered to continue teaching the content. Although they experienced challenges and felt they wanted to develop and grow their content knowledge, by teaching computer science lessons and seeing their students succeed, teachers felt confident and empowered. Jefferson said, "It seemed easy to do if I followed the lesson but it was still all new to me. Once I got into it [teaching] it was much better and easier than I thought that it was." Lisa said,

> The more I taught it, I could see what it was doing for my students and I felt like I was really a computer science teacher. Once you see that type of feedback you just know you can't take this away from them.

Heather noted, "When I heard computer science, I thought it would be difficult and almost intangible for the kids. It seemed very abstract until I actually tried it. Then I did see, it was extremely the opposite." Claudette said, "After the training when I planned my first lesson I was scared that I would get it all wrong. Once I started teaching it wasn't as scary as I thought."

Ellison said that learning computer science was challenging but teaching it did not require the same technical knowledge. She shared,

> I struggled in my classes for my authorization and it was really hard to learn. Teaching my students didn't require the same depth of knowledge so it was easier teaching an algorithm or debugging with my kids than it was for me to apply that in the coursework.

Code 18: Building professional learning networks. Participants noted that building professional learning networks helped them connect with other educators and find answers to questions they had about teaching computer science. Teachers found Twitter to be a valuable resource in connecting with other elementary teachers teaching computer science. Ellison said, "I can put out a question asking hey what do you think about this, or how would I teach this and people on Twitter will respond." Claudette referenced using her professional learning network on multiple platforms to ask questions but also find resources:

> Social media has been helpful because I use Twitter and YouTube to find resources and things that other teachers are putting out. There's a good group of teachers out there that are sharing things that are working so building a network outside of your immediate circle is pretty helpful.

Pamela said that she started using Twitter specifically to build up her knowledge and library of computer science resources, "There's a network of teachers all over the country that share things so it's much easier to find answers now than it was when I first started teaching."

Participants stated they were able to quickly get responses to their questions from multiple respondents by leveraging social media and building up their professional learning networks.

Code 19: Engaging for students. Participants said that they taught computer science because they saw how engaging it was for their students. Teachers shared stories about their students and classroom successes teaching computer science. Jefferson and Pamela indicated some of their struggling students were able to find success through learning opportunities in computer science. Jefferson said,

> Some of the kids who really struggle with the general academics completely

> latch onto it. They just take off with it and are amazed by it. And we’re talking all levels, like I have I’ve had students who are behavior issues but for some reason they really like coding and are willing to like actually sit down and try it and do their best on it.

Ellison said, “I had a parent call me and say it was the first time his son asked if he could go to school. This student was completely disconnected from school until we started coding.” Deborah noted how she saw her students begin to transfer their learning in computer science, “That purpose of teaching something that really is going to have a long-term positive effect on students was one of the things that I noticed and it helped me to see how amazing this could be because it impacted other things in class.”

Claudette, Pamela, and Jefferson felt some students found computer science to be Engaging because it was an alternative way for them to showcase their learning and find success.

Claudette said that students were engaged because the content was something new and different for them. She said, “I think the power of doing something different in this day that gets kids that are sometimes hard to reach is important. Having code as one of those options was amazing and wonderful.”

Pamela said,

> Maybe that confidence that it instills in them will inspire them to do better in the other subject areas and ultimately impact their life forever because they achieved what they needed to but in a different path. So because I think all kids want to learn, I just think for some kids their learning needs to be different and this really is something different.

Jefferson found computer science to support his students with special needs. He noted,

> They just take off with it and are amazed by it. And we’re talking all levels, like I’ve had students who are behavior issues but for some reason they really like coding and are willing to like actually sit down and try it and do their best on it. It’s a huge motivator. And then you have kids who are even special-needs kids. I had an RSP kid last year who really struggled academically. He could not really meet the grade level standards. We got him on coding and he was willing to try it and wrote tons of lines of codes and just was willing to like make it happen.

Code 20: Preparing for the future. Teachers said that teaching computer science was important because they felt they were preparing students for the future. Despite not feeling completely confident to teach computer science, preparing students for the future motivated teachers to provide computer science learning opportunities. Jefferson said,

> I heard this statistic that some high percentage of amount of jobs that are going to be out there don’t even exist yet. Our kids are going to be doing them, and they need to have some computer science to get those jobs and to be prepared.

Pamela had similar sentiments about the unknown future of jobs that students will need to be able to compete for by learning computer science. She stated,

> I heard about the research and how by 2020 most of our jobs aren’t going to be fulfilled because there aren't enough kids in school to fulfill those jobs and they won’t have the necessary skills to compete. I think if they’re exposed at a really young age they have a greater chance of developing the skills and can even build the interest in doing it.

Keisha felt students come to school with so much exposure to technology that teachers had a duty to teach and prepare their students. She felt teachers could no longer plan to learn how to integrate technology but should be required to include it as part of their instructional responsibilities. She stated,

> I just feel you can't teach like that in the 21st century because you're doing such a disservice to kids if you're not preparing them for their future. I'm teaching kindergarteners so I can't even imagine what they might have by the time they to high school you know, and everything is moving so fast we have to start as soon as they get in school.

Maria also noted her students were coming to school differently than students in the past. She said,

> I think students today are born using technology in ways teachers are not used to and we need to help them to problem solve and be creative with computers and technology. We have these digital natives and have to make sure they're prepared for the future.

Code 21: Shifting from instructor to facilitator. Teachers acknowledged not having a full understanding of computing concepts and shifting from a model of providing direct instruction to facilitating. Teachers felt they were able to play a critical role in supporting student learning even without a full understanding of concepts. They became classroom learning leaders seeking knowledge rather than being the primary source of information. Teachers developed the strategy of shifting from a model of being the primary source of knowledge as they discovered their students' abilities to learn computer science. Maria talked about not always having the answers but leading students to their own findings by facilitating their learning. She shared, "I ask students questions to lead them to the answers. I found myself asking them to think about their own reasoning as they worked through this [computer science].” Claudette said, “I found I had more challenges than the kids did and I got way more flustered than they did but we kind of learned together and I could rely on them to problem solve together.”

Participants shared that they experienced feelings of limiting their instructional authority in determining how and when students might learn computer science. Kim stated, “I had a feeling of letting go. I had to learn how to guide them because sometimes I didn’t know what the answer was or exactly how to get there.” Ellison described the shift as a form of gatekeeping. She said, “You’re used to giving the kids bit by bit. So if a student fails a math test you don’t give him the next lesson but I couldn’t really do that with this [computer science].” Deborah said,

> Some of the kids absolutely take off and get to a point where you have to let them keep going. You can’t hold them back just because you’re not ready so I had to get comfortable letting them continue. At that point you’re not teaching so much as directing and guiding.

Code 22: Learning it together. Participants found themselves making an instructional shift in learning content with their students. Teachers said they had little or no background in computer science and they developed a strategy in learning computer science alongside their students. This allowed them to be transparent about their lack of content knowledge with their students. Maria said she was surprised at the complexity of what she was able to learn with her students. She noted,

> I never knew that you could do block coding. I didn't know you could scale it down between like what a computer scientist does every day with like Java or the different computer languages and I didn't know there was anything below that that was like something that I could be successful at or that I could learn alongside my students.

Claudette said, "I know they need these skills and I don't know it, but I can help them learn it and go along the ride with them." Heather said,

> I was very upfront and let the students know that we were going to learn this at the same time. I let them know that even though I was a younger teacher, I didn't have a chance to learn any of this when I was in school. I think they really liked hearing that.

Jefferson shared experiences where individual students were not able to find solutions, and the class worked together to problem solve:

> And then some kids would get stuck on certain elements of it and I couldn't necessarily answer it right away. It would be, okay, let's think about this, and walk through together. So it took patience on the part of myself as well as the students to kind of troubleshoot and figure what the next step was.

Code 23: Students as experts. Participants said that students became experts in the classroom. Rachel said,

> They did a better job teaching me than I did teaching them. The kids really ended up being the experts and we kind of just did all the work together. If we got stuck we just skipped the puzzle and some kids would end up figuring out the answer.

Ellison shared how a student expert became empowered in class, developed his confidence, and was able to support his peers. She said,

> One student, and I will never forget this, his parent called and said. “I don’t know what you’re doing but this is the first time that Sofia has ever asked to go to school. She hates going to school and until this year it was a fight every morning. Please keep doing what you’re doing.” Sofia was super shy and just one of those kids that was struggling in a lot of areas and when we started coding she lit up. She became the class expert and would teach the other kids how to do things in code. It was one of those teaching moments you never forget.

Pamela had a similar experience with an English Language Learner. She noted,

> And this year I have an Asian kid who isn’t fluent and he’s not very vocal even in ELD. I teach ELD and it’s really hard to get him to say anything. I think it’s because he doesn’t feel confident but in code he was amazing, it’s all he wants to do. So some of my other kids were struggling in some of the puzzles and I said, “Brian do you think you can go over there and help?” I couldn’t believe it, he was speaking more than I’ve been able to get him to do all year. I've tried to draw it out of him and haven’t been able to but as soon as I asked him to do something with code he was over there, confident and excited and then he was just a ping pong around the room.

Heather said, “They found ways to solve the puzzles in ways I didn’t even see and the collaboration between them was incredible.” Claudette said,

> I rely on the other students so I’ll say, “Hey if you’re stuck on lesson 17, I notice that these two kids are at 19 maybe you could go over there and have them give you a hint.” So they’re learning how to rely on their friends to kind of give them guidance so you know we learn best with our peers.

