PREPARING TEACHERS IN DEVELOPING COUNTRIES FOR COMPUTATIONAL THINKING TEACHING IN PRIMARY EDUCATION: A NAMIBIAN CASE STUDY

by

Maria Ausiku

19262885

Submitted in fulfilment of the requirements for the degree

PhD in Information Technology

in the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT & IT

at the

UNIVERSITY OF PRETORIA

Study Leader:

Prof. MC Matthee

Date of submission

November 2022

ABSTRACT

In recent years, many countries in the developed world, have introduced computational thinking (CT) teaching in compulsory education, with few developing nations following. The introduction to teaching CT brought many challenges for teachers because these computing skills were not part of their initial teacher training and were less understood. Several professional development programmes have been developed to train teachers on the new CT content, but few studies have investigated the preparation of primary school teachers to teach CT and the impact of this training on the teachers' understanding of CT concepts and self-efficacy in a developing country context.

The main objective of this study was to develop a Professional Development for Primary School Teachers for the CT (PD4PCT) framework that can be used by training providers and researchers to integrate CT into teachers' professional development programmes. Constructionism was a pedagogical framework for this interpretive study and the conceptual frameworks of Desimone and three existing professional development CT frameworks (3C, CTTD and ADAPPTER). Different data collection methods were used for a single interpretive case study to investigate the impact of a professional development programme on primary school teachers (*n = 14),* their CT knowledge, beliefs and attitudes and self-efficacy of CT using a participatory design approach. Data was collected through a literature review, pre- and post-questionnaires, semi-structured interviews, and self-reporting journals. Expert reviewers validated the framework through an online questionnaire.

The study's findings indicated that teachers who participated in the professional development programme have considerably increased their CT knowledge, their beliefs and attitudes towards CT altered for the better, and they had a substantial rise in confidence to teach CT. Overall, the results indicate that most teachers can design lesson plans and activities incorporating algorithms, decomposition, and pattern recognition concepts but abstraction and debugging to a lesser extent. Subject matter knowledge of teachers influences the integration plans for certain CT topics.

To address the challenges teachers face in integrating CT into classrooms, the framework assists in identifying the components that must be considered to develop an effective professional development programme for teachers. The context of the school plays a vital role and should be considered as a first step in designing a teacher's professional development intervention. School leadership should support teachers with a collaborative environment where teachers can share CT knowledge and teaching strategies with others.

Keywords: *Computational Thinking (CT), professional development, primary teachers, unplugged, programming, participatory design, constructionism*

PUBLICATIONS FROM THE THESIS

Ausiku, M., & Matthee, M. (2021). Preparing Primary School Teachers for Teaching Computational Thinking: A Systematic Review. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12511 LNCS* (pp. 202–213). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-66906-5_19

DECLARATION

I declare that the work presented in this thesis is my own and has not been previously submitted at any other institution for the purpose of earning a degree. All of the references used in this work have been properly cited.

Signature: ________

Date: *21 November 2022*

DEDICATION

This thesis is dedicated to my parents Stella-Maris Andowa and Gosbertus Ausiku.

ACKNOWLEDGEMENTS

I want to express my gratitude to my PhD supervisor Prof. Machdel Matthee for her unwavering and positive support. Without her patient, hopeful encouragement and wise guidance, my study might have lost its way. Your support helped me to persevere, so thank you.

I would like to thank my family and friends for their tremendous support and understanding during my studies. I want to thank my mother for instilling in me a passion for learning and for always being curious about my progress along this journey.

I would like to thank my participants; the teachers and the experts who made this research feasible. I am grateful that you volunteered your time and knowledge to assist me in completing this research study.

Thank you to Dr Simon Akpo for inspiring me to pursue the PhD journey.

TERMS AND ABBREVIATIONS

ACW	Africa Code Week
CAPS	Curriculum and Assessment Policy Statement
CITE	Contemporary Issues in Technology and Teacher Education
CK	Curriculum knowledge
CoP	Community of Practice
CPD	Continuous professional development
CS	Computer Science
CSTA	Computer Science Teacher Association
CT	Computational thinking
CTS	Computational thinking scales
CTTD	Computational thinking teacher development
DLCS	Digital Literacy and Computer Science
EDR	Educational Design Research
GC	Graphic calculator
HoD	Head of departments
IC	Information and Communication
IJSR	International Journal of Science and Research
LPP	Legitimate peripheral participation
NIED	National Institute for Educational Development
OCD	Online co-design
PBL	Project-Based Learning
PCK	Pedagogical content knowledge
PD	Professional development
PD4PCT	Professional Development for Primary Computational Thinking

PECT	Progression of Early Computational Thinking
PK	Pedagogical knowledge
PROFILES	Professional Reflection Oriented Focus on Inquiry-based Learning and Education
SA	Strongly Agree
SD	Strongly Disagree
STEBI	Science Teaching Efficacy Belief Instrument
STEM	Science, technology, engineering, and mathematics
TCK	Technological content knowledge
TDC	Teacher Development Courses
TK	Technological Knowledge
TPACK	Technological pedagogical content knowledge
TPCK	Technological pedagogical content knowledge
TSECT	Teaching Self-Efficacy for Computational Thinking

Table of Contents

TABLE OF FIGURES

LIST OF TABLES

1 CHAPTER 1: INTRODUCTION

1.1 Background information

It is imperative for today's learners to attain Computational Thinking (CT) skills early in compulsory education to prepare them for a technologically advanced future workplace (Barr & Stephenson, 2011).

CT has come to the forefront in the past decade since the seminal article of Wing (2006), where she declared it a fundamental skill for the 21st century that everyone should possess. However, CT is not new; it is traced back to Papert's work on the LOGO programming language, where he prompted kids to utilise the computer, empowering them to create procedural reasoning utilising programming (Papert, 1980). Webb et al. (2017) contended that CT abilities are among the critical thinking approaches that enable students to discover solutions in any field.

CT includes concepts, such as algorithms, abstraction, decomposition, and pattern recognition largely from the discipline of Computer Science shared crosswise over different domains, such as science, mathematics, social science, language, and arts (Kafai & Burke, 2013; Lye et al., 2014). Many countries, such as the United States of America (USA), United Kinngdon (UK), Estonia, Finland and Malaysia have since introduced Computational Thinking skills in their compulsory education to prepare learners for future digitalised employment to enable them to think differently using computational concepts (Angeli, Voogt et al, 2016; Ung et al., 2022).

In Africa, numerous governments (Kenya, Mauritius, South Africa, Botswana, Zambia, etc,.) have concentrated on creating national ICT policies and approaches for ICT in education. Nonetheless, these ICT approaches are not constantly joined by detailed execution plans or the duty of the government to actualise them (Burns et al., 2019; Yonazi et al., 2012). While most African countries (Tanzania, Uganda, South Africa, Seychelles, etc,.) have introduced ICTs in the education system, the development and adoption stay moderate because of an absence of powerful ICT approaches and a sustainable supporting ICT foundation (e.g., power, Internet connection and computer devices, teacher capacity, and lack of finance) (Barakabitze et al., 2015; Nyakito et al., 2021). UNESCO has reported that in most Sub-Saharan African countries, the use of ICTs within education is at an early stage, and the percentage of essential infrastructure is under 15% in primary schools (Wallet, 2015). Only Seychelles, Mauritius, and South Africa were at the infusing stage regarding ICT implementation and use in education (Tilya, 2018). The region's main challenges of insufficient budget to sustain ICTs in schools, insufficient ICT infrastructure, untrained teachers and technicians on ICT integration, and ineffective coordination of ICT initiatives result in an unfriendly curriculum that lacks direction (Tilya, 2018). In South Africa, the Department of Basic Education introduced a new coding and robotics curriculum for Grades R to 9, hoping the curriculum will develop learners' ability to solve problems, think critically, and work collaboratively and creatively, which is essential for CT (Fares et al., 2021).

In their quest to reach 10 million people in Africa through the introduction of CT in schools, Google has joined forces with SAP, UNESCO and other key partners in supporting SAP Africa Code Week (ACW) to capacitate teachers and learners throughout the African continent, including Namibia. Over 21,000 teachers and 1.5 million youth were introduced to coding during the 6th edition of ACW in 2020 through virtual and in-person workshops (SAP, 2021).

Consistent with one of the objectives of Namibia Vision 2030 *(a national development blueprint)*, namely *"to integrate ICT education and training into the education and training system,"* the Ministry of Education adopted an ICT Policy for Education in 1995 through the National Institute for Educational Development (NIED) which was revised in 2000 and was under review in 2019. The ICT Policy for Education covers three aspects of the ICT curriculum: ICT Literacy, ICT as a subject, and Cross Curricular ICT (Ministry of Education, 2005).

The current Namibian national ICT curriculum reviewed in 2015 is structured as follows:

In the **junior primary phase (Grades 1-3)**, ICT is taught as a cross-curricular theme. The main activities are the use of educational games, educational software and educational videos.

In the **senior primary phase (Grades 4-7)**, the subject Information and Communication (IC) is compulsory. The senior primary syllabus focuses on the following computer skills: basic computer equipment and keyboard and mouse skills; navigating the operating system; file and folder management.

In the **secondary phase (Grades 8-12)**, Computer Studies is taught as an elective subject covering topics such as number systems (decimal, binary and hex), logic gates and the writing and interpreting of algorithms (National Institute for Educational Development, 2015).

1.2 Problem statement

Introducing CT in compulsory education has increased recently as countries around the world modified their curricula to integrate the subject starting at the primary level (Balanskat & Engelhardt, 2015; Brackmann et al., 2016).

However, in Namibia, similar to many other developing countries, the curriculum offers computer studies at the secondary level only as an optional subject, and it is silent at the primary level. CT skills are less understood by teachers at the primary level. Although Namibia has highlighted the significance of using ICT in its ICT Policy for Education, the government cannot afford to buy computer equipment for all schools and train teachers to acquire the necessary ICT skills (Kukali, 2013).

While many educators realise the need to integrate CT into compulsory education, most in-service teachers are not well prepared to teach CT (Nordby et al., 2022; Ung et al., 2022). This is because there is a shortage of qualified teachers to teach CT and programming to students (Barr & Stephenson, 2011; Voogt, Fisser, Good, Mishra, & Yadav, 2015).

Research has been carried out that focuses specifically on teachers and the difficulties they face when teaching CT concepts (Bower et al., 2017; Grover & Pea, 2013; Saidin et al., 2021; Sentance & Humphreys, 2015). Preparing teachers to understand the concepts of CT (i.e., learning how to link them to their own and other domains) and overcome these challenges will increase teachers' self-efficacy. Without knowing how concepts such as abstraction, decomposition, algorithmic thinking and debugging apply in the world; teachers cannot be expected to teach these things effectively and confidently. It would be unreasonable to expect teachers to incorporate CT concepts into their practice without support and opportunities to apply these ideas to authentic tasks (Saidin et al., 2021; Yadav et al., 2014).

Several frameworks for the infusion of CT into the school curriculum have been suggested that can support teachers (Jocius, Joshi, Dong, Robinson, Cateté, et al., 2020; Kirwan et al., 2022; Kong & Lai, 2021). One framework focuses specifically on primary schools (Kong & Lai, 2021). However, these studies proposed frameworks in the context of developed countries. A professional development framework for the infusion of CT into primary education is needed for developing countries because of different development context and challenges. The unique problems experienced by these countries, and the specific contexts must be considered when planning teachers' professional development interventions.

1.3 Research questions

The study is looking to answer the following research questions.

1.3.1 Main research question

How should in-service teachers be prepared to teach CT in primary schools in a developing country context?

1.3.2 Sub-research questions

The following sub-questions are supporting to answer the main research question.

1. What is the change in teachers' beliefs and attitudes towards CT resulting from a professional development program in the Namibian context?
2. What are the findings from previous research on professional development of teachers to teach CT?
3. How can in-service teachers participate in the design of learning material for CT in primary schools?
4. What are the components of a framework for the professional development of primary school in-service teachers for teaching CT?

1.4 Research Objectives

The objectives of this research are to:

- evaluate how teachers' knowledge, beliefs and attitudes change after participating in a professional development programme.
- explore how in-service teachers can be supported to integrate CT concepts into existing subjects.
- explore how in-service teachers can be design partners in a professional development programme.
- develop and evaluate a professional development framework that can guide teachers in integrating CT into existing subjects at the primary level in a developing country's context.

1.5 Justification for the research

While the introduction of CT is mostly targeted at the secondary and university level, there's a growing indication from researchers that CT should start as early as primary school (Duncan, 2019; Tran, 2018). There is a need to have a theoretical and pedagogical model that underpins CT concepts at the primary level as a starting point to develop interventions for promoting CT skills in developing countries. Recently, two frameworks were suggested that focus on primary education but in a developed context (Jocius et al., 2020; Kong & Lai, 2021).

For CT to be infused into compulsory education, teachers need to be educated about what CT skills are, how they relate to their subject-specific curriculum, and what they do on a day-to-day basis. Although some work has been done on teacher training at the high school and tertiary levels, less attention has been paid to integrating CT into primary education (Yadav, et al., 2019). Several researchers agree there is still little research done on integrating CT in education and that most primary school teachers lack the knowledge of content and pedagogy of CT (Ng, 2017; Rich et al., 2017; Saidin et al., 2021; Stanton et al., 2017; Voogt et al., 2015). In addition, Yadav et al. (2011) have eluded that few studies have methodically and widely investigated the integration of CT in pre-service and in-service teachers' training programmes.