Documents

I utilized the typological analysis model to identify emergent themes within the collected documents (Hatch, 2002). Documents included lesson plans, meeting notes, and publicly available documents related to computer science in the district. I printed each collected piece of data and arranged them by the type of document category. I listed a category number and page number for each document followed by a decimal point to indicate the number of the paragraph or section within the collected data; for instance, 1.3.5 would refer to the fifth paragraph on the third page of lesson plans. I used predetermined typologies linked to the research objective and guiding questions (Hatch, 2002). I created the typological labels while I was gathering research and writing the literature review. I set four typologies: (a) computer science pedagogy, (b) computer science goals, (c) approaches to computer science, and (d) interdisciplinary connections. The typologies directly related to the guiding research questions and objective of this study.

I read through the data multiple times and identified the typologies. I confirmed the typologies and assigned a color to quickly identify the differences in the collected data. I read and analyzed the documents, searching for information that related to the assigned typology (Hatch, 2002). I created a matrix with the collected data and identified patterns that existed between the codes. I identified three themes and referred back to the data to find evidence that supported the themes. I completed a comprehensive review of the collected data and identified powerful examples that represented the findings. Table 3 provides an overview of the typologies, the emergent themes, and excerpts from the collected documents that support the findings.

Table 3

A Typology of Collected Documents

Emergent Themes	Excerpts from Collected Documents relating to Computer Science
Theme A – Analog computer science	• Students will practice writing precise instructions as they work to translate instructions into the symbols provided. • Students will relate the concept of algorithms back to real-life activities by playing the Dice Race game. • By "programming" one another to draw pictures, students will begin to understand what coding is really about.
Theme B – Building an inclusive computing culture	• Students should be split into groups where they will have to create directions for other students to draw a specific monster. • Students learn the simplicity and utility of loops by “programming” their friends using the language. • Experiment with new ideas and consider multiple possible approaches. • Work with others to develop solutions that incorporate all contributors.
Theme C- Focus on equity	• Emphasize technology and coding as an attainable career path for all students and encourage diversity in these fields. • Dedicated to expanding computer science access and participation, particularly for women and underrepresented students of color. • Every K–12 student exposed to computer science so they can decide whether they want to pursue that field.

Note. Four typologies: (a) computer science pedagogy, (b) computer science goals, (c) approaches to computer science, (d) interdisciplinary connections

Theme A: Analog computer science. The first theme that emerged from the documents was a focus on teaching computer science lessons without having students access a device. The lessons were designed to provide students kinesthetic learning opportunities designed to help them digest complicated and abstract concepts in ways that relate to their own lives. Students did not have to learn specific programming languages or develop coding skills to learn computing concepts. The lessons conveyed fundamental computer science ideas without requiring specific access or knowledge of hardware and software.

Theme B: Building an inclusive computing culture. The documents revealed that teachers placed an emphasis on building an inclusive computing culture. There was an emphasis on building inclusive opportunities by developing students' collaboration skills and encouraging creativity as a part of computer science lessons. The lesson plans called out opportunities for students to work in groups in collaborative environments where communication was key to building successful outcomes. Learning opportunities were designed to encourage students to express their learning with peers through creative outputs such as songs and dance.

Theme C: Focus on equity. There was a very explicit focus on equitable computer science learning opportunities for all students. The district and school vision for computer science included initiatives to drive engagement and enrollment among underrepresented students of color and female students. Increasing diversity and participation among female students was highlighted in multiple documents.

Summary

Findings from the data analysis revealed that elementary teachers experience a wide range of barriers in teaching computer science. The types of barriers were both external factors that teachers could not control and internal factors. Elementary teachers were able to build and develop strategies in limiting the barriers so that they were able to support their students. Participants also identified multiple support needs that would help them improve their experiences teaching computer science. Ultimately, a shift in mindset helped participants approach teaching computer science in new ways that did not follow traditional instructional models used for other content areas.

Chapter 5: Discussion and Conclusion

Computer science significantly impacts the future of all students. The increased usage of software and hardware for all facets of daily life requires students to have an understanding of foundational concepts in computing. Students need the knowledge and abilities to compete and succeed in an increasingly competitive, digital, and globally connected economy. Furthermore, as part of a broad and comprehensive education, computer science allows students to learn how to solve complex problems through creativity thinking and collaboration. To provide students with this knowledge, early exposure in elementary school is key to building students confidence and self-perception of who can achieve in computer science.

Elementary teachers are critical gatekeepers to access and have the opportunity to foster students' abilities in computer science. The participants in this study described their experiences teaching computer science and provided insight into the necessary supports for elementary teachers. The documents collected revealed additional insight into their experiences teaching computer science.

This chapter presents a summary and discussion of the results and how they relate to the literature and conceptual framework. I present implications of the results for practice, policy, and theory. I conclude with recommendations for further research and a closing summary.

Summary of the Results

I utilized semistructured interviews to gather data from the participants. The data revealed that elementary teachers teaching computer science a diverse range of challenges. Teachers in the study experienced external and internal barriers to teaching computer science. External barriers presented challenges to teachers in being able to teach computer science as a core subject that was part of their instructional day. Internal barriers required teachers to reflect and reconsider their ideas of teaching and learning. Participants applied strategies teaching computer science that required a shift in their instructional approaches and mindset. Teachers indicated needing additional and continuing supports to feel that they were effectively teaching computer science while developing their own capacity to teach the content. All participants wanted to learn more about computer science from a knowledge and pedagogical perspective.

The documents in the study revealed elementary teachers are able to integrate computer science as part of their instructional day if they understand the importance of their students learning the concepts. Teachers committed to teaching computer science utilized part of their instructional minutes in mathematics to present learning opportunities for their students. The documents suggest an emphasis on equity and access for all students. Despite feeling overwhelmed with competing priorities, participants consistently planned time to teach computer science and acknowledged the valuable learning experiences it provided in their classrooms.

Discussion of the Results

Results: Research Question 1

The first research question identified the barriers teachers experienced teaching computer science. The participants experienced a wide range of barriers to teaching computer science but remained committed to teaching the content to their students. Teachers experienced external challenges beyond their control which included the schedule of instructional minutes, district priorities, lack of access to devices, and competing initiatives. The district recommended schedule of instructional minutes did not provide teachers an opportunity to insert time for computer science. Participants planned to teach computer science during their math time with three participants utilizing science or minimum day scheduling as alternative opportunities. Participants shared frustration with the new math curriculum that they were teaching during the time of this study. The demand on their time for planning and structuring lessons with the new curriculum proved challenging and at times limited their opportunities to teach computer science even when it was planned in their schedules. The new math adoption required teachers to teach mathematics in ways they were not familiar with led to increased times for some lessons. Teachers also expressed frustration with a lack of consistent access to enough devices for their students. Their school did not have enough computers for all students which required teachers and students to share the available resources through rotation and sharing models. Teachers felt they could increase their consistency in teaching computer science if they had readily available devices for their students. All of the stated factors were external barriers teachers experienced teaching computer science as they did not have control over those factors.

Teachers also experienced multiple internal barriers teaching computer science. Internal barriers were challenges teachers experienced that were in the scope of teachers' control. Teachers felt they needed to build their own knowledge of computing to better service their students. Typically, elementary teachers are required to be content experts in multiple areas as they are not limited to single subjects like their secondary peers. Elementary teachers lack of prior experiences and knowledge of computer science presented an internal challenge of teaching content in which they were not familiar or confident. Teachers noted experiencing difficult situations specific to computer science as they did not always know how and what to teach.

Despite experiencing internal barriers and feeling uncertain about their pedagogical strategies, teachers remained dedicated to providing the learning opportunities for their students.

Structurally and organizationally, elementary teachers are required to be content experts in multiple areas as they are not limited to single subjects like their secondary peers. Participants referenced having previous knowledge of how they learned other content areas as students. Teachers understood what learning and teaching core subjects such as mathematics and language arts because of their own experiences as students and educators. They referenced and reflected upon their experiences teaching other content areas which provided them context in how they could pivot their pedagogical strategies and approaches. Teachers indicated they did not have a similar history with computer science and thus were unable to reference prior experiences to guide their teaching practice. Teachers felt they had to problem solve and identify strategies when they faced pedagogical challenges teaching computer science.

Results: Research Question 2

The second research question identified the strategies teachers developed in their experiences teaching computer science. The semistructured interviews revealed that the participants understood the importance and value of computer science education for their students and developed a new mindset in their approaches to teaching and learning.. Teachers felt that digital technologies and computing knowledge were essential components that all students should learn. Despite feeling they had an incomplete knowledge of computer science, teachers felt they had a purpose and drive to prepare students for the future. They felt the uncertainty of work and a digital economy highlighted the significance of computer science education for their students. Teachers indicated an awareness that computer science was widespread and broad enough that all students should acquire a foundational knowledge of computing principles.

The semistructured interviews also revealed that elementary teachers need multiple options and consistent levels of recurring supports. The participants felt that a district coach would be beneficial in helping them plan, deliver, and assess lessons. They felt feedback on their classroom practices in teaching computer science would help them refine and improve their pedagogical approaches while building up their confidence. Lesson planning for computer science integration with other content areas was also an area of support teachers felt they needed in their professional practice.

The participants thought that site administrators should develop their awareness and understanding of computer science to be able to support their staff and prioritize computer science education in schools. Teachers felt that site administrators who did not have some knowledge of computer science would not be able to provide them with adequate levels of feedback and support. The participants discussed how requiring site administrators to receive professional development in computer science could help elementary teachers in making them feel that they had top level support at their school sites. Administrator knowledge would also help teachers be able to communicate and inform parents about the importance of computer science education in elementary schools.

Collected documents revealed teachers emphasized teaching computer science lessons in nondigital or online methods. The lessons emphasized learning computer science concepts rather than specific programming languages. Teachers felt more comfortable teaching computer science through these types of lessons because they did not require a concentrated depth of knowledge in technical skills. The lessons also required students to learn in collaborative environments through active group learning.

The documents indicated an organizational focus on equity, diversity, and inclusion. The district and school vision for computer science emphasized providing computer science learning opportunities for all students. There was a clear understanding of historical inequities that the district and school tried to counter with a focus on supporting underrepresented students of color and female students.

Discussion of the Results in Relation to the Literature

The available literature on elementary teachers' experience is limited as most of the research is based on secondary teachers' experiences. Computer science education in elementary schools is an emerging instructional focus with previous lessons often being relegated to digital literacy lessons (Krauss & Prottsman, 2017; Montoya, 2017; Sentance & Csizmadia, 2017). Participants in this study provided insight into the limited knowledge base of elementary teachers experiences teaching computer science. Their experiences will add to the available literature and will provide insight for future researchers.

Elementary schools currently have a shortage of teachers with computing knowledge who are able to teach computer science (Ozturk et al., 2018). Schools and districts face a challenge to increase the number of elementary teachers who are able to teach computer science (Krauss & Prottsman, 2017). The availability of elementary teachers who are able to teach computer science is impacted by competing initiatives among organizations. Several participants indicated a focus and priority on other content areas, particularly in mathematics that limited teachers' abilities to learn new content. Teachers felt a sense of having to prioritize the new math adoption over other content areas due to messaging from the district office and their principal. Based on the nature of teachers' perceptions on their work priorities through organizational communication channels, teachers felt a stronger emphasis from the district office to prioritize computer science was necessary to increase participation. Furthermore, participants felt that a district or school site mandate to teach computer science should include corresponding layers of support and professional development to increase the pool of elementary computer science teachers.

Participants felt that elementary teachers needed a variety of supports to feel prepared to confidently teach computer science. The participants indicated receiving multiple layers of support would help them feel better prepared to teach and plan computer science lessons. Many participants began teaching computer science after receiving professional development through conferences or district provided opportunities (Ryoo et al., 2015). Teachers were eager to implement but soon found themselves needing additional opportunities to grow and develop their own abilities. Participants stated that follow up training opportunities would help build their knowledge in computer science.

The literature indicated teachers felt teaching and learning computer was extremely difficult (Yadav et al., 2017). Once teachers began teaching computer science lessons with their students, they expressed feeling a sense of surprise in their abilities to teach the content without necessarily having strong technical knowledge in computing (Margolis et al., 2012). Consequently, participants felt capable in being able to learn and teach computer science and felt they could improve their abilities if they were provided with appropriate supports and resources. Teachers stated that having a district coach facilitate their learning and professional development would increase their capacity to plan, teach lessons, and assess student learning in computer science. This finding confirmed the available research on teachers' perceptions on perceived difficulty in teaching computer science. Significant insight from the findings revealed that teachers' perceived efficacy levels increased once they began teaching the content in their classrooms. They were surprised at their own abilities to teach computer science without having what they felt was a strong background in computing knowledge.

Many of the participants reported utilizing Twitter and YouTube as an option for self-directed professional development tool. Literature on the use of social networks to develop professional practice is limited as professional learning networks are unconventional avenues for capacity building opportunities (Visser, Calvert Evering, & Barrett, 2014). Participants confirmed that professional learning networks provided opportunities for meaningful communication, collaboration, and sharing with other educators facing similar challenges. Although teachers found Twitter and YouTube to be useful for personal use, they did not utilize the resources for interactions with students.

Teaching computer science allowed teachers to utilize the technology in their classrooms in different ways. Teachers modified the usage of their instructional technology resources to facilitate student creation and creativity. The available hardware was used to guide student learning (Carter, 2014; Peters & Araya, 2011). Teaching computer science, participants noted the change in student learning roles shifting from passive to active users of technology.

Elementary teachers experience a transition in mindset with computer science education. Their self-efficacy increases with changes in mindset as they create new epistemological beliefs (Goldsmith, Doerr, & Lewis 2014). Participants who felt they were able to strengthen their self-efficacy emphasized providing peer-collaborative, active learning environments (Michalsky, 2012). They applied pedagogical strategies that did not rely on a heavy focus on direct instruction and lectures. Incorporating new teaching strategies, teachers felt they were able to increase their content and pedagogical knowledge and abilities to teach computer science concepts (Latham & Carr, 2012). Elementary teachers reconsidered their beliefs about computer science content and their ability to teach which impacted their pedagogical approaches (Bender et al., 2016).

Limitations

Limitations are constraints beyond a researcher's scope of control that may affect the outcome of the study (Hatch, 2002). The study was limited to only the experiences and perceptions of a small sample of elementary teachers from one school in one school district. The participants reflected experiences across all grade levels. Limitations of this qualitative case study included the limited amount of time to conduct research, my focus on a single district, and the self-reporting by participants of their experiences and needs.

Sample

The study was limited to only the experiences and perceptions of a small sample of elementary teachers from one school in one school district. Out of the 12 potential participants, 10 decided to participate in the study as two of the teachers decided to decline joining their colleagues. Purposeful sampling allowed carefully selected participants who met the criteria needed for research to share their experiences in meaningful ways (Creswell, 2013). Purposeful sampling was a limitation in the study because the small sample size presents a challenge in generalizing the results of the study to a larger population (Hatch, 2002). The participants reflected the diverse experiences across all grade levels in an elementary setting. The participants shared their experiences and the collected data reveals findings from the sample and not of other elementary teachers teaching computer science.

Study Design

For this research, I utilized a qualitative case study with two sets of interviews and collected documents. Interviews with participants were limited by the interview questions I decided to use for this study. The information I collected and analyzed in the qualitative case study was limited by my experience as a novice researcher.

Research Method

A case study is used to address a complex question or phenomenon in real-world situations (Creswell, 2013). This study was designed to explore elementary teachers' experiences teaching computer science by understanding the barriers they faced and the strategies they developed to overcome those barriers. The experiences are unique to the group of elementary teachers who participated in this study. The findings from this study revealed multiple themes from elementary teachers teaching computer science but they are not generalizable to all elementary teachers teaching computer science.

Data Collection

The information collected in this study was limited as the data consisted of interviews and documents. Interviews were a large source of data collection. The data were also limited as the information came from one small sample of elementary teachers during an 8-week period. Another limitation was the level of information shared through the collected documents. Some documents revealed more information and were more complete compared to others. Some participants wrote large amounts of information in lesson plans and meeting notes while some used short bullet points or phrases. The collected documents indicated when and how teachers planned to teach computer science. The lesson plans revealed the structure of their instruction, practice, and assessments. Meeting agenda plans showed where teachers planned to teach computer science during the week. Teachers often placed the lessons as a part or extension of their instructional math time with some teachers using instructional minutes for science. My analysis of the documents could not extend past the provided artifacts as I was unable to observe the participants' classrooms. I was unable to confirm if all participants followed their lesson plans and weekly plans. The inclusion of observations were not possible because of the limited time for this study.

Implications of the Results for Practice, Policy, and Theory

The findings from this study can help guide practice, policy, and theory. This section presents the potential impact of this study in relation to practice and policy in connection to the current literature. The section concludes with a presentation of the results as it relates to the conceptual framework in this study.

Practice

Local educational agencies and schools could implement practice recommendations in this section to build a coherent computer science implementation plan. Implementing the practice recommendations could empower and support teachers, engage stakeholders, and impact student learning in computing. Focusing on individual practice recommendations may allow for an easier implementation roll out as organizational shifts would not involve too many changes occurring at once.

Role of the teacher. Teachers should become comfortable with new teaching and learning experiences where they act as a facilitator and lead learner in their classrooms. In this new teaching role, teachers are not required to fully understand and have a mastery of content before they are able to teach their students. Teacher roles transition from being the source of knowledge in the classroom to a leader working with their students to solve problems and seek knowledge in a collaborative environment. Although teachers may shift their roles in the classroom, they should still utilize their professional expertise and discretion to engage all students and ensure everyone in the classroom has an opportunity to learn computer science in a safe environment.

Integrated teaching approach. In elementary grades, computer science can be taught in an integrated delivery approach. Computer science does not have to be a standalone course because of the way elementary schools are designed. Elementary schools have more flexibility than secondary schools in integrating content in the instructional schedule as they are not bound by explicit period changes. Students can learn computer science in integrated, grade appropriate lessons in elementary school and enroll in standalone courses in middle and high school.

Computer science can be integrated into classroom instruction with an emphasis on interdisciplinary teaching and learning structures. Examples of integrated teaching models might include but are not limited to: students learning how to debug algorithms as a skill to proofread and revise writing samples in language arts, teaching students how to build computational artifacts to include graphing in mathematics, and creating a science fair presented in digital formats with hypertext markup language (HTML) and cascading style sheets (CSS). Introducing computer science in elementary grades will help prepare students to become computational thinkers who understand how computing technologies function and impact their lives. Students will also be able to build their foundational knowledge in computing and develop their self-confidence for future computer science coursework in middle and high school.

Utilize instructional units for computer science. Instructional units dedicated to computer science can provide an entry point for teachers to develop their capacity. The units can lower teachers' wariness of teaching computer science and increase their confidence to teach computing concepts. Unit design should also include active learning and nondigital lesson plans to demonstrate how computing concepts can be taught without the use of a computer or tablet.

Building classroom community. Teachers should teach computer science lessons requiring students to actively learn with others to build a strong classroom community. Teachers should encourage students to interact with their peers through multiple learning opportunities. Students should have many options to communicate, solve problems, and create computational artifacts with other learners through classroom discussions and extended group projects. Teachers should actively build inclusive classroom cultures where all students feel they are able to share ideas and listen to and respond to ideas others bring up with through their unique perspectives.