The way to successfully expose learners to CT is when teachers are engaged in the activities and decision-making process, as it is complex and needs fundamental change (Barr & Stephenson, 2011). With changes in the curriculum, educators unavoidably will face challenges. If teachers do not feel adequate for showing computational reasoning, learners will likely have a negative experience learning the ideas (Israel et al., 2015). Also, there is no far-reaching understanding of techniques for instructing and evaluating the dimension of computational reasoning improvement in students (Brennan & Resnick, 2012). Teachers will likely not know what teaching methods are suitable for teaching CT.

Past studies on CT have generated different definitions of what CT is and its concepts (Barr & Stephenson, 2011; Selby, 2014; Sung et al., 2017). Others have developed CT frameworks, where the most used method to apply CT is through programming languages, such as Scratch (Angeli et al., 2016; Brennan & Resnick, 2012; Curzon et al., 2014).

In a developing context like Namibia, the primary school level has no Computer Science-related subjects yet, and most schools do not have computer equipment. Therefore, the questions this thesis attempts to address are how CT can be infused into primary school education in a developing world context and, more importantly, how teachers can be prepared to implement it.

1.6 Expected contribution

This research contributes to the development and assessment of professional development programmes for primary school teachers in developing countries who are learning to teach and incorporate CT. There has been little research in this area, and this research is important because CT is becoming mandatory to teach in many nations that are changing or have modified their primary school curricula. Computing education academics, professional development providers, and educational authorities will require evidence to guide how they create, administer, select, and evaluate professional development programmes designed to prepare teachers for these evolving curricula.

The outcomes of this exploratory research resulted in a relevant and tested framework that will assist teachers and schools in incorporating CT into their primary level teaching. This approach will also address educators' difficulties when incorporating CT into their curricula.

These contributions have the potential to impact the design and execution of professional development programmes for primary school teachers studying CT, and this work has practical implications for future research and development of professional development programmes.

1.7 Brief Chapter Overview

This thesis is divided into eight chapters.

Chapter 1: I describe the background of CT in primary school curricula, and the challenges and opportunities associated with teaching those skills in primary school. I list the research questions addressed in this thesis and explained the significance and contributions of my study.

Chapter 2: I examine the relevant literature on CT in primary school, using constructionism in educational research, the issues associated with teaching CT in primary school, and CT professional development programmes. I identify gaps in the published studies in the literature review, notably for studies on professional development for CT in developing countries.

Chapter 3: I explain how the proposed Professional Development for Primary Teachers for CT Integration Framework, and its components were developed.

Chapter 4: I describe how the underlying epistemological theory of my study, constructionism, influenced the methodologies utilised to answer the research questions. I describe the research design, the instruments utilised to measure the outcomes in my study, and how mixed methods analyses were employed to answer each research question. This chapter also includes data quality evaluation and ethical issues.

Chapter 5: I describe how I put the recommended framework into action. It describes the processes used during the interactive training session that was conducted using a single case study.

Chapter 6: The findings from the case study analyses undertaken to answer the research questions are presented.

Chapter 7: This chapter discusses the findings of the case study.

Chapter 8: This chapter presents the validation of the framework and its refinement.

Chapter 9: This chapter concludes with a discussion and summary of sub-sections. In those sub-sections, I examine how the conclusions from answering each of the study questions are linked or not with findings from the relevant literature.

2 CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

CT research at the primary school level has grown over recent years, but there is a critical need for studies to build a well-researched body of knowledge (Angeli et al., 2016; Jevsikova et al., 2019; Kjällander et al., 2021; Nordby et al., 2022; Tsarava et al., 2022). While most scholars agree on the value of incorporating CT into primary education, there are differing views on how this can be accomplished. According to Angeli et al. (2016), computing educational programmes in primary schools must concentrate on CT. The Computer Science Teachers Association and the International Society for Technology in Education (CSTA & ISTE, 2011) question, whether students can develop and use computational reasoning ideas in critical thinking activities as a major component of subject material other than Computer Science.

2.2 Definition of Computational Thinking

CT is a logical thinking technique that can be applied in various fields (Jacob & Warschauer, 2018). Wing described computational thinking in 2006 as, "*Solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental of Computer Science*" (Wing, 2006, p.33).

When her work became more widely recognised in various fields, the argument was explored while also being deciphered in various ways. Wing reimagined CT at that time as follows: "Computational *thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."* (Wing, 2011, p.1).

From her revised description, one can deduce that CT skills are not device-dependent but that a problem can be solved by a person with or without a computer using rational critical thinking skills.

Another meaning is that CT is a method of thought in which algorithms find solutions (Aho, 2012). The Royal Society describes CT as the use of Computer Science methods and tools to reason about and comprehend artificial and natural systems and processes (Royal Society, 2012).

Barr and Stephenson characterised CT in the sense of compulsory education as a problem-solving approach that uses a computer via automation and can be transferred and applied to various domains (Barr & Stephenson, 2011).

Denning (2009) argued that CT has a long and illustrious tradition of all sciences, not just Computer Science. He looked at how, since the 1950s, CT has existed as algorithmic thought – defined to solve problems by performing input and output conversions using algorithms.

The ISTE and the Computer Science Teacher Association (CSTA) have established a CT definition for use in compulsory education that highlights nine key concepts: *"data collection, data analysis, data representation, problem decomposition, abstraction, algorithms, automation, parallelization and simulation"* (CSTA & ISTE, 2011).

Another definition by Sysło and Kwiatkowska (2013) emphasises CT as a collection of high-level thinking skills capable of solving problems without programming.

Selby and Woollard (2014) described CT as a problem-solving method that incorporates thinking processes and uses decomposition, algorithms, abstractions, evaluation, and pattern recognition.

Lavigne, Presser, Rosenfeld, Wolsky and Andrews (2020) defined CT based on Wing's perspective as "a creative way of thinking that empowers individuals to be systematic problem-solvers, enabling them to identify problems, then brainstorm and generate step-by-step solutions that can be communicated and followed by computers *or* humans" (para. 2).

While there is no systematic description in the literature, according to Barr and Stephenson (2011) and Grover & Pea (2013), researchers generally agree that CT skills include algorithmic reasoning, exploring different levels of abstraction, breaking problems down into small parts, and presenting knowledge through models. CT can be taught with and without computers (Bell et al., 2009). CT is not programming, it is not a quick-fix solution, and is not even machine literacy (National Research Council, 2010).

Although CT does not imply programming, it implies understanding current technological advancements that will resolve problems across all domains (Bundy, 2007). CT and Computer Science (CS) are not synonymous, but they are related in that CS fundamentals are used to improve CT skills (Wing, 2006). CT not only prepares students for computing careers, but it also prepares them to be critical thinkers who can solve problems with or without a computer, whether in their personal, academic, or professional lives (CSTA & ISTE, 2011; Selby, 2014).

2.3 Computational Thinking Frameworks

Researchers and educators have used several frameworks and definitions to operationalise CT based on Wing's (2008) meaning. Barr and Stephenson (2011) created a model that illustrates the core CT principles and capabilities and how they can be integrated across domains. Data processing, data analysis, data representation and analysis, abstraction, analysis and model validation, automation, testing and verification, algorithms and procedures, problem decomposition, control structures, parallelisation, and simulation are examples.

Brennan and Resnick (2012) created a framework for developing CT skills focusing on programming using Scratch. The framework defines CT using three dimensions, namely, "computational concepts (the concepts designers engage with as they programme, such as iteration, parallelism, etc.), computational practices (the practices designers develop as they engage with the concepts, such as debugging projects or remixing others' work), and computational perspectives (the perspectives designers form about the world around them and about themselves)" (Brennan & Resnick, 2012, p.1).

Curzon et al. (2014) created a system to assist teachers in teaching CT by defining what CT is, how to teach it, and how to assess it. Description, principles, classroom strategies, and evaluation are the four interconnected developmental stages of the framework. The core CT concepts are described at the concepts level (algorithmic thinking, evaluation, decomposition, abstraction, generalisation). With examples, these ideas were related to classroom techniques. This framework is specifically for the new Computing curriculum in the UK.

Angeli et al. (2016) created a framework for CT curricula focused on abstraction, generalisation, decomposition, algorithms, debugging abilities, and competence metrics that progressed through educational levels. The framework aims to engage children as problem-solvers and thinkers by identifying a solution, automating it using algorithm thought, and applying pattern recognition to generalise it to other problems. The system aims to improve all CT skills in all elementary grades, though at varying levels of competence that will vary from school to school and classroom to classroom. It also addressed technological pedagogical content knowledge, detailing the body of knowledge that teachers must possess to teach CT in a K-6 setting.

Kalelİoğlu et al. (2016) created a paradigm that depicts CT as a problem-solving approach that can be applied to various learning situations. It all begins with the identification of a problem through abstraction and decomposition. Then, to comprehend and solve a problem, the method of collecting, describing, and analysing data should be used. Data collection and analysis, pattern recognition, conceptualising, and data representation are the key activities that should be considered during this phase. Some cognitive methods, such as parallelisation and algorithmic thinking, can provide more precise solutions. Automation, modelling, and simulations may apply these solutions while testing and debugging are used to validate the solution.

Via constructionism and social-constructivism theories, Kotsopoulos et al. (2017) created a pedagogical paradigm that includes unplugging, tinkering, creating, and remixing as pedagogical experiences. Unplugged tasks are those that are completed without a computer. During the tasks, the main goal of the creating experience is to create new things. Tinkering practices result in modifying existing objects while remixing activities are those in which objects are appropriated for new purposes.

Weintrop et al. (2016) developed a practice taxonomy that includes data practices, modelling and simulation practices, computational problem-solving practices, and systems thinking practices. Each activity is divided into five to seven subsets. Since engaging in scientific inquiry necessitates the expertise and skills unique to each activity, they are referred to as activities rather than skills or concepts.

The Progression of Early Computational Thinking (PECT) Model, developed by Seiter and Foreman (2013), is a model for understanding and evaluating CT in the primary grades (Grades 1 to 6). The model combines measurable evidence from student work with more abstract, wider coding design patterns, which are then translated into CT concepts.

Table 2.1 Summary of CT definitions and frameworks

	Abstraction	Algorithms	Data	Decomposition	Parallelisation	Testing and debugging	Generalising	Control structure	Simulating
Barr and Stephenson (2011)	✓	✓	✓	✓	✓	✓		✓	✓
Brennan and Resnick (2012)	✓	✓	✓	✓	✓	✓		✓	
Selby (2014)	✓	✓		✓			✓		✓
Grover and Pea (2013)	✓	✓	✓	✓	✓	✓		✓	
Angeli, Voogt et al. (2016)	✓	✓		✓		✓	✓		
Kalelİoğlu et al. (2016)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kotsopoulos et al (2017)	✓	✓		✓		✓			
Weintrop et al (2016)	✓	✓		✓		✓			
Curzon, Dorling et al (2014)	✓	✓		✓		✓			
Seiter and Foreman (2013)	✓	✓	✓	✓	✓				
The Royal Society (2012)	✓	✓		✓			✓		
CSTA and ISTE (2011)	✓	✓	✓	✓					
Sysło and Kwiatkowska (2013)	✓			✓		✓	✓		

The common denominator among these definitions and frameworks is that they define CT as a set of fundamental CS components. In this thesis, these computational-related components are referred to as CT skills. As illustrated in Table 2.1, despite disagreements among researchers, certain CT skills, namely abstraction, algorithms, decomposition, parallelisation, debugging, and data, are universally recognised in these various definitions and frameworks. At the core of these definitions and frameworks, CT is viewed as a thought process that uses these abilities to solve problems. Thus, this thesis defines CT as a thought process for solving problems through applying fundamental computing skills (CT skills), as discussed below, regardless of the application discipline.

2.4 Computational Thinking Core Concepts

The following concepts are based on a study of developed frameworks and documents from several organisations that have provided useful CT descriptions and categories for educators (Angeli, Voogt et al., 2016; Computing at School, 2014; Barr & Stephenson, 2011; Brennan & Resnick, 2012; Curzon, Dorling et al., 2014; Google for Education, 2016; ISTE, 2021). Decomposition, abstraction, algorithms, data collection, pattern recognition, and evaluation are all common CT components among researchers.

2.4.1 Algorithms

Algorithms are at the heart of both CT and CS because, in CS, the solutions to problems are algorithms, not just answers. An algorithm provides critical knowledge by providing a step-by-step solution to a problem and removing minor details to find a single solution that can be applied to similar problems (Humphreys et al., 2015). Algorithmic thinking differs from coding (i.e., the technical skills used to use a programming language) because it is a problem-solving skill related to devising a step-by-step solution to a problem (Selby, 2014). Algorithmic notions of sequencing (i.e., designing an algorithm, which entails placing acts in the correct order) and algorithmic notions of the flow of control (i.e., the order in which individual instructions or measures in an algorithm are evaluated) are both essential aspects of CT (Selby, 2014).