Consider flexible implementation models. Based on the needs and available resources in local educational agencies, schools and districts may need to consider multiple implementation models to provide computer science instruction for all students. Implementation models should be based on the needs of the organizational context, capacity, and availability of resources. Exposure for students may need to begin with basic, introductory entry points in elementary schools with teachers and administrators working to extend and broaden exposure with a detailed plan.

Multiple implementation models should be considered by the decision makers in schools. Elementary teachers might begin with specific instructional units that are integrated into the general education classroom. Teachers would be able to provide limited exposure with the curriculum scheduling designated time for computer science. If the school has an existing specialty classroom such as a computer lab or makerspace, students could learn computer science when they visit those classrooms as the content can be integrated into those specialty spaces. Schools could also utilize an instructional model similar to specialty classes such as music and art. In those classes, a specialist teacher might push-in to provide students with learning opportunities or pull-out groups of students for dedicated computer science time.

Establish clear communication. Understanding what computer science is and why it is important for students is important for all stakeholders. As educational leaders, school and district administrators should communicate a clear vision regarding the need for computer science education and its importance for all students in a digitally connected world. Consistent and cascading messaging with computer science education can help ensure coherence across the organization. The messaging with elementary teachers can help them understand the defined vision for computer science and understand their role in its implementation. For administrators to build their knowledge, they need to be provided professional learning opportunities specific to their roles while also being encouraged to join their teachers in professional development opportunities.

Build family and community awareness and support. Improving the dissemination of information on the importance and value of computer science could help increase access for all students. Engaging the larger community outside of teachers and administrators will require a concentrated effort in educating families and communities. Families and communities could help encourage students to explore computer science because "when educators, parents, other citizens, and organizations work together to help students succeed, they strengthen the sense of community beyond the school" (Epstein, p. 611).

Creating community partnerships can build support systems that further develop awareness and access to computer science education. Community partnerships are essential to generate resources for effective schooling and students' growth and wellbeing. The connections between schools and the community stakeholders directly and indirectly promote students' social, emotional, physical, and intellectual development (Sanders, 2006). Student-centered school-community collaborations emphasizing computer science learning could provide more opportunities to reach all students and increase student engagement with connections between school and the community.

Expanded learning opportunities. Opportunities to learn computer science can be improved through activities outside of students' classroom experiences. Local educational agencies should work to collaborate and build supplemental computer science learning experiences with community stakeholders and partners. Activities with partner support does not need to be limited to the instructional school day as events could be hosted after hours and on weekends. Improving access to computer science education through expanded learning opportunities can help students make connections to their daily lives. Some examples of expanded learning opportunities include but are not limited to community stakeholders mentoring students, local companies providing technology donations and training for schools, and institutions of higher education hosting weekend computer science camps.

Multiple options for professional development. Elementary teachers need multiple and sustained professional development opportunities to build computer science content knowledge and relevant pedagogical practices. Professional learning materials and evidence-based professional development should prepare teachers to prepare and encourage all students to learn foundational computer science concepts. Professional learning for teachers can also be designed in multiple ways to provide a wide range of accessible options for staff.

Professional development for teachers can look different from traditional models of learning. Teachers can use professional learning networks as a way to receive informal professional development to best meet their individual needs. The value of teachers' convenience and device accessibility are considerations for providing alternative channels of support. Districts and schools should consider traditional and alternative methods of training and collaborative learning through digital resources such as YouTube and Twitter. Teachers are able to build professional learning networks outside of their schools and districts through globally connected platforms.

Professional development can help prepare teachers to leverage resources to customize their learning experience. Trainings should include ample time for teachers to experience learning concepts in learner-centered environments. Teachers should have time to explore and reflect on their learning and collaborate with their peers to help plan how they can teach and assess computing concepts in their classrooms. Professional development materials should prepare teachers of best practices in how to engage and encourage all learners to build awareness of computer science opportunities.

To provide students with more opportunities to learn computer science in elementary settings, teachers should receive multiple supports for lesson planning, lesson delivery, and professional development. Teachers need a wide range of supports to effectively teach computer science as they must balance learning new content knowledge and pedagogical knowledge.

Build technology infrastructure. Computer science concepts can be taught without access to hardware and software but students and teachers eventually need to transfer and apply their learning on computing devices. Districts and schools need to provide consistent access to the tools needed for computer science education. Local educational agencies will need to consider the technological upgrades needed to support students. The number of devices available for computer science learning will need to be sufficient to meet students' needs. Furthermore, increasing the number of students who learn computer science in classrooms will require upgrades to the network so students are able to connect simultaneously without negatively impacting their learning experience with poor connections and bandwidth issues. Local educational agencies will need to plan for an increase in computer network components and properly maintain their local operating system and the network operating system.

Policy

Policies impact implementation rates and help districts and schools prioritize initiatives. Efficient policies designed to expand access to computer science can increase the number of elementary and secondary students who are able to access rigorous, high quality computer science learning opportunities. Policies regarding state plans, funding models, state standards, high school graduation requirements, building infrastructure, and teacher preparation can address foundational educational needs in expanding access to computer science for all students. The recommended policies emphasize potential changes across entire states to ensure that computer science becomes a core component of education for all students with equitable access to learning opportunities across K–12 systems.

Comprehensive state plans for K–12 computer science can address perceptions and messaging in computer science. States can make computer science a part of the state’s educational priorities through articulated goals and strategies for achieving those goals. States can also create timelines with explicit action items to implement the strategies and help local educational agencies achieve the goals. The state plan can also include a state-level computer science administrator who is able to help scale statewide support for computer science education and partner with local educational agencies to provide guidance on implementation plans and professional learning. The position could help districts draft appropriate implementation plans that fit the needs of their community while providing input from a governance perspective that aligns with the state department of education’s mission and vision for computer science education.

Designate funding to support computer science. Providing dedicated funding to scale computer science education at state levels can help expand access for all students. State funding for computer science should be marked for professional development, teacher support, and instructional materials. Local educational agencies should be provided funding to provide professional development opportunities to grow teacher capacity in their organizations. State funding for high-quality professional development helps existing teachers teach computer science rather than having to hire new computer science teachers. Local educational agencies can also utilize funding to appoint a computer science coach or specialist who is able to coach teachers. A district employee who would be able to provide support and guidance as they develop their pedagogy in computer science would help build teacher efficacy and future computer science leaders.

Adopt state K–12 computer science standards. State standards for computer science can provide equitable and foundational expectations for student learning. Standards can also provide learning opportunities for all students rather than specific subsets of students who may have had additional resources outside of school to learn the content. Standards can also provide a roadmap for educators that define computer science education with a unified vision on what should be taught to students.

State standards are the basis for what students should know and be able to do. Standards can help explicitly separate computer science concepts from digital literacy, information technology, and general educational technology goals. Digital literacy provides direction on how to teach students in using technology as a resource. Computer science is distinctly different as the focal learning objectives are centered on creation through technology. States should draft and approve discrete computer science standards to better provide guidance for all educational stakeholders.

In states with adopted state standards for computer science, communication protocols should be clearly defined and implemented to build awareness and understanding of the standards. Local educational agencies and their administrators and teachers will need consistent access to communication channels from their state education departments.

Appoint a state-level computer science administrator. State departments of education should create and appoint a computer science administrator to lead policy initiatives and reflect states' commitments to equitable computer science education for all. The state-level computer science administrator can work with local educational agencies to align district computer science plans, develop and implement a strategic state plan, and collaborate with other state leaders on best practices. The creation and appointment of a state-level computer science administrator would also message to local educational agencies that computer science education is an important content area that should be provided for all students across K–12 education.

Preservice teacher preparation programs. Schools of education prepare teacher candidates to enter classrooms with knowledge of content and pedagogy. If computer science is a foundational part of students' curriculum and learning experiences, teacher candidates should receive access to relevant computer science courses in their programs. Schools of education should offer courses that include computer science content, pedagogical strategies, and state standards for computer science. Teacher candidates who have the responsibility to teach computer science standards will need to have an understanding of what the standards are and how they can be taught. Schools of education can also have a computer science education specialist as they do for core content areas in mathematics and English language arts. The specialist can teach courses in preparation programs and also provide coaching to teacher candidates during their student teaching experiences.

Require computer science in all secondary schools. Although there has been an increase of course offerings, many high schools across the country still do not offer computer science courses (Margolis et al., 2012). Given the impact computer science has on ensuring that all students are able to compete in a digital and global economy, creating a shift in policy were all high schools are required to offer at least one computer science course can address issues of access and equity. If computer science is recognized as a foundational skill for all students, policy changes should also require students to take at least one credit of computer science to meet graduation requirements.

Secondary schools can approach this policy with a flexible stance designed to meet the needs of their communities. This means computer science could be considered as a single, distinct subject on student transcripts, or may be written across course catalogs to satisfy a requirement for a mathematics, science, or technology credit. Students should be able to receive credit in computer science in any existing core credit requirement to provide local educational agencies the flexibility to expand computer science education and increase enrollment across all student groups.

Creating a policy change in requiring all students to take at least one high quality computer science course to meet graduation requirements would impact middle schools and elementary schools. The teachers and administrators in elementary and middle schools would need to ensure that their students are prepared to be able to take a computer science course at the high school level. Recognizing issues of access and equity, creating a graduation mandate rather than an option could push down opportunities for students to learn computer science. All elementary and middle schools would have to make sure their students were prepared to take the high school computer science course so that they can graduate with their peers.

Allow computer science to satisfy higher education admissions. Elementary teachers prepare students for college, career, and world readiness. The foundational learning that occurs in elementary levels helps students prepare for the future. States and districts could work with institutions of higher education to align coursework required for student admissions. Admission policy changes to have computer science meet mathematics or science entrance requirements could help increase learning opportunities for all students. Aligning policies where students have an option to learn computer science outside of an elective credit could encourage students to explore computer science courses earlier. Elementary teachers would also understand teaching computational literacy in elementary levels would help students in later years meet core credit requirements for high school graduation.

Constructivist Theory

The findings of this study suggest that elementary teachers construct meaning through their experiences teaching computer science. The participants experienced building content and pedagogical knowledge in computer science based on their own learning and reflections. The study indicates that teachers build their knowledge in computer science through a combination of past experiences, interaction, and current learning (Dev, 2016).

Participants constructed knowledge through social interaction with other elementary teachers teaching computer science (Ferguson-Patrick, 2018; Papan & Sompong, 2012; Wild, 2015). The findings revealed the impact the Internet can have in providing social, collaborative learning opportunities with professional learning networks. Utilizing professional learning networks, teachers were able to construct new beliefs and build upon their own learning and perceptions they acquired in their roles as elementary computer science teachers. Teachers found a mechanism to address the external barrier of limited support systems they felt they had access to in the organization. Teachers built and constructed knowledge by engaging with other educators through collaborative discussions, connecting with colleagues all over the country who had shared experiences teaching computer science (Papert, 1980). The findings support the constructivist learning theory that learners build and form knowledge through productive exchanges with others.

Teachers in the study found difficulty collaborating with their colleagues and generally found increased perceptions of success through their own professional learning networks. Providing teachers with support on establishing and maintaining successful professional learning communities on their school sites could help build teachers' collaboration skills and strategies in computer science. Professional learning communities can foster collaborative learning among colleagues and can be used to form working groups of practice-based learning. As teachers learn how to collaborate and plan for computer science with their grade levels or teams, they can build up their knowledge and increased feelings of efficacy as constructivist learning happens through building things that are tangible and shareable (Ackermann, 2001; Dev, 2016).

Participants in this study indicated utilizing constructivist teaching strategies as a coping mechanism for the some of the internal barriers they experienced teaching computer science. Teachers had shared experiences in shifting their roles as a provider of direct instruction to a facilitator of learning. Teachers felt they did not have the content knowledge in computing to use traditional teaching models and found success in allowing students to become active learners who discovered new knowledge through collaborating and communicating with their peers (Gupta & Gupta, 2017).

Recommendations for Further Research

The velocity of movement in computer science education reveals a need for further research. My recommendations for further research include the following: This qualitative case study should be replicated. More participants should be included in further research. Classroom observations should be included in further research to add to this study. Videos can add an additional layer of valuable data for future research into elementary teachers' experiences with computer science.

Replicating the Study

This qualitative case study should be replicated on a larger scale. The study should be replicated with more schools and districts in California as well as other states. Increasing the scope of this study will help find identify dominant themes that are shared across multiple settings. Replicating this study with more elementary teachers will add to the limited body of research that currently exists. The addition of shared experiences can help reveal additional barriers teachers experience teaching computer science and potential coping mechanisms that were not discovered in this study.

Replicating this case study with additional participants could lead to additional themes that were not uncovered in this study. Qualitative studies require sample sizes large enough to obtain enough data to adequately describe the phenomenon and answer the research questions (Creswell, 2013). A larger sample could provide more reliable data and would be more likely to be representative of the population (Creswell, 2013). Furthermore, outliers are much more likely to be removed. Replicating this study with an increased sample size may broaden the findings of possible data and help future researchers form a better picture for analysis.

Observations

Direct observations of elementary teachers teaching computer science can add an additional source of evidence in future studies. An observation protocol can be utilized to guide formal data collection and provide rich information about elementary teachers' experiences. Observing the interactions between teachers and students can help future researchers gain an even deeper understanding of elementary teachers' experiences (Creswell, 2013). Future researchers could use observations to find data that would be less likely to come to the surface using interviews (Hatch, 2002). Sensitive information that participants may not want to discuss could also be uncovered by utilizing observational data in future research (Hatch, 2002). The inclusion of observations in future studies would require member checking of observations to help ensure the reliability of the collected data (Creswell, 2013).

Future researchers who implement observations will need to consider the time commitment required for adding in this form of data collection and analysis. Scheduling times and observing participants will require a significant amount of time. Additionally, researchers will need to consider the impact they can have on observations as the act of observations may prompt changes in participants' activities (Merriam, 2009). One way to minimize the effect on participants is to build trust with the observation participants which requires an additional time commitment.

Videos

Image-based research could be considered for future research. Video recordings can be a powerful medium to capture data and improve future studies. Video recordings are possible with technological improvements but have been underutilized for data collection because of confidentiality issues (Hatch, 2002). Video based research can work as a documentary function that can supplement fieldnotes and improve the chances of capturing nuances and complexities (Hatch, 2002). Video data can inform future studies as the data can provide researchers insight into the alignment of participants' self-assessments and observable behavior in their natural setting.

Many of the participants discussed how they used online videos to build their own knowledge and create informal learning opportunities. Videotaping elementary teachers teaching computer science in their classroom could be used to share how teachers respond to the barriers they experience in their classrooms. The videos could be played back for participants in future studies and their discussions could also be recorded as an additional layer of data (Tobin, Wu, & Davison, 1989).

Additional Recommendations

Future researchers should be sensitive to the nature of elementary schools and the expectations on teachers to be able to teach multiple content areas. To further expand this study, future research could be based on the experiences of teachers who utilize pedagogical shifts in computer science to other content areas. The ability of elementary teachers to provide students with new learning experiences in other content areas with an increase in voice and agency could be a part of future studies. Participants in this study noted changes in their mindset teaching computer science as part of a coping mechanism to counter the challenges they experienced teaching the content. It was not revealed whether the teachers internalized the shift in pedagogical strategies teaching computer science and applied the learning to their daily practice.

Future studies could research teachers’ abilities to transfer computer science teaching methods as a part of their pedagogical content knowledge to manage learning content in other subject areas.

Conclusion

The purpose of this study was to understand the experiences of elementary teachers teaching computer science. The key findings were that the participants experienced multiple barriers to teaching computer science. The participants described coping mechanisms to address the barriers they faced and discussed the unique challenges of teaching computer science in elementary grades.

In this chapter I discussed the results of the study and provided details and context of the findings with answers to the research questions. I was able to identify a number of themes emerging from the data that included specific external and internal barriers teachers experienced teaching computer science. Elementary teachers discussed the impact of organizational structures as external barriers they could not directly control within their roles. Teachers also provided information on the internal barriers they experienced teaching computer science as they could work to limit the impact of those challenges. The participants in this study identified coping mechanisms.

This dissertation has addressed the limited research on elementary teachers’ experiences teaching computer science through the lens of constructivism. The findings and discussion in this study contribute to the current available research and can be used to help guide future studies. The methodology of qualitative case study was utilized to learn more about this group of elementary teachers and to provide a rich, thick description of their experiences teaching computer science.

References

Ackermann, E. (2001). Piaget’s constructivism, Papert’s constructionism: What’s the difference? *Constructivism: Uses and Perspectives in Education*, 1–11. https://doi.org/10.1.1.132.4253

Allen, M., Webb, A. W., & Matthews, C. E. (2016). Adaptive teaching in STEM: Characteristics for effectiveness. *Theory into Practice*, *55*(3), 217–224. https://doi.org/10.1080/00405841.2016.1173994

American Psychological Association. (2010). *Ethical principles of psychologists and code of conduct.* Retrieved from http://www.apa.org/ethics/code/index.aspx

Ashcraft, C., & Breitzman, A. (2012). Who invents IT? An analysis of women’s participation in information technology patenting. Retrieved from www.ncwit.org/patentreport

Aslan, A. & Zhu, C. (2017). Influencing factors and integration of ICT into teaching practices of pre-service and starting teachers. *International Journal of Research in Education and Science, 2*(2), 359–370.

Bell, R. L., Maeng, J. L., & Binns, I. C. (2013). Learning in context: Technology integration in a teacher preparation program informed by situated learning theory. *Journal of Research in Science Teaching*, *50*(3), 348–379. https://doi.org/10.1002/tea.21075

Bender, E., Schaper, N., Caspersen, M. E., Margaritis, M., & Hubwieser, P. (2016). Identifying and formulating teachers’ beliefs and motivational orientations for computer science teacher education. *Studies in Higher Education*, *41*(11), 1958–1973. https://doi.org/10.1080/03075079.2015.1004233

Bernard, H. R. (1995). *Research methodology in anthropology: Qualitative and quantitative* *approaches*. Walnut Creek, CA: Alta Mira Press.

Blazar, D., & Kraft, M. A. (2017). Teacher and teaching effects on students' attitudes and behaviors. *Educational Evaluation and Policy Analysis, 39*(1), 146–170.

Bloomberg, L. D., & Volpe, M. (2012). *Completing your qualitative dissertation: A road map from beginning to end.* Thousand Oaks, CA: Sage.

Buckler, C., Koperski, K., & Loveland, T. (2018). Is computer science compatible with technological literacy? *Technology and Engineering Teacher, 77*(4), 15–20.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research, 87*(4), 834–860. https://doi.org/10.3102/0034654317710096

Burrows, L. (2013). Women remain outsiders in video game industry. *The Boston Globe.* Retrieved from http://www.bostonglobe.com/business/2013/01/27/women-remain outsiders-video-gameindustry/275JKqy3rF ylT7TxgPmO3K/story.html

Buss, R. R., Wetzel, K., Foulger, T. S., & Lindsey, L. (2015). Preparing teachers to integrate technology into K–12 instruction: Comparing a stand-alone technology course with a technology-infused approach. *Journal of Digital Learning in Teacher Education, 31*(4), 160–172. https://doi.org/10.1080/21532974.2015.1055012

Buzzetto-More, N., Ukoha, O., & Rustagi, N. (2010). Unlocking the barriers to women and minorities in computer science and information systems studies: Results from a multi-methodological study conducted at two minority serving institutions. *Journal of Information Technology Education*, *9*, 115–131.