2.4.2 Abstraction

Abstraction is tied in with improving things to enable us to oversee complexity. It necessitates understanding the most important aspects of a problem and obfuscating the other explicit nuances on which we need not focus. The important features can create a model, or a disentangled representation of the first issue we were dealing with (Humphreys et al., 2015). Individuals gather important data (and discard unnecessary information) from complex structures to build designs and discover commonalities among diverse representations, which is the main component underpinning CT (Wing, 2006). The skill of abstraction is the ability to choose the best information to hide so the problem becomes simpler without sacrificing something valuable. Different representations make various tasks easy (Humphreys et al., 2015).

2.4.3 Decomposition

Decomposition entails breaking complicated problems into smaller, more manageable components and then focusing on each smaller issue individually (National Research Council, 2010; Wing, 2011). We can simplify a complex problem by breaking it down into smaller pieces that are easy to comprehend (Humphreys et al., 2015).

2.4.4 Pattern Recognition

Spotting patterns is a significant piece of the CT process; when we consider problems, we may perceive likenesses among them, and that they can be tackled in comparative ways. This is called pattern matching, and it is something we normally do in our everyday life (Humphreys et al., 2015).

2.4.5 Debugging and Evaluation

Debugging is detecting and correcting errors when behaviours do not adhere to instructions (Selby, 2014). Errors in logic can be thought of as parts of a tale where the plot does not make sense, and errors in coding and syntax can be thought of as bad spelling, punctuation, and grammar. When debugging, it is recommended that you adopt a straightforward four-step process based on rational reasoning: anticipate what might happen, discover what happens, figure out where something went wrong, and repair it (Humphreys et al., 2015).

Evaluation is about distinguishing the potential solutions to a problem and deciding, which is the best to utilise if they will work in certain circumstances but not others and how they can be improved. When deciding on our solutions, we must consider a scope of variables (Computing at School, 2014).

2.5 Computational Thinking Practices

2.5.1 Tinkering

This method entails doing something new to learn what it does and how it works. It strongly connects to logical reasoning. Being comfortable with tinkering allows us to see change as an opportunity rather than a threat, allowing us to keep our skills current and take advantage of new technology. It also helps to build perseverance (Humphreys et al., 2015). Tinkering experiences provide a framework for experimenting with gradual changes without the added cognitive challenge of actually constructing the product. The emphasis is on application, simulation, and problem-solving. These encounters eventually explore what-if scenarios (for example, What if I change this part of the code? What if I demolish this portion of the building? What if I add this to the object?) Learning moments that provide insight and sometimes lead to more questions (Sneider et al., 2014). Resnick et al. (2009) were among the first to incorporate the art of tinkering and remixing into CT.

2.5.2 Creating

Creating entails thinking about new ways to build and make valuable products (Humphreys et al., 2015). Some projects include a variety of media, each of which allows for artistic expression. We develop programmes that solve challenges or take advantage of opportunities. We brainstorm what we want to make or solve, analyse the problem, design, write, debug the code, and evaluate our work. Making things is also a powerful means of learning (Humphreys et al., 2015). Electronic prototyping kits like the LilyPad Arduino, a fabric-based construction kit that allows novices to design and build their soft wearables and other textile artefacts, is one effort to introduce CT skills (Buechley et al., 2008). Video games like Robo-Builder, a blocks-based game designed to introduce students to core computing aspects, use a constructivist style to challenge players to come up with and execute strategies to manipulate an on-screen robot using a custom visual programming language (Weintrop & Wilensky, 2013). However, this method, according to Lu and Fletcher (2009), does not help dispel the myth that CT is "programming."

2.5.3 Persevering

We must persevere, practice, train, and rehearse to gain competence in something complex. This is true in various fields, including art, music, dance, sport, chess, science, computer gaming, and programming. Systems and problems in computing and elsewhere may be complex, with unfamiliar contexts (Weintrop et al. 2016). We may need to explore various options or use unfamiliar technologies; we may even need to move from our natural, instinctive way of thinking to something more deliberate and logical. Perseverance entails being determined, resilient, and tenacious – never giving up and persevering in the face of adversity (Weintrop et al., 2016).

2.5.4 Collaborating

Collaboration entails working with others to produce the best performance. To improve good practice, teachers collaborate and observe one another (Hoic-Bozic et al., 2019; Zhang, 2020). Collaboration encourages us to persevere with assignments that would otherwise be too perplexing or daunting to complete. Computer scientists and software engineers often use or build on the work and coding of others, which is greatly encouraged by open-source software. Problems and processes are broken down into individual tasks. Different teams with various specialities collaborate on software creation. For example, computer games specialisations include programming, game design, painting, and animation (Barr et al., 2011; Computing at School, 2014).

2.6 The teaching of CT

The ISTE and the CSTA in the United States of America collaborated to develop a practical definition of CT that provides an overview and simplifies terminology for compulsory education teachers (CSTA & ISTE, 2011). They define computational thought as a problem-solving method that entails articulating challenges, applying logical order to data, presenting data through abstraction, analysing possible solutions for efficacy, and identifying correlations that will allow these solutions to be applied to various problems (Dong et al., 2019). ISTE highlights the objective of assisting all students to become computational thinkers who understand and control computing to solve issues in novel ways (ISTE, 2021).

Since CT is a problem-solving technique based on principles from CS, several experiments have been conducted using programming environments to introduce students to CT. For instance, Scratch, a common visual programming language, has been used to introduce children to programming (Resnick et al., 2009). Several studies focused on incorporating CT into content subjects, such as Social Studies, Music, Mathematics, Science, and Geography (Bråting & Kilhamn, 2020; Kallia et al., 2021; Montiel & Gomez-Zermeño, 2021; Sengupta et al., 2013; Settle et al., 2013; Weintrop et al., 2016). CT integration into primary education requires preparatory steps. Implementing CT at all compulsory education levels has resulted in a massive increase in demand for in-service teacher training (Bocconi et al., 2016; Mason & Rich, 2019).

About the fact that the majority of CT literature has focused on defining CT and identifying core principles, as reported by Grover and Pea (2013), a recent trend has arisen that examines how to teach these CT concepts to pre-service teachers (Alqahtani et al., 2021; Bower et al., 2017; Sands et al., 2018; Yadav et al., 2016). As Margolis (2010), Yadav et al. (2017), and Yadav et al. (2016) demonstrated, teachers cannot provide high-quality CT instruction to their students until they understand the concepts and are confident about incorporating them into their subjects. If teachers lack confidence in their ability to teach CT, this can result in negative interactions with learners when studying the concepts (Israel et al., 2015).

2.7 Existing Curricula for CT at the primary phase

In England, the Department of Education unveiled the national curriculum for CS in 2014. The national curriculum for computing aims to ensure that all students can comprehend and apply the fundamental principles and concepts of CS, including abstraction, logic, algorithms, and data representation, can analyse problems in computational terms, and have repeated practical experience writing computer programs to solve such problems, and can analyse problems in computational terms. The curriculum is separated into two levels. At the first stage point, students should comprehend what algorithms are, how they are implemented, and that programmes operate by explicit and unambiguous instructions. Second, students should be able to develop and debug basic programmes and forecast the behaviour of simple programmes using logical reasoning. At stage 2, students should be able to solve issues by decomposing them into smaller sections, use sequence, selection, and repetition in programmes, and deal with variables and other types of input and output (Curzon, Dorling, et al., 2014).

The Curriculum and Assessment Policy Statement (CAPS) has been revised by the Basic Education Department of South Africa to include coding and robotics in South African schools. The courses in coding and robotics are designed to guide and prepare students to solve issues, think critically, collaborate and be creative, and operate in a digital and information-driven environment. In the foundation phase (Grades R-3), the topic has been organised into Pattern Recognition, Algorithms, and Coding. This includes physical coding exercises that advance from Grade 1 to easy-to-learn, engaging digital platforms. In the intermediate phase (Grades 4-6), microcontrollers will be programmed utilising a block-based coding environment (Department of Basic Education of South Africa, 2021)

The Massachusetts Digital Literacy and Computer Science Curriculum incorporate digital literacy and CS from kindergarten through Grade 6; it is another current CT primary curriculum. Their goal is to engage students in digital literacy and CS skills and ideas by integrating activities and drawing links between what they already know and the environment in which they live. Their CT curriculum consists of abstraction, algorithms, programming, data, and development, e.g., modelling and simulation (Massachusetts Department of Elementary and Secondary Education, 2016).

2.8 The rationale for computational thinking in primary education

The majority of research on the development of CT has been conducted at the secondary and tertiary levels, with a few studies conducted at the primary level (see, for example, Butler & Leahy, 2021; Cateté et al., 2018; Hickmott, 2020; Linde-koomen, 2019; Rich, Yadav, & Larimore, 2020; Seiter & Foreman, 2013). Some scholars argue that the abstraction skills needed for computational reasoning are simply not developed enough in young children (Csizmadia et al., 2019; Rijke et al., 2018; Statter & Armoni, 2016). These arguments are often based on Piaget's early work, which maintains that "as children enter the formal organisational stage of growth, they acquire the potential for abstraction thinking" (Piaget, 1955, p.77). Piaget (2001) later rebutted his argument.

Sysło and Kwiatkowska (2013) concluded that introducing examples and specific artefacts could be more useful than describing CT principles in-depth. If learners are exposed to CT skills early in compulsory schooling, they can probably understand the concepts by the time they reach the job market, as developing a way of thought requires time (Yadav et al., 2014). Additionally, researchers suggest that since CT is "cross-disciplinary" in nature, it makes sense to implement it in primary school, where all subjects are logically integrated for learners into a single environment (Repenning et al., 2017). Previous research indicates that even preschool-aged children can grasp fundamental CT concepts (Bers et al., 2014; Sullivan & Bers, 2016).

2.9 Preparing teachers to teach computational thinking

To further improve CT teaching and learning, teachers must be trained in comprehensive ways on how to design CT learning exercises, how to teach CT, how to measure CT, and how to use technology to teach CT concepts (Boulden et al., 2021). According to the literature, teachers have mostly been exposed to CT skills through professional learning interventions. Thus, in-service teacher professional development plans must be incorporated, while teacher educators must find opportunities to incorporate CT instruction into their pre-service classes to best prepare pre-service teachers (Angeli & Giannakos, 2020).

Bean et al. (2015) conducted a professional development workshop for pre-service teachers on how to incorporate programming and CT into other subject areas, and the findings showed that more teachers plan to incorporate Scratch into upcoming classes. Rich and Yadav (2019) provided a familiar unplugged framework for teachers by incorporating CT concepts through current Mathematics and science practices. Rich and Yadav (2019)'s initial findings recommended three methods for primary school teachers to incorporate CT into their classrooms and improve instruction: using CT to direct teacher preparation, structuring lessons with CT, and introducing CT ideas as general problem-solving strategies. Duncan et al. (2017) commented on the preliminary findings of an ongoing trial in New Zealand investigating the teaching of new primary school topics using CT. The pilot results indicate that 'typical' teachers can engage students with the curriculum. Yadav, Stephenson, et al. (2017) provided guidelines for teacher educators on integrating CT into teacher education systems through instruction, key concepts, methods classes, teamwork, and teacher education.

Other researchers, Yadav et al. (2018), developed a professional development activity for elementary teachers to incorporate CT into science and investigated how their knowledge of CT developed during professional development. The findings indicated that teachers' thoughts were centred on what CT is, what it entails, and how it could benefit other realms. Kaya et al. (2019) incorporated CT into elementary science classrooms to increase pre-service teachers’ engagement, confidence, and self-efficacy in CT teaching through exposure to CT during an undergraduate-level science teaching methods course. According to Araujo et al. (2019), although elementary teachers recognise the value of CT, well-organised teacher training in elementary science teaching methods, courses are lacking. According to Muñoz del Castillo et al. (2019), pre-service teachers agree that CT skills are critical and should be integrated into their teaching.

Israel et al. (2015) examined how elementary school teachers with no previous programming experience integrated CT into their instruction and discovered several key concepts, including varying implementation models focused on teaching contexts to be used, continuous professional growth to be enhanced, and enhanced teacher interest in CS education. By completing the professional development programme, teachers' faith in their teaching ability has increased dramatically (Israel et al., 2015).

2.9.1 Improving elementary teachers’ CT content knowledge

Numerous studies exist that propose ways to educate teachers on CT. The ISTE published the Computational Thinking Competencies Standards for Educators in 2018, noting that teachers should recognise CT as a fundamental and cross-curricular ability and possess content awareness of its core components (Kaya et al., 2019).

Although extensive research has been conducted on teacher career development and technology adoption in the classroom, little research has been conducted on training primary school teachers to teach CT skills (Marksbury, 2017). A robotic intervention with scaffolded programming scripts was utilised in European research to enhance pre-service teachers’ algorithmic thinking and debugging abilities (Angeli, 2022). According to the findings of research conducted to develop pre-service teachers' CT skills using a programme design that included content that teachers could utilise in their everyday lives, the CT knowledge of the teachers increased after the intervention (Uzumcu & Bay, 2020). According to Mason and Rich (2019), the research on pre-service computing teacher education is in its infancy, with the majority of studies conducted in 2017 and 2018. Most of these experiments have aimed to improve teachers' subject knowledge and attitudes toward CT (Mason & Rich, 2019). These experiments ranged in length from two 50-minute sessions to an entire semester.