California State Board of Education. (2018). *Computer science standards timelines*. Retrieved from https://www.cde.ca.gov/be/st/ss/cacompsciencetimeline.asp

Capacho, J. (2016). Teaching and learning methodologies supported by ICT applied in computer science. *Turkish Online Journal of Distance Education*, *17*(2), 59–73. https://doi.org/10.17718/tojde.48315

Carter, A. (2014). The implementation of social software in authentic literacy activities. *Inquiries Journal*, *6*(11), 1–4.

Cetin, N. I. (2017). Effects of a teacher professional development program on science teachers' views about using computers in teaching and learning. *International Journal of Environmental and Science Education*, *11*(15), 8026–8039. Retrieved from http://files.eric.ed.gov/fulltext/EJ1118317.pdf

College Board. (2016). *AP state report: California* [Data file]. Retrieved from https://research.collegeboard.org/programs/ap/data/participation/ap-2016

College Board. (2017). *AP Computer Science Principles course and exam description* [PDF document]. Retrieved from https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf

Computing At School. (2013). *Running a computing at school hub*. Retrieved from http://www.computingatschool.org.uk/data/uploads/Hub_OperationsManual.pdf

Creswell, J. W. (2013). *Qualitative inquiry and research design* (3rd ed.). Thousand Oaks, CA: Sage.

Cutts, Q., Robertson, J., Donaldson, P., & O'Donnell, L. (2017). An evaluation of a professional learning network for computer science teachers. *Computer Science Education*, *27*(1), 30–53. https://doi.org/10.1080/08993408.2017.1315958

Dabbagh, N., & Kitsantas, A. (2012). Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *Internet and Higher Education*, *15*(1), 3–8. https://doi.org/10.1016/j.iheduc.2011.06.002

DeJarnette, N. (2012). America's children: Providing early exposure to STEM initiatives. *Education*, *1*, 77–84.

Dev, M. (2016). Constructivist approach enhances the learning: A search of reality. *Journal of Education and Practice*, *7*(25), 59–62.

Epstein, J. L. (2011). *School, family, and community partnerships: Preparing educators, and improving schools*. Boulder, CO: Westview Press.

Equal Employment Opportunity Commission (2016). *Diversity in high tech.* Retrieved from https://www.eeoc.gov/eeoc/statistics/reports/hightech/

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2008). Contextualization of programming learning: A virtual environment study. In *2008 38th Annual Frontiers in Education Conference*. https://doi.org/10.1109/FIE.2008.4720544

Falkner, K., & Vivian, R. (2015). *Coding across the curriculum: Resource review.* Adelaide, AU: The University of Adelaide.

Ferguson-Patrick, K. (2018). The importance of teacher role in cooperative learning: The effects of high-stakes testing on pedagogical approaches of early career teachers in primary schools. *Education 3-13, 46*(1), 89–101. https://doi.org/10.1080/03004279.2016.1189946

Fluck, A., & Dowden, T. (2013). On the cusp of change: Examining pre-service teachers' beliefs about ICT and envisioning the digital classroom of the future. *Journal of Computer Assisted Learning, 29*(1), 43–52. https://doi.org/10.1111/j.1365-2729.2011.00464.x

Gal-Ezer, J., & Stephenson, C. (2010). Computer science teacher preparation is critical. *ACM Inroads, 1*(1), 61–66. https://doi.org/10.1145/1721933.1721953

Gauntlett, D. (2014). *LEGO studies: Examining the building blocks of a transmedial phenomenon.* New York, NY: Routledge.

Gershenson, S., Holt, S. B., & Papageorge, N. W. (2016). Who believes in me? The effect of student–teacher demographic match on teacher expectations. *Economics of Education Review, 52*, 209–224.

Giannakos, M. N., Pappas, I. O., Jaccheri, L., & Sampson, D. G. (2017). Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies, 22*(5), 2365–2382. http://dx.doi.org.cupdx.idm.oclc.org/10.1007/s10639-016-9538-1

Gibson, W. J., & Brown, A. (2009). *Working with qualitative data.* Thousand Oaks, CA: Sage.

Gill, P., Stewart, K., Treasure, E., & Chadwick, B. (2008). Methods of data collection in qualitative research: Interviews and focus groups. *British Dental Journal, 204*(6), 291–295. https://doi.org/10.1038/bdj.2008.192

Gillham, B. (2000). *Case study research methods*. London, England: Continuum.

Goldsmith, L. T., Doerr, H. M., & Lewis, C. C. (2014). Mathematics teachers' learning: A conceptual framework and synthesis of research. *Journal of Mathematics Teacher Education, 17*(1), 5–36. https://doi.org/10.1007/s10857-013-9245-4

Gómez, M. (2015). When circles collide: Unpacking TPACK instruction in an eighth-grade social studies classroom. *Computers in the Schools, 32*(3–4), 278–299. https://doi.org/10.1080/07380569.2015.1092473

Grandy, G. (2010). *Instrumental case study.* Thousand Oaks, CA: Sage.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A Review of the state of the field. *Educational Researcher, 42*(1), 38–43.

Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? An experiment with data saturation and variability. *Field Methods, 18*(1), 59–82.

Gupta, R., & Gupta, V. (2017). Constructivist approach in teaching. *International Journal of Humanities and Social Sciences, 6*(5), 77–88.

Guzdial, M. (2014). Meeting student and teacher needs in computing education. *Communications of the ACM*, *57*(12), 10–11. https://doi.org/10.1145/2682922

Harkinson, J. (2015, July 2). *The combined black workforces of Google, Facebook, and Twitter could fit on a single jumbo jet.* Retrieved from https://www.motherjones.com/politics/2015/07/black-workers-google-facebook-twitter-silicon-valley-diversity/

Harmon, J. (2018). *Educating for equity and access in computer science*. Retrieved from http://newsroom.ucla.edu/stories/educating-for-equity-and-access-in-computer-science

Hatch, J. A. (2002). *Doing qualitative research in education settings.* Albany, NY: State University of New York Press.

Heitin, L. (2016). *What is digital literacy?* Retrieved from https://www.edweek.org/ew/articles/2016/11/09/what-is-digital-literacy.html

Hepp, P. K. (2015). Teacher training: Technology helping to develop an innovative and reflective professional profile. *Universities and Knowledge Society Journal*. *12*(2), 30–43.

Hogan, A., & Roberts, B. (2015). Occupational employment projections to 2024. *Monthly Labor Review*. https://doi.org/10.21916/mlr.2015.49

Hur, J. W., Andrzejewski, C. E., & Marghitu, D. (2017). Girls and computer science: Experiences, perceptions, and career aspirations. *Computer Science Education*, *27*(2), 100–120. https://doi.org/10.1080/08993408.2017.1376385

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82,* 263–279.

Judson, E. (2006). How teachers integrate technology and their beliefs about learning: Is there a connection? *Journal of Technology and Teacher Education, 14*(3), 581–597.

Khoury, G. (2007). *CSTA certification committee report.* Retrieved from http://csta.acm.org/ComputerScienceTeacherCertification/sub/CertificationResearch.html

Krauss, J. & Prottsman, K. (2017). *Computational thinking and coding for every student: The teacher's getting-started guide*. Thousand Oaks, CA: Corwin.

Kvale, S. & Brinkmann, S. (2009). *Interviews: Learning the craft of qualitative research* *interviewing*. Los Angeles, CA: Sage.

Lachney, M. (2017). Computational communities: African-American cultural capital in computer science education. *Computer Science Education*, *27*(3–4), 175–196. https://doi.org/10.1080/08993408.2018.1429062

Ladner, R., & Israel, M. (2016). Broadening participation: For all in computer science for all. *Association for Computing Machinery. Communications of the ACM, 59*(9), 26–28.

Latham, G., & Carr, N. (2012). Building on authentic learning for pre-service teachers in a technology-rich environment. *Journal of Learning Design*, *5*(1), 66–77.

Lee, M. J., & Ko, A. J. (2011). Personifying programming tool feedback improves novice programmers' learning. *In proceedings of the seventh international workshop on computing education research, 11,* 109–116.

Level Playing Field Institute. (2015). Disparities in access to CS courses in California high schools. *Path Not Found, 1*. 1–16.

Liberman, N., Kolikant, Y. B. D., & Beeri, C. (2012). Regressed experts as a new state in teachers' professional development: Lessons from computer science teachers' adjustments to substantial changes in the curriculum. *Computer Science Education, 22*(3), 257–283. https://doi.org/10.1080/08993408.2012.721663

Love, T. S., & Strimel, G. J. (2016). Computer science and technology and engineering education: A content analysis of standards and curricular resources. *Journal of Technology Studies, 42*(2), 76–88.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12? *Computers in Human Behavior, 41,* 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Maehr, M., & Meyer, H. (1997). Understanding motivation and schooling: Where we've been, where we are, and where we need to go. *Educational Psychology Review, 9*(4), 371–409.

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning computational thinking and scratch at distance. *Computers in Human Behavior*, *80,* 470–477. https://doi.org/10.1016/j.chb.2017.09.025

Margolis, J., Estrella, R., Goode, J., Holme, J., & Nao, K. (2012). *Stuck in the shallow end: Education, race, education, and computing.* Cambridge, MA: MIT Press.

Marshall, C. & Rossman, G. (2011). *Designing qualitative research*. Thousand Oaks, CA: Sage.

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *Journal of Educational Psychology, 108*(3), 424–437. http://dx.doi.org.cupdx.idm.oclc.org/10.1037/edu0000061

Merriam, S.B. (2009). *Qualitative research: A guide to design and implementation.* San Francisco, CA: Jossey-Bass.