Despite these distinctions, pre-service CT teacher training has shown largely positive effects, especially in increasing teachers' knowledge of CT (Mason & Rich, 2019).

So far, few studies on in-service elementary teacher preparation have shown promising outcomes. Every study demonstrated an improvement in subject comprehension, teacher attitudes, or beliefs. The Code, Connect and Create (3C) professional development (PD) methodology was created by Jocius et al. (2020) to assist middle and high school content area teachers in incorporating CT into their classes (discussed in more detail in section 2.13.3). Teachers' understandings of the importance of CT in subject area classrooms and their self-efficacy and views about CT integration into disciplinary material were said to have shifted due to the 3C professional development approach (Jocius et al., 2020). Numerous researches emphasised improving content knowledge, but few focused on developing teachers' pedagogical content knowledge. These studies have been squarely focused on teacher development, with no reports about how students have improved or how improvements in student knowledge have influenced teacher growth in subject knowledge, attitudes, or opinions about CT (Mason & Rich, 2019). Mason and Rich (2019) summarised the literature review findings on preparing K-6 teachers to teach CT, indicating that teachers value more realistic and hands-on opportunities for developing their subject skills. Curzon, McOwan et al. (2014), Falkner et al. (2018) and Bower et al. (2017) provided detailed workshops for teaching teachers' CT concepts and demonstrated that they develop teachers' confidence, fill teachers' knowledge gaps about CT, and can be a realistic and effective method of teaching computing to primary school learners. A study that assessed the CT skills of primary and secondary school teachers who attended basic robotic coding in-service training found that those who participated had boosted their CT knowledge (Çakır et al., 2021).

2.9.2 Teaching strategies for Integrating CT into teacher's professional development

Numerous attempts have been made to create mechanisms for CT incorporation into teachers' PD. Sengupta et al. (2018) suggested pedagogical guidelines for incorporating CT into K-12 STEM, emphasising the importance of re-conceptualising CT as discursive, perspective-taking, and embodied experiences. Another model incorporates three stages of CT incorporation: (a) defining CT principles and practices already presents in the curriculum, (b) strengthening the links between disciplinary and CT concepts through additional lessons, and (c) broadening disciplinary knowledge through the integration of CT-related activities (Waterman et al., 2019). Existing research indicates that computational experience will help students improve their comprehension of subject content, their CT skills, and their knowledge of the importance of computation through STEM fields (Lee et al., 2020). This literature suggests a continuum of integration methods while noting the difficulty of achieving complete integration within mathematics (Israel & Lash, 2020). Along with these pedagogical frameworks, CT integration is motivated by pragmatic concerns. For instance, integration is often used to address the challenge of fitting CT into an often hectic school day and a lack of resources for stand-alone classes (Boulden et al., 2018; Nordby et al., 2022; Yadav et al., 2017). Specific strategies are considered in integrating CT into teachers' PD in the following.

Unplugged strategy

This strategy applies to CT problems that do not include using computing devices. Ouyang et al. (2018a) and Rich, Yadav and Larimore (2020) assisted teachers in presenting CT ideas and improving students' CT abilities by non-programming/unplugged science and mathematics exercises. Curzon, McOwan et al. (2014) proposed detailed workshops for teaching primary school teachers CT principles through unplugged practise, demonstrating that the approach develops teachers' confidence and closes skills gaps in CT. Grgurina et al. (2014) demonstrated the versatility of unplugged practices and digital storytelling.

Brackmann et al. (2017) investigated the effectiveness of unplugged activities by carrying out a quasi-experiment in two primary schools in Spain. The findings showed that the students who participated in the experiment significantly enhanced their CT skills more than those in the control group. This proves that the unplugged approach may be suitable and effective in developing CT skills (Brackmann et al., 2017).

Below is the activity done by learners using abstraction, decomposition and algorithm skills in a simpler remake of the Code Master board game developed by the ThinkFun company. "In this activity the student is supposed to find a route between two nodes using the allowed colours for each path. All the colours had to be used, leaving no blank spaces. The number located on the left side is the start point and on the right side the finish point" (Brackmann et al., 2017, p.5).

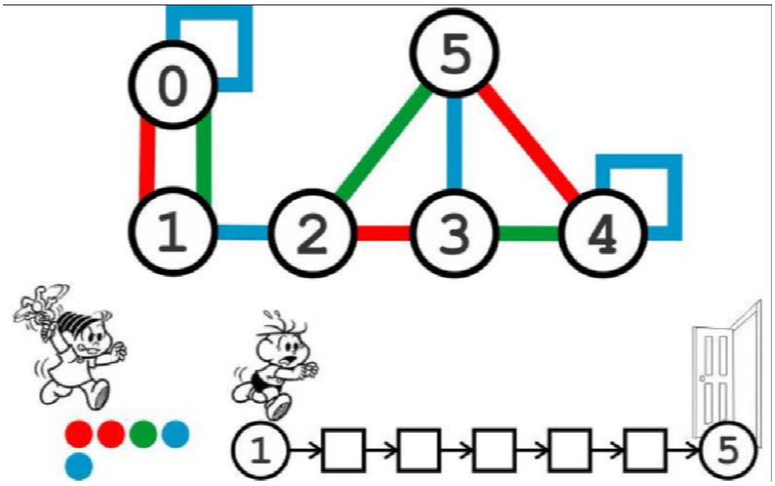


Figure 2.1 "Monica's Automata": Unplugged activity - Source: Brackmann et al., (2017)

Some past studies have investigated the unplugged strategy, which could allow for the development of CT skills in non-computer environments (Brackmann et al., 2017; Curzon, McOwan, et al., 2014; Faber et al., 2017; Jagušt et al., 2018; Looi et al., 2018; Mensan et al., 2020; Olmo et al., 2020). Given Wing's (2006) assertion that CT is a fundamental ability that everybody can possess, this approach enables the reach of many teachers and students who lack access to computing devices, especially in developing countries. Unplugged practices expose teachers to CT principles without distracting them with technology and gadgets and enable teachers to incorporate these concepts in various realms independent of their socio-economic status (Ausiku & Matthee, 2021). Although the majority of the literature concluded that the easiest fields to incorporate CT are mathematics and physics, some teachers see opportunities to integrate CT into subject areas, such as the arts and social studies (Gadanidis, et al., 2017; Lamprou & Repenning, 2018; Muñoz del Castillo et al., 2019).

According to Huang and Looi (2020), there is a need for a more fundamental analysis that hypothesises and tests models or processes explaining how unplugged methods evolve CT – including logical and procedural understandings, and problem-solving abilities. Though unplugged practices do not have the immediate gratification associated with product development, they are suitable for encouraging deeper contemplation on large concerns about emerging technology by repurposing time otherwise spent studying the technological specifics of a programming language. Rather than limiting students' understandings of CS to those developed by computer scientists, one should broaden the discipline's reach across various perspectives (Huang & Looi, 2020).

The Computer Science Education Research Group at the University of Canterbury has an initiative called CS Unplugged (https://www.csunplugged.org/en/) that supports unplugged activities and offers ready-made teaching tools. In order to effectively explain what computer science entails to junior high schools with limited access to technology, the CS Unplugged initiative was started in the 1990s (Bell & Vahrenhold, 2018). In order to simulate the behaviour of binary digits, for instance, data representation is investigated using playing cards and a magic trick (Bell & Henderson, 2022). Exercises for CS Unplugged were developed as practical, physical activities that might be utilised in a classroom without computers. Instead of starting with programming, which can cause students to become bogged down in dealing with low level details, they provide them the chance to engage with computer science concepts at a high level (Bell et al., 2016). The "CS Unplugged at home" material (https://www.csunplugged.org/en/at-home/) was made accessible at the time when children were frequently obliged to stay at home due to the COVID-19 pandemic and was created for use in cases when a teacher would not be present. Step-by-step directions, including images and what to say, were supplied since it was intended to be used in a situation where the primary support would be an adult without any prior knowledge in computer science or education (Bell & Henderson, 2022).

Programming

According to the literature, an important method of educating teachers about CT principles is to teach them to programme using basic programming languages and hands-on programming activities. Kong et al. (2020) developed a professional learning curriculum for in-service teachers focusing on the connection between CT and programming. The findings of CT concepts and activities assessments administered during a teacher development programme for CT about programming showed that students made steady progress during courses 1 and 2 (Kong et al., 2020). During an extended duration of course 2, the attending teachers were given the opportunity to teach programming to incorporate CT, and many shared a desire to use Scratch in their future teaching, recognising how Scratch would enrich classes (Kong et al., 2020).

Geldreich et al. (2018) offered Teacher Development Courses (TDCs) for teachers to learn teaching CT skills by programming, and the findings indicated that pre-service teachers who completed the TDCs developed their CT practices continuously. It demonstrated how helpful and advantageous it is to expose teachers to programming activities when they are still in university. Geldreich et al. (2018) discovered that teachers are more receptive to instruction where it is directly linked to their current teaching and provide opportunities for collaboration with peers in a supportive setting.

Gleasman and Kim (2020) investigated block-based programming as a possible technique for conceptually teaching primary mathematics based on pre-service teachers' linkages between CT and Mathematics. Following the intervention, the descriptive analysis demonstrated changes in pre-service teachers' opinions toward the proposed teaching strategy (Gleasman & Kim, 2020).

Cetin (2016) researched how Scratch-based training affected pre-service teachers' understanding of basic programming principles and attitudes toward programming. According to the study's findings, the teachers who participated in the training showed much greater comprehension of basic computing concepts. According to qualitative data, Scratch-based education was shown to be effective in creating a more relevant learning environment for pre-service teachers (Cetin, 2016).

These findings by Gleasman and Kim (2020), Cetin (2016) and Kong et al. (2020) demonstrated that when teachers are taught to use Scratch (block-based programming) in CT, they demonstrate a constructive reaction, including stronger comprehension and respect for the programme's usefulness. Linde-koomen (2019) and Bean et al. (2015) conducted an intervention module about using computing and computer technology as instructional techniques in other topics. According to the post-survey, students plan to incorporate Scratch into upcoming classes. Falkner et al. (2018) and Bower et al. (2017) enhanced in-service teachers' CT skills through seminars that incorporated programming as a teaching technique and Hopscotch, Blockly, Scratch, and Beebots as software tools. Following the interventions, teachers gained a thorough knowledge of CT and its sub-components and developed various techniques for teaching CT in their classrooms.

Marcelino et al. (2018) and Haduong and Brennan (2019) demonstrated that Scratch could be successfully taught and learnt through distance education as teachers produced high-quality instructional materials for their classrooms.

While CT is more than CS, these results relating to CS show that interventions involving active teacher involvement will boost teachers' computing self-efficacy, attitudes, and knowledge regarding CT. This is supported by Mason and Rich's (2019) comprehensive study of literature on pre-service and in-service initiatives that enhance K–6 teachers' attitudes about teaching computing, coding, or CT.

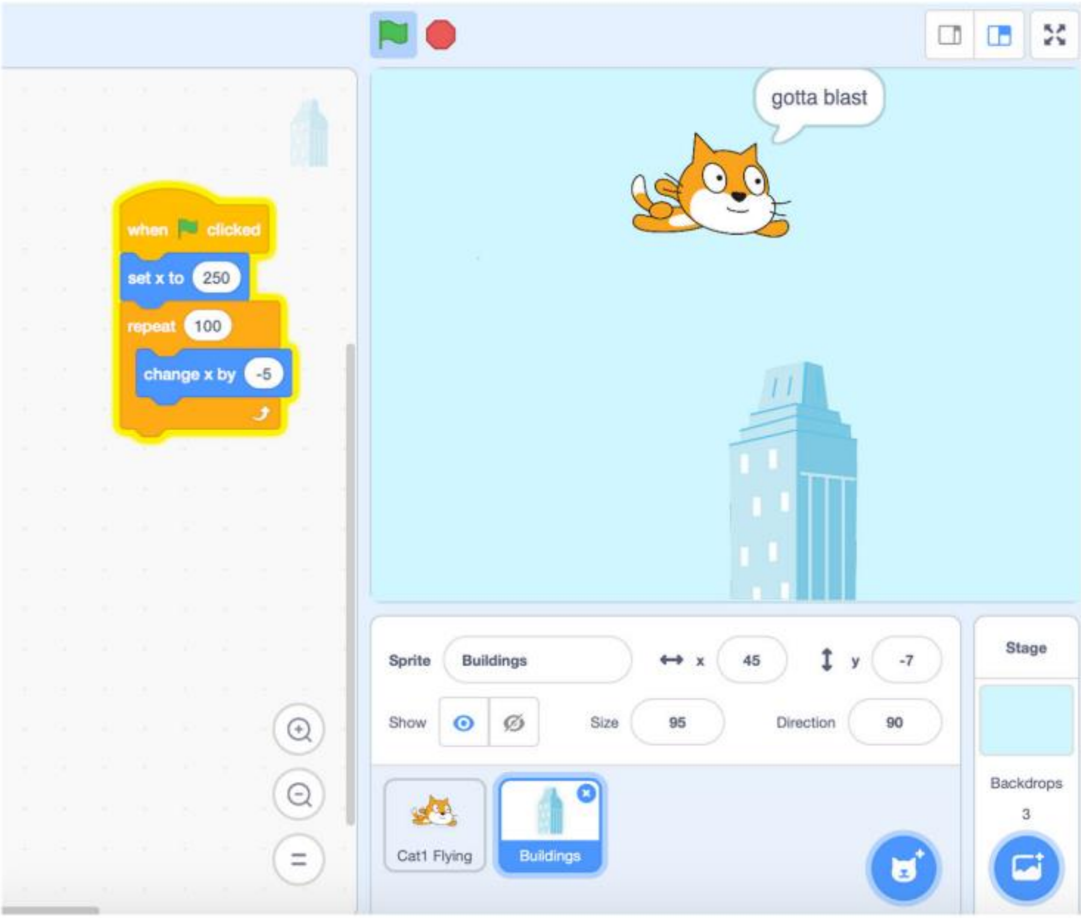


Figure 2.2 Segment of learner's activity in Scratch - Source: Gleasman and Kim (2020)

Game-based learning

Leonard et al. (2017) and Leonard et al. (2018) discovered that game design allowed aspirant teachers to extend common STEM ideas to various dynamic activities to generate models and representations using LEGO. Nickerson et al (2015) outlined a method for teaching various CT concepts and examined how teachers who use the Scalable Game Design curriculum incorporate these contexts into their instruction. AgentSheets/AgentCubes, which are 2-D and 3-D programming tools, were used, and it was shown that CT could be taught successfully when scaffolding for basic skill acquisition is offered. Game creation enables teachers to communicate with one another and interact with computer scientists to better understand CS principles (Nickerson et al., 2015). Research to see how science teachers new to computing and trying to integrate CT into their lessons helped students solve computational problems. Science teachers in three US school districts blended computer game design into teaching climate systems and global warming in the project from which this research was derived (Tucker-Raymond et al., 2021).