Merriam, S. B., & Tisdell, E. J. (2016). *Qualitative research: A guide to design and implementation* (4th ed.). San Francisco, CA: John Wiley & Sons.

Michalsky, T. (2012). Shaping self-regulation in science teachers' professional growth: Inquiry skills. *Science Education*, *96*(6), 1106–1133. https://doi.org/10.1002/sce.21029

Montoya, A. (2017). Computer science for all: Opportunities through a diverse teaching workforce. *Harvard Journal of Hispanic Policy, 29*, 47–62.

Mouza, C., Marzocchi, A., Pan, Y., & Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education, 48*(2), 84–104.

Ni, L., & Guzdial, M. (2012). Who am I? In *Proceedings of the 43rd ACM technical symposium on Computer Science Education, 12,* 499–504.

Ogunkola, B. J. (2008). Computer attitude, ownership and use as predictors of computer literacy of science teachers in Nigeria. *International Journal of Environmental and Science Education*, *3*(2), 53–57.

Ozturk, Z., Dooley, C. M. M., & Welch, M. (2018). Finding the hook: Computer science education in elementary contexts. *Journal of Research on Technology in Education*. Taylor and Francis Inc. https://doi.org/10.1080/15391523.2018.1431573

Page, M. B., & Margolis, R. L. (2017). Co-creating collaborative leadership learning environments: Using adult learning principles and a coach approach. *New Directions for Adult and Continuing Education*, *2017*(156), 77–87. https://doi.org/10.1002/ace.20261

Papan, N., & Sompong, N. (2012). A development of training model based on Constructivism theory for teachers under the jurisdiction of the basic education commission. *Procedia - Social and Behavioral Sciences, 64,* 665–670. https://doi.org/10.1016/j.sbspro.2012.11.078

Papanikolaou, K., Makri, K., & Roussos, P. (2017). Learning design as a vehicle for developing

TPACK in blended teacher training on technology enhanced learning. *International Journal of Educational Technology in Higher Education*, *14*(1), 1–14. https://doi.org/10.1186/s41239-017-0072-z

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York, NY: Basic Books.

Park, G. J., Johnson, H. J., Vath, R. J., Kubitskey, B. W., & Fishman, B. J. (2013). Examining the roles of the facilitator in online and face-to-face PD contexts. *Journal of Technology and Teacher Education*, *21*(2), 225–245.

Patrick, F., Elliot, D., Hulme, M., & McPhee, A. (2010). The importance of collegiality and reciprocal learning in the professional development of beginning teachers. *Journal of Education for Teaching: International Research and Pedagogy*, *36*, 277–289.

Patton, M. (2015). *Qualitative research & evaluation methods*. Thousand Oaks, CA: Sage.

Peng, J., Wang, M., & Sampson, D. (2017). Visualizing the complex process for deep learning with an authentic programming project. *Journal of Educational Technology & Society*, *20*(4), 275–287.

Peters, M. A., & Araya, D. (2011). Transforming American education: Learning powered by technology. *E-Learning and Digital Media*, *8*(2), 102–105. https://doi.org/10.2304/elea.2011.8.2.102

Piaget, J. (1952). *The origins of intelligence in children*. New York, NY: W.W. Norton.

Plourde, L., & Alawiye O. (2003). Constructivism and elementary preservice science teacher preparation: Knowledge to application. *College Student Journal*, *37*(3), 334–342.

Pourhosein-Gilakjani, A., Mei-Leong, L., & Nizam-Ismail, H. (2013). Teachers’ use of technology and Constructivism. *International Journal of Modern Education and Computer Science*, *5*(4), 49–63. https://doi.org/10.5815/ijmecs.2013.04.07

Richardson, V. (1997). *Constructivist teacher education: Building new understandings*. Washington, D.C.: Falmer Press.

Rogoff, B., & Lave, J. (1984). *Everyday cognition: Its development in social context*. Cambridge, MA: Harvard University Press.

Rotter, J.B. (1975). Some problems and misconceptions related to the construct of internal vs external control of reinforcement. *Journal of Consulting and Clinical Psychology, 43*(1), 36–67.

Ryoo, J., Goode, J., & Margolis, J. (2015). It takes a village: Supporting inquiry- and equity-oriented computer science pedagogy through a professional learning community. *Computer Science Education*, *25*(4), 351–370. https://doi.org/10.1080/08993408.2015.1130952

Saldaña, J. (2015). *The coding manual for qualitative researchers*. Thousand Oaks, CA: Sage.

Sanalan, V. A. (2016). Computerphobia in preservice teachers. *International Education Studies*, *9*(3), 217 –223. https://doi.org/10.5539/ies.v9n3p217

Sanders, M. (2006). *Building school-community partnerships: Collaboration for student success.* Thousand Oaks, CA: Corwin Press.

Schutt, R. K. (2018). *Investigating the social world: The process and practice of research*. Thousand Oaks, CA: Pine Forge Press.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K–12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351–380. http://dx.doi.org.cupdx.idm.oclc.org/10.1007/s10639-012-9240-x

Seidman, I. (2006). *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. New York, NY: Teachers College Press.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495.

Sentance, S., Humphreys, S., & Dorling, M. (2014). The network of teaching excellence in computer science and master teachers. *In Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 80–88). Berlin, DE: ACM.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*. Elsevier Ltd. https://doi.org/10.1016/j.edurev.2017.09.003

Skoretz, Y., & Childress, R. (2013). The impact of a school-based, job-embedded professional development program on teachers' efficacy for technology integration: Findings from an initial study. *Society for Information Technology & Teacher Education International Conference*, *21*(4), 461–484. Retrieved from http://www.editlib.org/p/39898/

Srikoom, W., Hanuscin, D. L., & Faikhamta, C. (2017). Perceptions of in-service teachers toward teaching STEM in Thailand. *Asia - Pacific Forum on Science Learning and Teaching, 18*(2), 1–23.

Stake, R. (1995). *Multiple case study analysis*. New York, NY: The Guilford Press.

Srisupawong, Y., Koul, R., Neanchaleay, J., Murphy, E., & Francois, E. J. (2018). The relationship between sources of self-efficacy in classroom environments and the strength of computer self-efficacy beliefs. *Education and Information Technologies*, *23*(2), 681–703. https://doi.org/10.1007/s10639-017-9630-1

Tang, C., Baer, J., & Kaufman, J. C. (2015). Implicit theories of creativity in computer science in the United States and China. *Journal of Creative Behavior*, *49*(2), 137–156. https://doi.org/10.1002/jocb.61

Teo, T., & Zhou, M. (2017). The influence of teachers' conceptions of teaching and learning on their technology acceptance. *Interactive Learning Environments*, *25*(4), 513–527. https://doi.org/10.1080/10494820.2016.1143844

Thomas, D. (2006). A general inductive approach for analyzing qualitative evaluation data. *American Journal of Evaluation*, *27*(2), 237–246.

Tobin, J. J., Wu, D. Y., & Davidson, D. H. (1989). *Preschool in three cultures: Japan, China, and the United States.* New Haven, CT: Yale University Press.

Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2006). *A model curriculum for K–12 computer science: Report of the ACM K–12 Task Force Computer Science Curriculum Committee* (2nd ed.). New York, NY: ACM.

Turkle, S. (1984). *The second self: Computers and the human spirit*. New York, NY: Simon and Schuster.

Twining, P., Raffaghelli, J., Albion, P., & Knezek, D. (2013). Moving education into the digital age: The contribution of teachers' professional development. *Journal of Computer Assisted Learning*, *29*(5), 426–437. https://doi.org/10.1111/jcal.12031

Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review, 88*(1), 26–52. https://doi.org/10.17763/1943-5045-88.1.26

Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (2016). Landscape of K–12 computer science education in the U.S.: Perceptions, access, and barriers. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 645–650.

Wild, A. (2015). Relationships between high school chemistry students' perceptions of a constructivist learning environment and their STEM career expectations. *International Journal of Science Education, 37*(14), 2284–2305. https://doi.org/10.1080/09500693.2015.1076951

Wilkerson-Jerde, M., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the DeltaTick Project. *Science Education, 99*(3), 465–499. https://doi.org/10.1002/sce.21157

Winch, C. (2013). Curriculum design and epistemic ascent. *Journal of Philosophy of Education, 47*(1), 128–146.

Windschitl, M., & Sahl, K. (2002). Tracing teachers' use of technology in a laptop computer school: The interplay of teacher beliefs, social dynamics, and institutional culture. *American Educational Research Journal, 39*(1), 165–205.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2017). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education*, *26*(4), 235–254. https://doi.org/10.1080/08993408.2016.1257418

Yin, R. (2010). *Qualitative research from start to finish*. New York, NY: The Guilford Press.

Yin, R. (2014). *Case study research, design and methods* (5th ed.). Thousand Oaks, CA: Sage.

Zendler, A., McClung, O. W., & Klaudt, D. (2015). A cross-cultural comparison of concepts in computer science education: The U.S.-Germany experience. *International Journal of Information and Learning Technology, 32*(4), 235–256. https://doi.org/10.1108/IJILT-05-2015-0014

Appendix A: Invitation to Participants

Dear potential participant:

This letter is an invitation to consider participating in a study I am conducting as part of my Doctoral degree at Concordia University–Portland. The purpose of this study is to gain a better understanding of elementary teachers' experiences teaching computer science. This email is an invitation to you to participate in this study. Your input may provide insight into how schools and districts can better support and prepare elementary teachers to teach computer science.

Participation is completely voluntary and you are free to withdraw at any point during the study. If you choose to participate, you would share documents such as lesson plans and collaboration notes regarding computer science instruction in your classroom. You will also be asked to participate in two interviews which will be scheduled to accommodate your schedule. You may decline to share any documents and/or answer any of the interview questions.