Robotics

According to the literature, robotics is the second most researched method for teaching CT skills to primary school teachers (Ausiku & Matthee, 2021). Chalmers (2018) investigated how primary school teachers in Australia integrated robots and coding into their classrooms using LEGO robot kits and discovered that teachers increased their confidence and skills. Jaipal-jamani and Angeli (2017) and Esteve-Mon et al. (2019) investigated pre-service teachers' understanding of science concepts, self-efficacy, and CT as they engaged in robotics activities in a science methods course using LEGO and MakeyMakey as tools. The findings indicate that robotics activities sparked interest in robotics and increased CT skills. According to research, pre-service teachers' interactions with robots may affect their intent to utilise them in the classroom (Alqahtani et al., 2021). Two groups of participants used robots to solve mathematical problems, and one group had the opportunity to plan and implement activities involving robots and first-graders. The findings show that allowing pre-service teachers to study, contemplate, and experience robotic technology can help them integrate them into their future teaching (Alqahtani et al., 2021).

Project-based learning

Project-based learning is a form of instruction in which students' learning is organised around tasks. Projects are lengthy assignments focused on difficult questions or challenges that require students to engage in design, problem-solving, decision-making, or investigation activities; PBL allows students to work comparatively independently for long periods and culminates in practical items or presentations (Jones et al., 1997). A study of teachers' engagement with CT ideas and co-construction of meaning in their project-based activity offered a precise scaffolded pattern of incorporating CT principles into a project-based math methods course (Mardi, 2020).

In a pilot study, Du and Igwe (2018) developed a project-based technology-based workshop for teachers who introduce CT principles and assess their aptitudes and ideas for incorporating these learning tools into their curricula. The findings imply that technologically driven PBL design principles can boost participants' self-esteem and inspire them to think about implementing CT related pedagogical practices in their classrooms, particularly in subjects related to maths and science.

Ozturk et al. (2018) investigated the use of Project-Based Learning (PBL) to implement CT and discovered that teachers collaborated with grade level team members during joint project preparation sessions to apply standards. According to a meta-analysis conducted by Hsu et al. (2018), most CT teaching (in general) incorporates PBL, problem-based learning, and cooperative learning into its CT operations. This approach allows teachers to examine real-world problems through the lens of CT principles. A study by Hsieh et al. (2022) looked into how to include physical robots in PBL courses for thinking skills training to increase the learning performance of the CT capacity. The results reveal that the PBL method combined with the teaching material of the robotic visual programmes approach was considerably more successful in boosting students' learning successes than the traditional teaching method approach (Hsieh et al., 2022).

2.10 Perceptions and attitudes of teachers towards CT

To ensure the fruitful incorporation of CT into primary education, it is necessary to understand the teachers' expectations and attitudes prior to implementation. Most research has incorporated CT into mathematics and science teaching to explore teachers' experiences incorporating CT or practice activities (Gleasman & Kim, 2020; Rich, Yadav, & Schwarz, 2019). It was found that teachers make correlations between their maths and science teaching and their pre-existing understanding of CT-related terms (Nordby et al., 2022; Rich, Yadav & Schwarz 2019). Teachers perceive CT as a problem-solving technique, primarily equating it to algorithmic reasoning (Fessakis & Prantsoudi, 2019; Rich, Yadav & Schwarz, 2019). Fessakis and Prantsoudi (2019) discovered that pre-service teachers had a favourable outlook toward introducing CT into the curriculum. The teachers are not on the same level in terms of prior education and have shown an interest in attending further appropriate training.

Attitudes about programming and CT are related to the association teachers create with prior experience. Gleasman and Kim (2020) discovered that participants interpreted the proposed teaching technique of using block-based programming tools and CT to teach mathematics in primary schools as both useful and realistic. Cetin (2016) examined pre-service teachers' attitudes toward programming using Scratch-based teaching and discovered that teachers gained a better grasp of coding principles and a more optimistic outlook. Meanwhile, Tankiz and Atman Uslu (2022) discovered that pre-service teachers' CT and self-efficacy views of teaching improved dramatically after taking a course in which instructional games were created in a block-based programming environment. This increase had a minor impact on CT skills but a significant impact on self-efficacy impressions of the skill's teaching.

According to Ketelhut et al. (2019), teachers' views of CT reach across several dimensions, including the personal domain, where they express their knowledge, beliefs, and attitudes toward CT integration; the domain of practice, where they prepare their lessons that included CT skills; and the domain of consequence, where they could explain the outcomes of the lessons. For instance, pre-service teachers believe they successfully learnt how to programme but are unsure about their acquired CT skills.

In contrast to Ketelhut et al. (2019), Lamprou and Repenning (2018)'s results indicate that pre-service teachers have a strong understanding of how to relate CT principles to programming but are not generally able to code. Yadav et al. (2011), Yadav et al. (2014) and Linde-koomen (2019) conducted an instruction module with pre-service teachers and found that students recognised that CT is more than the use of electronic devices and technologies. They have grasped integrating CT to promote problem-solving and critical thought. About half of the students suggested including programming as a lesson objective in upcoming classes (Linde-koomen, 2019). Zha et al. (2020) examined the effects of a flipped classroom module on pre-service teachers' knowledge, self-efficacy, and attitudes toward CT. While there was initial fear and uncertainty, the findings showed that as they practised with step-by-step manuals, they developed an interest in the coding tasks and strengthened their skills. Teachers mostly have a favourable opinion of the CT learning experience.

However, the existing reliance on programming as a teaching method may discourage teachers without computing experience or from schools that lack access to computers and programming platforms from applying CT skills in other subject realms (Ausiku & Matthee, 2021).

2.11 Challenges faced by teachers when integrating CT into their classrooms

2.11.1 Pedagogical and Content Knowledge

Teachers face various obstacles when they attempt to incorporate CT into their classes. According to Hsu et al. (2018) literature review, teaching staff should pursue a comprehensive education in CT. Sentance and Csizmadia (2017) surveyed over 300 teachers currently teaching CT and discovered that teachers' most frequently reported difficulty is a lack of own computing subject knowledge. According to some reports, some primary school teachers also lack a firm grasp of the CT concepts (Bower et al., 2017; Corradini et al., 2017; Kaya et al., 2019; Kong & Wong, 2017; Voogt et al., 2015). Non-cognitive considerations, such as fear and motivation, always affect how teachers teach and cannot be overlooked (Bower et al., 2017; Corradini et al., 2017; Kaya et al., 2019; Kong & Wong, 2017). Additionally, teachers expressed reservations about incorporating CT into their classes due to insufficient teaching time and concerns about approaching high-level CT in developmentally appropriate ways (Rich, Spaepen et al., 2019).

Although curriculum knowledge (CK) alone cannot devise successful instruction, the academic content of a subject serves as the bedrock for teaching and learning. Academic content must underpin pedagogical advances (Littlejohn & Stefani, 1999;

Zhao et al., 2001). Effective classroom approaches require designers to ensure that the material is adequately entertaining to pique children's interest in the world of learning (Brown, 1992, p. 173). Teachers cannot be expected to plan an informative and entertaining CT-infused lesson if they are unfamiliar with the subject's academic material. Insufficient knowledge of their subject matter may also have serious consequences; for example, a teacher could be unable to assess the complexity of the material, preventing them from facilitating learning development. Additionally, researching what teachers know will assist them in planning their career advancement more purposefully, and therefore, more attention can be paid to investigating their CK (Zhang, 2020). Kong and Lao (2019) created an instrument to assess pre-service teachers' comprehension of CT, focusing on CT procedures through the lens of Brennan and Resnick's (2012) framework.

2.11.2 Technological difficulties

A major issue is school funding and access to computers and other technologies (Adler & Kim, 2018; Ozturk et al., 2018). The findings of a study by Ozturk et al. (2018) on how elementary teachers with limited PBL skills and programming expertise endured integrating CT into their lessons revealed that in addition to a lack of subject knowledge, teachers encountered resource and technology difficulties. Lloyd and Chandra’s (2020) research involved pre-service teachers from Australia conducting CT lessons in a rural Malaysian school, demonstrating the importance of background regarding how the intended programme is implemented in the real-world. Pre-service teachers must understand the importance of being adaptable and practical.

Many pre-service teachers cannot extend their skills and experience in classrooms due to a lack (or failure) of technical tools, as the proposed programme required the use of an online version of Scratch that was affected by a slow Internet connection. Also, in developed countries, such as Australia, access to technologies and technical personnel capable of troubleshooting technological issues on-site can be challenging (Lloyd & Chandra, 2020). Ketelhut et al. (2019) examined how teachers intended to incorporate CT into Science lessons before and during a professional learning curriculum. The study's authors addressed how some of the participating teachers thought they needed additional technical services before implementing CT and how some of these teachers could not obtain such resources due to a lack of funding.

Additionally, researchers discovered that K-12 teachers often face obstacles due to a shortage of adequate instructional tools when integrating coding and CT lessons (Bower et al., 2017; Kadirhan et al., 2018; Kidd & Morris, 2017; Rich, Browning et al., 2019;).

2.11.3 Lack of shared vision

Another issue frequently found in studies of introducing coding and CT in K-12 is a lack of teamwork among school staff (Bower et al., 2017; Rich et al., 2017; Sentance & Csizmadia, 2017; Sherwood et al., 2020). According to those reports, the issue is an insufficient collaboration among teaching personnel and insufficient collaboration among teaching staff, administrative staff, and school leaders. For instance, Rich et al. (2017) stated that teachers faced difficulties executing computing lessons due to a lack of contact between teachers and their principal about how the lessons could be executed, which the authors called a “Lack of a Shared Vision” (p. 459). Rich et al. (2017) established enablers and inhibitors that K-6 teachers at a single school in the United States experienced while integrating computing and engineering lessons in their research. They argue that the biggest impediment to implementing these programming and engineering lessons was a lack of shared vision. According to Rich et al. (2017), some teachers could not introduce computing and engineering lessons because their school leaders did not explicitly explain the intent of teaching these topics to them or because their colleagues did not collaborate with them on preparing computing and engineering lessons. These results emphasise the critical role of teamwork among school staff (including teachers, administrative staff, and school leaders) in implementing coding and CT curricula (Hickmott, 2020).

2.11.4 Inadequate time for learning and teaching

Another issue often found in surveys of teachers’ adoption of coding and CT in grades K-12 is a lack of time. According to some scholars, teachers face difficulties due to a lack of time for studying coding and CT and adapting instructional materials and lesson plans (Bower et al., 2017; Kadirhan et al., (2018); Ketelhut et al., 2019; Rich et al., 2017). Bower et al. (2017) described a lack of time as one barrier to teachers’ feeling positive about teaching coding and CT in their research.

Bower et al. (2017) facilitated a one-day PD for coding and CT workshops for in-service K-12 teachers, polled participants about their confidence in teaching coding and CT before and after the workshop and described common barriers to teachers feeling secure in teaching these skills. Before the workshop, several teachers who replied to the survey expressed a lack of confidence in their belief that they lacked coding and CT topic skills. Following the workshop, a lack of subject expertise was a less frequently cited cause in survey responses, and other factors, such as a lack of time, discouraged teachers from feeling positive about teaching coding and CT. Bower et al. (2017) do not clarify whether the lack of time relates to teaching or preparation time, although the article's included comments imply that the lack of time refers to a lack of time for coding and CT lesson training. Similarly, Kadirhan et al. (2018) addressed how several teachers who reacted to their survey struggled to find time to adapt current coding and CT instructional materials for classroom use.