I will record the interviews with a mobile device and immediately upload them into a secure, password protected computer. I will have a duplicate copy of the recording on a password protected external hard drive. Both the laptop and hard drive will be locked in a secure cabinet inside my office. The files will be transcribed by me, the principal investigator, and the recordings on both the laptop and external hard drive will be deleted when the transcription is complete

All information is completely confidential and your name and any other identifiable information will be shared in the data or analysis resulting from the study. Any personal information you provide will be coded so it cannot be linked to you. Any name or identifying information you give will be kept securely via electronic encryption. There are no known or anticipated risks to you as a participant in this study.

If you have any questions about this study or need additional information, please contact me by email at [redacted].

I look forward to speaking with you and gathering your insight for this study.

Sincerely,

Steve Kong

Appendix B: Principal Contact Communication

Dear [redacted],

I am writing to ask for your approval to contact teachers at your school site to participate in a case study. The research study is titled, "A Case Study on Elementary Teachers' Experiences Teaching Computer Science."

The purpose of the study is to gain a better understanding of elementary teachers' experiences teaching computer science. This case study will require access to elementary teachers across the district who will share their perspectives. To focus the study, the participants will be pulled from a group of teachers who have previously received professional development in computer science. I will be conducting a primary interview, secondary interview, and requesting documents teachers may want to submit for the study. Your site currently has [redacted] teachers I would like to ask to participate in this case study. I will accept the first 10 teachers to respond to participate in the study.

The interview portions of the study will last no more than 45 minutes and will occur twice during the course of this research. Interviews will not interfere with teachers' instruction or planning times and will be conducted to best accommodate your teachers' preferences.

Based on the findings of the study, I hope to identify common challenges elementary teachers face with computer science, the strategies to develop to overcome those challenges, and perceived needs for support.

I will take measures for ethical protection of all participants by maintaining confidentiality, obtaining informed consent, and protecting participants from harm. The names of participants will be shielded from others at all times by codes and pseudonyms will be used in the written report. Furthermore, access to data will be limited to myself and supervising faculty members. Informed consent will be obtained from all participants through the attached consent form. No vulnerable populations will be included in this study.

Thank you for considering this request,

Steve Kong

Appendix C: Primary Semistructured Interview Questions

Participant Code:

Interview #:

1. How long have you been teaching in the school district?
2. How long have you taught total?
3. What is your current credential?
4. What supplemental authorizations do you hold?
5. What sort of professional development have you received?
6. What made you decide to teach computer science before the district adoption of state standards?
7. What previous experience did you have with computer science?
8. What did you expect the experience of teaching computer science would be like at elementary levels?
9. What challenges did you encounter trying to teach computer science?
10. What strategies did you develop to address those barriers?
11. What additional supports do you feel would help you in teaching computer science?
12. Is there anything else you would like to add?

Appendix D: Secondary Semistructured Interview Questions

Participant Code:

Interview #:

1. After analyzing all interviews, elementary teachers seem to face diverse and unique challenges with computer science education. Can you expand on some of the challenges you've experienced teaching computer science?
2. In your experience facing these challenges with computer science education, what do you think are some supports that would be most helpful for you?
3. Based on the interviews, elementary teachers seem to use a common curriculum for computer science education, can you talk about your experiences with what you're currently using and why you do or don't use additional resources?
4. How would building connections with other subjects help to increase computer science learning opportunities for students?
5. Teachers indicated needing to have their administrators go through training for computer science, can you talk more about why that would be important?
6. How important is it for teachers, administrators, and parents to explicitly understand the benefits for students to learn computer science?
7. Is there anything else you would like to include that was not discussed in our two interviews?

Appendix E: Consent Form

Research Study Title: A Case Study on Elementary Teachers' Experiences Teaching Computer Science
Principal Investigator: Steve Kong
Research Institution: Concordia University
Faculty Advisor: Dr. Edward Kim

Purpose and what you will be doing:
This research project is designed to collect information on the experiences of elementary teachers with computer science (CS) education. The purpose of this study is to identify and describe the perceived barriers of elementary teachers teaching computer science, the strategies they used to overcome those barriers, and perceived needs for support. You were chosen for this study because you have had exposure to computer science professional development. No one will be paid to be in this study. Enrollment will begin on September 7, 2018 and end on October 7, 2018.

Procedures:
If you agree to be in this study, you will be asked to participate in two interviews and provide documents that may be relevant to this study. Each interview will take between 30-45 minutes. You will also be asked to member check where you review and comment on the accuracy of the findings of the study. This should take no more than 30 minutes. Documents you may want to provide include curriculum, meeting notes, planning documents, or other items relevant to computer science in your classroom.

Logistics:
Each interview will be conducted in a location that is easiest for you. If you are unable to schedule a face-to-face interview, contact will be made via phone call.
Depending on your preference, the format for reviewing and commenting on the findings can take place over phone or via e-mail.

Risks:
There are no risks to participating in this study other than providing your information. However, we will protect your information. Any personal information you provide will be coded so it cannot be linked to you. Any name or identifying information you give will be kept securely via electronic encryption. When we or any of our investigators look at the data, none of the data will have your name or identifying information. We will only use a secret code to analyze the data. We will not identify you in any publication or report. Your information will be kept private at all times and then all study documents will be destroyed 3 years after we conclude this study.

Benefits:
With the release of California Computer Science State Standards, the findings from this study will benefit schools and districts identify the best possible ways to support teachers with implementing computer science education.

Confidentiality:
This information will not be distributed to any other agency and will be kept private and confidential. The only exception to this is if you tell us abuse or neglect that makes us seriously concerned for your immediate health and safety.

Right to Withdraw:
Your participation is greatly appreciated, but we acknowledge that the questions we are asking are personal in nature. You are free at any point to choose not to engage with or stop the study. You may skip any questions you do not wish to answer. This study is not required and there is no penalty for not participating. If at any time you experience a negative emotion from answering the questions, we will stop asking you questions.

Contact Information:
You will receive a copy of this consent form. If you have questions you can talk to or write the principal investigator, Steve Kong at email [redacted] If you want to talk with a participant advocate other than the investigator, you can write or call the director of our institutional review board, Dr. OraLee Branch (email obranch@cu-portland.edu or call 503-493-6390).

Your Statement of Consent:
I have read the above information. I asked questions if I had them, and my questions were answered. I volunteer my consent for this study.

______________________________ __________
Participant Name Date

______________________________ __________
Participant Signature Date

______________________________ __________
Investigator Name Date

______________________________ __________
Investigator Signature Date

Investigator: Steve Kong email: [redacted]
c/o: Professor Dr. Edward Kim
Concordia University–Portland
2811 NE Holman Street
Portland, Oregon 97221

Appendix F: Researcher Created Program

```
function findText(item) {
  //generate a color and save it in a variable to use as a highlight color
  var background = '#' + (Math.random().toString(16) + "000000").substring(2,8)

  //add the parameter to the log to test that it is working as expected
  Logger.log(item)

//create the variable to show the results of finding the text
  var searchResult

  //search for the text stored in the item variable
  searchResult = DocumentApp.getActiveDocument().getBody().findText(item)

  //log the results of the search for testing
  Logger.log(searchResult)

  //while the search results for very are not empty do the following
  while (searchResult !== null) {

  //highlight the search results in red
```

```
    searchResult.getElement().asText().setBackgroundColor(searchResult.getStartOffset(),
searchResult.getEndOffsetInclusive(), background)

    //find text again that is stored in the item variable
    searchResult= DocumentApp.getActiveDocument().getBody().findText(item,searchResult)

  //end of the loop
  }

}

function highlightProblem(){
  //create an array to store all of the words to search
  var words = ["very", " so ", "totally", "really", "any", "in other words", "its", "any"]

  //find each item in the array
  words.forEach(findText)

}
```

Appendix G: Statement of Original Work

The Concordia University Doctorate of Education Program is a collaborative community of scholar-practitioners, who seek to transform society by pursuing ethically-informed, rigorously-researched, inquiry-based projects that benefit professional, institutional, and local educational contexts. Each member of the community affirms throughout their program of study, adherence to the principles and standards outlined in the Concordia University Academic Integrity Policy. This policy states the following:

Statement of academic integrity.

> As a member of the Concordia University community, I will neither engage in fraudulent or unauthorized behaviors in the presentation and completion of my work, nor will I provide unauthorized assistance to others.

Explanations

What does "fraudulent" mean?

> "Fraudulent" work is any material submitted for evaluation that is falsely or improperly presented as one's own. This includes, but is not limited to texts, graphics and other multi-media files appropriated from any source, including another individual, that are intentionally presented as all or part of a candidate's final work without full and complete documentation.

What is "unauthorized" assistance?

"Unauthorized assistance" refers to any support candidates solicit in the completion of their work, that has not been either explicitly specified as appropriate by the instructor, or any assistance that is understood in the class context as inappropriate. This can include, but is not limited to:

- Use of unauthorized notes or another's work during an online test
- Use of unauthorized notes or personal assistance in an online exam setting
- Inappropriate collaboration in preparation and/or completion of a project
- Unauthorized solicitation of professional resources for the completion of the work.

Statement of Original Work (continued)

I attest that:

1. I have read, understood, and complied with all aspects of the Concordia University- Portland Academic Integrity Policy during the development and writing of this dissertation.

2. Where information and/or materials from outside sources has been used in the production of this dissertation, all information and/or materials from outside sources has been properly referenced and all permissions required for use of the information and/or materials have been obtained, in accordance with research standards outlined in the *Publication Manual of The American Psychological Association*

Signature

Steve Kong
Name (Typed)

5-13-2019
Date