According to some scholars, teachers have often felt constrained by a lack of instructional time, as it was difficult for teachers to incorporate coding and CT into the content they were already teaching (Bower et al., 2017; Israel et al., 2015; Ketelhut et al., 2019; Rich, Yadava & Schwarz, 2019). Coding and CT incorporate additional material into curricula that teachers frequently feel are overcrowded, according to Grover and Pea (2013), which is especially difficult for K-6 teachers who are still expected to teach most subjects (Falkner et al., 2015). The teachers in the Rich et al. (2017) study were not required to teach computers or engineering at the time of the study, and some expressed fear that teaching such subjects would divert classroom time from the “basics” (p. 457), such as science and mathematics. Israel et al. (2015) examined the introduction of coding lessons in a single elementary school in the United States and described “Limited Instructional Time” (p. 271) as a frequent implementation hurdle faced by teachers. The teachers in that study could incorporate coding into the various subjects they taught as a way to overcome the constraint of insufficient instructional time.

2.12 CT in developing country context

The literature indicates that the context of developing countries is under-researched (Ausiku & Matthee, 2021). Most studies have been conducted in the United States and

Europe, while few have been in Africa (Emembolu et al., 2020; Lin & Shaer, 2016; Ogegbo & Ramnarain, 2022; Talib et al., 2019).

A few studies focusing on developing countries were identified: Muñoz del Castillo et al. (2019) conducted a study in the context of developing countries and found that while teachers in Colombia accept that CT skills should be integrated into their teaching, there is still a long way to go before well-organised teacher training systems are implemented in developing countries. Another research planned, conducted and evaluated a 20-hour PD programme for 21 teachers from Antioquia, Colombia's public middle and high schools. According to preliminary findings, engaged instructors boosted their expertise and motivation in incorporating CT methods into their disciplinary learning contexts (Espinal et al., 2021). Cetin (2016) examined the impact of Scratch-based instruction on pre-service teachers' comprehension of fundamental programming principles and attitudes toward programming in Turkey and discovered, along with previous studies, that Scratch-based instruction was effective at creating a more meaningful learning experience for pre-service teachers. In mainland China, Hong Kong and Brunei Darussalam, Kong and Wong (2017) investigated primary school teachers' perceptions and perspectives on attending PD programmes that teach CT skills by coding. The findings of this study revealed that some primary school teachers still do not understand the concept of CT and that teaching coding is difficult for teachers because they lack both pedagogical subject knowledge and content knowledge (Kong & Wong, 2017).

A few papers were considered to lay the groundwork for the teaching and learning CT in Nigeria in the framework of science, technology, engineering, and mathematics (STEM). For instance, current research has concentrated on creating the framework for learning and teaching to increase teachers' ability to promote CT education (Emembolu et al., 2020; Talib et al., 2019). Emembolu et al. (2020) recruited educators in nine Nigerian States under the TeachAKid2Code initiative to give training and capacity building to expand the number of STEM educators in Nigeria. Insufficient STEM educators and limited access are barriers that the study discusses in relation to the need for teaching young people in STEM fields. The study adopted a MOOC format and direct teaching utilising online resources using a constructionist approach and connectivism learning theory to teach and train volunteers to deliver and facilitate

STEM outreach activities to young children in primary and secondary schools in Nigeria (Emembolu et al., 2020).

In a different context, Talib et al. (2019) studied graphic calculator (GC) technology to improve students' critical thinking and CT skills. This study shows that GC may be optimised as an educational tool to improve the CT skills of students. This project investigated the possibilities of leveraging GC technology to promote CT abilities. The study's data were acquired from secondary sources through a systematic review technique. The study revealed that GC seems a useful tool for implementing STEAM education and strengthening students' CT by analysing and decomposing real-world problems, engaging in the abstraction process, and coding an algorithm. The findings justify the influence of GC on the development of CT skills, maximising the educational advantages for pupils (Talib et al., 2019).

There are not many studies conducted in an African context, and additional studies should be conducted in developing countries, especially in an African context where teaching techniques and resources should differ (Ausiku & Matthee, 2021).

2.13 Professional Development of Teachers

As this study focuses on preparing primary school teachers to teach CT, this section of the literature review focuses on different approaches followed in the PD of teachers. The literature casts a broad net on what constitutes PD, which Little and Lieberman (1987) defines as "any activity that is intended partly or primarily to prepare paid staff members for improved performance in present or future roles in the school districts" (p. 491). Teachers learn in various contexts during their careers, including their classrooms, school environments, and PD activities or seminars (Borko, 2004).

Desimone (2009) suggested a conceptual framework for effective PD for teachers, discussed below.

2.13.1 Desimone's (2009) conceptual framework for effective professional development

To succeed, teacher PD should exhibit essential characteristics: subject emphasis, active learning, coherence, suitable duration, and collective involvement as shown in Figure 2.3 below (Desimone, 2009).

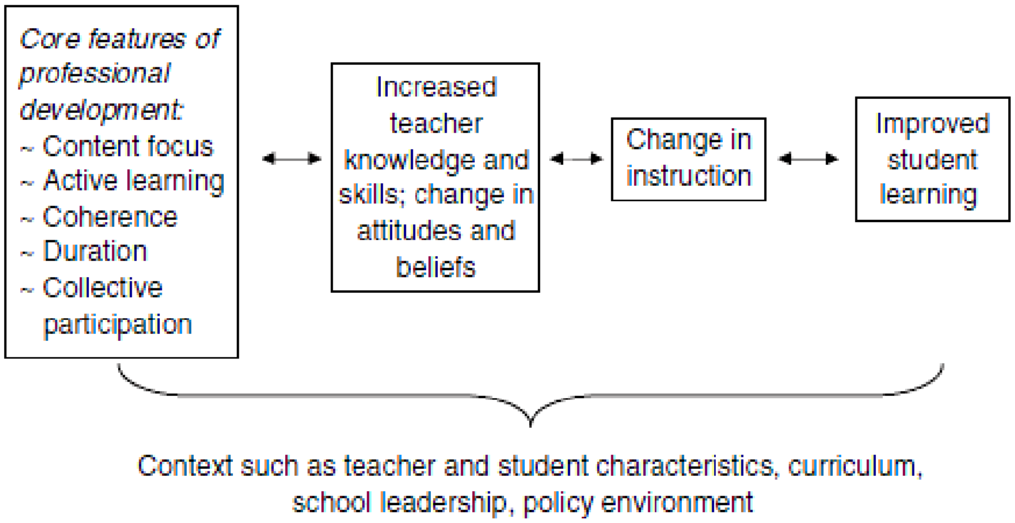


Figure 2.3 Core conceptual framework for studying the effects of professional development of teachers and students. (Source: Desimone, 2009)

Core features of teachers' professional development

Content focus

Over the last decade, a growing body of research has established a connection between programmes that concentrate on subject matter content and how students learn the content and increases in teacher knowledge and skills, practice progress, and, to a lesser degree, student achievement (Cohen (1990) in Desimone (2009)). PD with a strong curriculum emphasis has been shown to positively affect teacher performance compared to PD that lacks a strong subject matter focus (Desimone, 2009).

Active learning

Active learning opportunities for teachers are also associated with the effectiveness of PD (Garet et al., 2001; Louckes-Horsley et al., 1998). In contrast to passive learning, which is typically defined by listening to a lecture, active learning may take a variety of forms, including observing expert teachers or being observed by them, followed by interactive feedback and discussion; reviewing student work in the topic areas covered; and leading discussions (Darling-Hammond et al., 2017; Desimone, 2009).

Active learning is more effective than passive learning regarding teacher career development (Desimone, 2009).

Coherence

Coherence is the third critical aspect stressed in the literature; it refers to how much teacher instruction is associated with teachers' experience and values. Another critical feature of coherence is the alignment of the classroom, district, and state reforms and regulations with what is learnt in professional learning (Desimone, 2009). Desimone (2009) identified two critical components of coherence: first, instruction must align with teachers' experience and beliefs; and second, PD material must align with classroom, regional, and national policy.

Duration

According to research, intellectual and pedagogical reform requires PD programmes to be sufficiently lengthy, both in terms of the time period covered by the practice (e.g., one day or one semester) and the number of hours expended on the activity (Desimone, 2009). Since there is no definitive “tipping point” for length, research supports practices spaced over a semester (or intensive summer institutes with follow-up during the semester) and including at least 20 hours of interaction time (Desimone, 2009).

Collective participation

Mutual engagement is important, and this characteristic can be achieved by including teachers from the same school, grade, or department. Such structures facilitate dialogue and debate, which can be an effective method of teacher education (Desimone, 2009). PD is more successful when teachers from the same school, grade, or department collaborate (Desimone, 2009).

Changes in knowledge, skills, attitudes, and beliefs

Teachers’ knowledge and abilities are improved, and their attitudes and beliefs are altered due to effective PD. This link between teacher knowledge, practice and learners’ outcome is reflected in the literature (Desimone, 2009).

Changes in instruction and student learning

Desimone's (2009) path model's third aspect is "change in instruction," followed by "improved student learning" (p. 185). Desimone's framework enables one to test both a theory of instruction (e.g., whether PD changes teacher experience, values, or practice) and a theory of teaching (e.g., whether altered practice affects student achievement), which are both important for completing our understanding of how PD functions. Teachers apply their newly acquired information, skills, attitudes, and beliefs to improve their lessons' content, pedagogical style, or both. Increased student learning results from instructional improvements.

Context

Context, which underpins Desimone (2009)'s model's four components, "acts as an important mediator and moderator" (p.185). The context comprises "teacher and student characteristics, curriculum, school administration, and policy environment" (p. 185). Numerous scholars and academics have written about the importance of context and the need for teacher educators to balance PD with the environments in which teachers work (Desimone, 2009).

2.13.2 TPACK Framework

Another commonly used framework for assisting teachers in determining how to change their instructional methods to include instructional technology to address particular subject areas is technological pedagogical content knowledge (TPACK) (Koehler & Mishra, 2009). The TPACK framework relates to this study because of teaching approaches to developing CT skills that use technological tools, such as programming and robotics.

Mishra and Koehler (2006) developed TPACK as a model for teachers who build on Shulman's (1986) concept of pedagogical content knowledge (PCK) by incorporating technology expertise. TPACK is organised on the premise that the curriculum (what you teach) and teaching process (how you teach) must be the driving force behind whatever technology you want to use in your classroom to enhance learning (Mishra & Koehler, 2006).

Other scholars have explored related concepts, most frequently through the lens of a different labelling system. The definition of TPACK, as presented here, evolved over time and through several publications, with the most comprehensive explanations of the architecture appearing in Mishra and Koehler (2006) and Koehler and Mishra (2009). Shulman (1986) defined three broad areas of skills that good teachers can possess. The first, content knowledge (CK), refers to the teacher's knowledge of the subject being taught. The second, pedagogical knowledge (PK), refers to the techniques of instruction used by the teacher when instructing students. Another critical area is PCK, which refers to the approaches used to train basic concepts in a teacher's subject area (Shulman, 1986). Similarly, important to the model are the relations within these collections of bodies of how they are integrated and recombined within the TPACK model (Mishra & Koehler, 2006). Figure 2.4 below shows the TPACK framework.

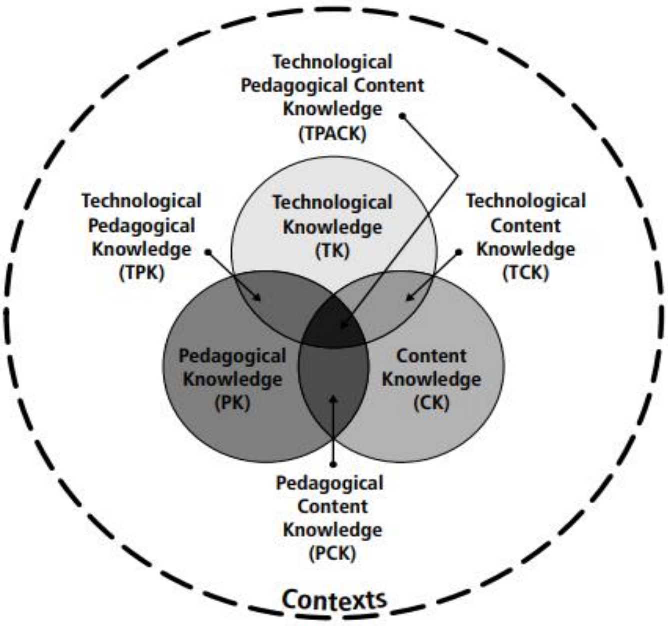


Figure 2.4 The TPACK Framework and its knowledge components (Source: Mishra and Koehler (2006))

TPACK Approaches to Teacher Development

Additionally, researchers and practitioners examine "where to begin" while designing strategies for developing TPACK in pre-service and in-service teachers. Numerous strategies for teachers' development of TPACK have been suggested. Two of these strategies ("PCK to TPACK" and "TPK to TPACK") use teachers' previous expertise and familiarity with one or more of the central knowledge bases. The third, "Developing

PCK and TPACK concurrently," takes a systemic approach to technical TPACK development by focusing on teachers" perspectives identifying, planning, and refining instructional artefacts to address specific learning difficulties. Table 2.2 below summarises three strategies for creating TPACK.

Table 2.2 Approaches for developing TPACK in pre-service and in-service teachers

Approaches	Description
From PCK to TPACK	Teachers draw upon their existing PCK to form insights into which technologies might work well for specific learning goals (Doering et al., 2009; Harris & Hofer, 2009).
From TPK to TPACK	In general, teachers build on their knowledge of technology to develop expertise in using technology in learning contexts; they then use that knowledge to identify and develop specific content that benefits from teaching with technology strategies (Angeli & Valanides, 2009).
Developing PCK and TPACK simultaneously	Teachers gain experience and knowledge through projects that require them to define, design and refine solutions for learning problems and scenarios. The design process serves as the locus for activities that produce insights into how technology, pedagogy, and content interact to create specialised forms of knowledge (Brush & Saye, 2009; Mishra & Koehler, 2006).

Application of TPACK to the professional development of CT teachers

A CT curriculum was implemented using a TPACK-based PD framework during a teachers' training intervention engaging 291 in-service senior primary teachers (Kong & Lai, 2021). The intervention employed a module-based design, and four modules were provided. Four 3-hour sessions make up each module, each with a different title and topic. The evaluation conducted after the teachers had completed Module 1, which intended to prepare them to teach the Level 1 curriculum for promoting students' CT development using Scratch programming, was the basis for the findings provided in this study (Kong & Lai, 2021).

Angeli, Valanides and Christodoulou (2016) defined content knowledge (CK) in the context of CT education as an understanding of skills such as sequencing, loops, conditionals, and data, and PK as general pedagogical knowledge applicable to all other content domains, such as the use of questions to promote comprehension, using examples, explanations, and demonstrations, in addition to CT.

When CT is taught, it should include knowledge of how to use technical instruments (technology), knowledge of how to teach computational reasoning and the subject (pedagogy), and comprehension of CT and the subject (content) (Kale et al., 2018).

While multiple types of knowledge and understanding are necessary for a well-developed TPACK for promoting CT, incorporating technological tools into the classroom requires the belief that technology is beneficial for teaching and learning (Ertmer et al., 2012). Teachers' perceptions of value in the CT curriculum can be based on how it integrates with their instruction. Without understanding the value that CT adds to their classrooms regarding technology usage, content learning, and pedagogy, teachers may believe it is useless and will be reluctant to teach it (Kale et al., 2018).

Mouza et al. (2017) used the construct of TPACK in their study in two ways: as a framework for designing the educational technology course and as an analytic lens for examining pre-service teacher outcomes as illustrated in course products. The purpose of the study was to describe the development of an educational technology course for pre-service teachers who focus on integrating CT in K-8 classroom settings. Following that it analyses how the course affects pre-service teachers' attitudes and understanding of CT principles and how such knowledge may be coupled with content and pedagogy to create meaningful student outcomes. The data were gathered through a self-reported survey and case studies examining the design, execution, and results of CT related lessons in K-8 classrooms. According to the results, the training had a favourable effect on pre-service teachers' knowledge of CT concepts, tools, and practices. Nonetheless, some participants demonstrated only a cursory understanding of CT and could not design lessons that meaningfully integrated CT concepts and tools with disciplinary content and pedagogy (Mouza et al., 2017).

Çakıroğlu and Kiliç (2020) used the TPACK framework to provide a course model with data collecting tools for assessing teachers' pedagogical subject knowledge in teaching CT via robot programming. The study addressed many measures to be employed during teacher training sessions by utilising the benefits of virtual educational robotics. To begin, they provided information about lesson plans, curriculum, and students' requirements to the teachers, which included robotics concepts. Second, they needed to activate and expand teachers' robotics programming and CT knowledge and practices related to teaching students to solve robotic challenges in an inquiry and design-based context. While teachers were challenged with sample robotic challenges during the course, they were also assigned homework exercises to reinforce the information obtained during the course were urged to include robotics in their classroom instructions (Çakıroğlu & Kiliç, 2020).

Researchers, such as Angeli et al. (2016) and Bers et al. (2014) have used the TPACK method to define what teachers need to know outside the programming language or technique being used to teach CS and CT skills.

In 2015, Angeli et al. (2016) enrolled fifteen elementary school teachers pursuing a master's degree in instructional technology in a course focused on teaching CT in K-6 classrooms. None of the teachers had prior expertise in CT or computer programming. The teachers met weekly for 13 three-hour sessions. The participants worked hands-on with the Scratch computer programming environment. The learning-by-design approach was used in the course to engage teachers in creating models of various problem situations prior to developing computer programs to solve the problems (Angeli et al., 2016).

Bers et al. (2014) evaluated a PD workshop focused on robotics, engineering, programming, and pedagogies for early childhood schools using the TPACK framework. Bers et al. (2014) described the convergence of content, pedagogical, and technological knowledge, including knowledge about robotics as a "domain that integrates technology and engineering" and knowledge about how to teach STEM and CT. Few robot construction sets are designed especially for young learners with building and programming skills (Bers et al., 2014).

Kong et al. (2020) developed and adopted a teacher learning programme to increase teachers' expertise in promoting CT in primary school students through a programming curriculum (discussed in more detail in 2.13.3). The study's results will help shape teacher preparation initiatives focused on CT creation through programming education that adheres to the TPACK framework.

2.13.3 Professional development frameworks for computational thinking

To address the challenges of training teachers to integrate CT into their classrooms, few CT frameworks were created for teacher PD and to guide the integration. Three frameworks are highlighted here and described in-depth in the following sub-sections.

ADAPTTER Framework

Kirwan et al. (2022) conducted research demonstrating the development of a framework for teaching CT to secondary school students. Educational Design Research (EDR) led to the creation of the ADAPTTER educational framework. EDR is advised when topic knowledge is new, instructors' knowledge or availability of instructional resources are limited, teaching and pedagogical expertise are ambiguous, and complex societal issues are present (Kelly, 2013). ADAPTTER is an acronym for Activities, Demonstrations, Application, Pre-activation, Transparency, Theory, Exemplification, and Reflection. These elements produced a course in CT of high-quality, practical, interesting, effective, and low threshold. Low threshold pertains to prerequisite resources and knowledge. The course is meant to be taught using technical tools in a typical Irish secondary classroom, such as a projector and a teacher's computer. No prior understanding of CS or programming is required for this course (Kirwan et al., 2022).

This EDR study followed the iterative three-phase strategy Plomp (2013) suggested: preliminary, prototype, and summative. The preliminary analysis phase's objective was to better understand "the educational dilemma" of how to effectively teach and learn CT in the Irish setting. It included context and needs analysis, literature research, and developing a hybrid conjecture map for the intervention's design. In addition, an exploratory survey and school visits were conducted. This phase's findings led to the creation of Version 1 of the CT course (Kirwan et al., 2022).

The prototype stage involved piloting, developing, and assessing the CT course and the expanding instructional framework. It included Version 1 and Version 2 (Kirwan et al., 2022).

The summative phase is the concluding evaluation step. Evaluations were conducted to determine whether the ADAPTTER-based training succeeded, engaging, of good quality, low threshold, and practical (Kirwan et al., 2022).

Version 1 of the course was created from a combination of collaborative unplugged activities. The unplugged exercises were picked or created with real-world issues in mind. The course material was condensed into a teacher's manual, including lesson plans, each lesson's design layout, lecture notes on the same background material on CT, and directions for using the unplugged activities (Kirwan et al., 2022).

Two subject matter experts with competence in CS and familiarity with the subject or CT were also provided with Version 1 of the guide. The relevance of the course material to CT and its ability to advance students' comprehension of the same were issued to these experts and addressed in their responses Kirwan et al., (2022).

The examination of gathered data verified the following design elements: unplugged activities, knowledge demonstration, and knowledge application. The obtained data also validated the course's effectiveness and engagement (Kirwan et al., 2022).

Using unplugged activities was a low-barrier (including low-cost) technique to teach CT. It also permitted a systematic evolution of CT principles. This has been reported to lessen the cognitive burden of CT and enable the introduction of its ideas prior to programming. Finally, it has been shown that the ADAPTTER framework generates low threshold, high-quality, engaging, effective, and valuable courses. Using the ADAPTTER framework, teachers and academics may create a CT course, grasp its components, and facilitate discussions (Kirwan et al., 2022). The ADAPPTER framework is shown below in Figure 2.5.

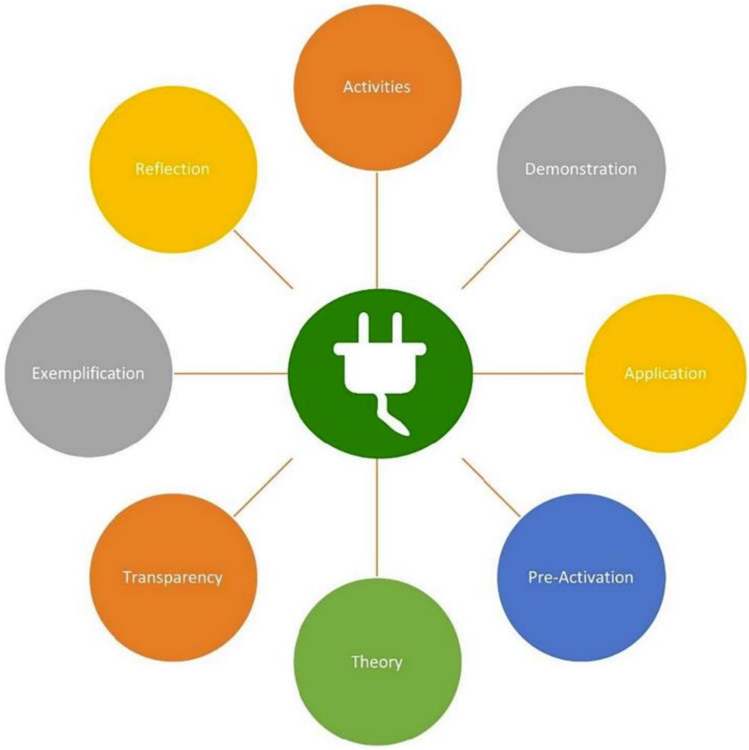


Figure 2.5 The ADAPTTER framework (Kirwan et al., 2022)

Computational thinking teacher development (CTTD) framework

In another recent study, a framework for teacher development for CT was developed by Kong and Lai (2022). The framework emphasises four content-related aspects of Mishra and Koehler's (2006) TPACK (Technological Pedagogical Content Knowledge) model.

TCK focuses on learning to programme using a block-based programming environment. Knowledge of CT principles, practices, and views is the subject of CK. PCK focuses on CT pedagogies that exclude the usage of programming environments. TPACK emphasises the integration of technology, pedagogy, and the content of CT in context. Based on these factors, a seven-step lesson structure for learning to teach a unit of curricular material was suggested (Kong & Lai, 2021).

First, content knowledge (CK) refers to the definition of CT, including CT concepts, practices, and views (Brennan & Resnick, 2012). Second, technological content knowledge (TCK) refers to the understanding of utilising block-based programming environments, such as Scratch and App Inventor, to develop the CK of students throughout the programming process. Third, PCK refers to the ability to construct pedagogical activities to develop students’ CT without programming environment-related tools. Fourth, technological pedagogical content knowledge (TPACK) integrates technology, pedagogy, and the CK of CT in the context of improving students' CT (Kong & Lai, 2021).

They identified three primary pedagogies that do not entail the usage of a block-based programming environment and may be utilised to cultivate students' CT to become creative problem-solvers (Kong & Lai, 2021).

• “To play, to think, to code, and to reflect” pedagogy for developing programming planning.

• Unplugged computational practice and concept development.

• Project-based learning led by design thinking to cultivate innovative problem-solvers.

CT teachers should be able to blend CT material, technology, and pedagogy in a specific environment for pedagogical delivery through both plugged and unplugged modes. The idea is to build contextualised curricular units and empower students to take charge of their learning alone and in cooperation with peers through pair programming (Kong & Lai, 2021).

The components of the computational thinking teacher development (CTTD) framework consist of the four dimensions based on the TPACK model: CK, TCK, PCK, and TPACK, as shown in the shaded region in Figure 2.6. They solely concentrate on the four content-related dimensions rather than the PK, Technological Knowledge (TK), and employing block-based programming environments to teach material other than programming (TPK) that make up the original TPACK model by Mishra and Koehler (2006), which included seven knowledge dimensions (Kong & Lai, 2021).

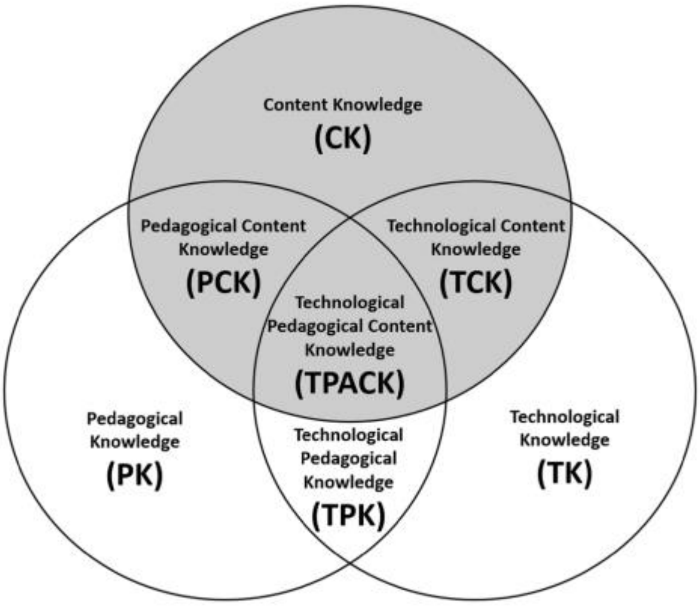


Figure 2.6 Computational thinking teacher development (CTTD) framework highlighting content-related knowledge dimensions of CK, TCK, PCK, and TPACK (in the shaded areas) in the TPACK model (Kong & Lai, 2021)

To prepare teachers to apply the curriculum, they provided a PD programme for teachers from the schools participating in the project. The programme utilised a module-based approach with four modules following the CTTD framework. Each module consists of four 3-hour courses, each with a unique title and topic. Modules 1 through 3 focus, correspondingly, on Levels 1 through 3. Module 4 aims to enable teachers to become leaders in CT education, capable of building their school-based curricular modules and promoting the development of other CT educators. Teachers were strongly encouraged to enrol in the modules according to their professional requirements. This study's conclusions are based on the analysis after the teachers finished Module 1— the first stage in training them to teach the Level 1 curriculum for promoting children’s CT development through Scratch programming. Throughout Module 1, the curricular units utilised a seven-step structure based on the CTTD framework to familiarise teachers with delivering a CT lesson (Kong & Lai, 2021). As seen in Figure 2.7, this seven-step method is based on the four content-related knowledge dimensions of the CTTD framework.

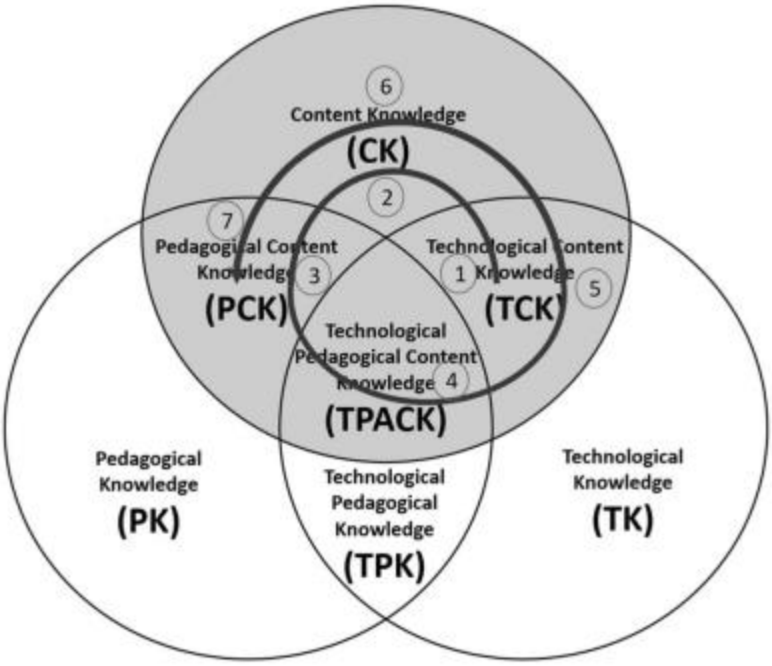


Figure 2.7 A seven-step guide for learning how to teach a unit of CT curriculum content based on the CTTD framework (Kong & Lai, 2021)

These seven phases encompass the four types of knowledge: CK, TCK, PCK, and TPACK. Through repeated exposure to these programme phases, teachers can comprehend how to teach and construct their own CT lessons. According to the findings of the course evaluation, teachers believed that the programme's design might assist them in developing their capabilities in teaching CT, with the “seven stages” being the most commonly mentioned part of the course (Kong & Lai, 2021).

The seven phases give a blueprint for teachers to follow when delivering a curricular unit in CT. This is helpful for teachers who lack confidence in teaching CT and have no idea how to teach CT. As teachers get more familiar with the seven processes and the links between the four dimensions of the CTTD framework, they will likely be able to use these steps more flexibly and in a different order (Kong & Lai, 2021).

3C Professional Development Framework

Jocius et al. (2020) created the Code, Connect, and Create (3C) PD framework to aid middle (ages 11-13) and high (ages 14-18) school content area teachers in introducing CT into their classes. Three main elements make up the suggested (3C) PD model: Code (Bootcamp), Connect (tying discipline content and pedagogy to computational thinking), and Create (the development of CT-infused learning segments). The 3C model is an integral part of a three-year research project, Infusing Computing, which intended to describe how middle and high school teachers construct and deliver interdisciplinary, CT-infused curricula (Jocius et al., 2020).

The research and facilitation team, which comprised computer scientists, education faculty members who have taught in the classroom, in-service CS teachers, and in-service topic area teachers, drew on their knowledge and expertise to establish the 3C design and development process (Jocius et al., 2020).

As teachers strive towards the 3C model's common objective of incorporating CT into their curricula, they broaden the models, terminology, modes of interaction, and participation structures within the CS subject. Participants had to apply their growing knowledge of CT to build a lesson plan using Snap!, a programming language based on Scratch, by the end of each week-long PD session. The following elements were to be produced by teachers: a Snap! prototype, a thorough lesson plan, and extra pedagogical materials like slides, links, or handouts. All components of the 3C model were established to guide teachers toward an increasingly nuanced understanding of CT. In addition, 3C strives to assist teachers in identifying possibilities for interdisciplinary and disciplinary integration (Jocius et al., 2020). Figure 2.8 shows the 3C framework.

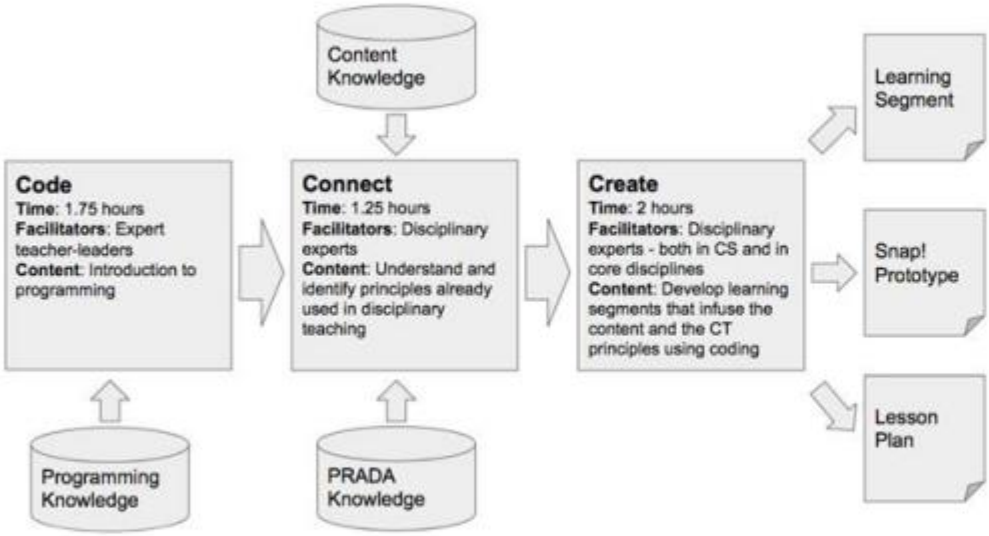


Figure 2.8 Code, Connect, Create (3C) Professional Development Framework

The first 3C model component, Code, aims to provide learners with the chance to deepen their comprehension of CT principles, discover programming ideas, and obtain practical programming experience using Snap! (Jocius et al., 2020).

To improve their comprehension of the CT concepts defined in the PRADA model by Dong et al. (2019), teachers are encouraged to collaborate with peers from related subject areas during the Connect sessions. The information covered in the Connect sessions reinforces the language and principles of CT covered in the Code sessions. These daily sessions concentrate on a specific PRADA subject (pattern recognition, abstraction, decomposition, and algorithms) and start with a whole-group discussion in which teachers present specific instances of how their prior teaching methods featured PRADA components, even if they did not use the PRADA language. Teachers evaluate their standards and generate examples of plugged or unplugged activities that might assist students' disciplinary understandings and CT ideas to help them uncover integration opportunities and highlight the underlying linkages between CT and their disciplinary teaching (Jocius et al., 2020).

The 3C models create sessions that allow participants to participate in CT infusion and programming as both students and teachers. Teachers create goals for developing and constructing their learning segment, explore areas of need, and reflect on their new learning from the two-morning sessions during the conversation that starts each

Create session. Then, participants create their learning segments either individually or in teams. On the last day of the course, individuals displayed their learning segments for other participants and invited visitors and school officials during a Demo Fair. Teachers’ usage of Create session time depends on their intended learning segment; therefore, it is adaptable (Jocius et al., 2020).

Teachers’ self-efficacy and perceptions regarding the integration of CT into the disciplinary curriculum were examined as part of the 3C model’s impact analysis. The findings of a quantitative survey analysis showed that teacher self-efficacy had significantly changed (Jocius et al., 2020). The 3C model also enhanced teachers' knowledge, abilities, and performances to enable them to use their new understanding in the classroom, according to further analysis of teacher implementation surveys (n=26) (Jocius et al., 2020).

The study's conclusions have significant ramifications for expanding access to CT by helping teachers include CT in middle and high school content classes. According to data analysis following its deployment, the 3C model was successfully used to assist changes in teacher attitudes and self-efficacy regarding the role of CT in P-12 schools. Teachers also valued learning to code, critically thinking about disciplinary material and CT concepts, and cooperating with colleagues to facilitate discipline-specific and transdisciplinary learning (Jocius et al., 2020).

The three existing PD frameworks for CT are compared in Table 2.3 below according to the criteria of framework elements, teaching approaches that can be used, target population, target education level, and intervention country context.

Table 2.3 Comparison of the three existing frameworks

	3C	ADAPPTER	CTTD
Components	**Code** (*programming knowledge*) **Connect** (*content and PRADA knowledge)* **Create** (*Learning segment, Snap! Protoype, Lesson Plans*)	Activities, Demonstrations, Application, Pre-activation, Transparency, Theory, Exemplification, and Reflection	Content knowledge (CK) Pedagogical Content Knowledge (PCK) Technological content knowledge (TCK) Technological pedagogical Content knowledge (TPACK)
Teaching Approaches	Programming (Snap!), Unplugged	Unplugged	Programming (Scratch & App Inventor), Unplugged, project-based learning
Target population	In-service teachers (Math Science, Social studies, English, Arts)	In-service teachers	In-service teachers
Target Level	K-12: middle and high school	Secondary schools	Primary schools
Country context	USA	Ireland	Hong Kong

2.14 Learning theories in the professional development of teachers for CT

This section describes the learning theories that have been used in training teachers on how to integrate CT through professional development programmes. The two learning theories described in the next sub-sections are situative learning and constructionism.

2.14.1 Situative Learning Theory

The situated learning paradigm is based on cognitive theories and social psychology, which emphasise the importance of context-specific social engagement, learning communities, and authentic learning (Brown et al., 1989). According to the situated learning paradigm, learning occurs because of behaviour taken in social interactions within an engaging and shared environment (Brown et al., 1989; Lave & Wenger, 1991; Takahashi, 2011).

Learning is conceptualised from a positioned viewpoint as a method of enculturation into a domain-specific culture. The above is sometimes called a “Community of Practice” (CoP), a concept coined by Lave and Wenger (1991) to describe “practices in which individuals have learnt to participate, rather than on knowledge that they have acquired" (Gramm et al., 2012, p. 8). From this vantage point, the objective of education is to become a full member of a CoP.

Although Lave and Wenger (1991) illustrate their point with illustrations of apprentices in diverse sectors, Putman and Borko's work demonstrates the importance of situated learning for teacher professional growth (Putnam & Borko, 1997; Putnam & Borko, 2000). For example, situating learning does not require all teacher learning opportunities to occur in the classroom, but rather that the learning context is situated in their experience: "the situative perspective holds that all knowledge is (by definition) situated. The question is not whether knowledge and learning are situated but in what contexts they are situated. For some purposes, situating learning experiences for teachers outside of the classroom may be important–indeed and essential for powerful learning" (Putnam & Borko, 2000, p. 6).

Three critical principles exist within situated learning theory. To begin, the CoP concept emphasises the critical role of relationships in learning. Second, the concept of legitimate peripheral participation (LPP) recognises that learners progress from beginner to professional status as they integrate into the CoP. Third, there is tension between new and established members, which Lave & Wenger (1991) refer to as continuity replacement, as new ideas overtake established ones and the group changes.

Situative learning theory and the integration of CT into the professional development of teachers

Ozturk et al. (2018) used a situative learning perspective to examine how elementary teachers with little knowledge of CT and PBL viewed applying CT to PBL. Every month, teachers collaborated in grade level teams to prepare, study, and update curricula while they learnt how to adapt CT. Although the teachers had little or no previous knowledge of CT, they were all accomplished elementary educators. The team drew on the literature on successful PD to involve teachers in professional learning. To begin, teachers were recognised as adult learners who brought important skills and experience as elementary educators and curriculum designers to the school (Knowles, 1984). Since the professional learning was intended to be problem-centred, applicable, and realistic, the teachers developed into a relational group collaborating to solve instructional problems.

Via project-based, integrated learning, elementary teachers will bring to life what Seymour Papert meant when he said, “The problem of the education innovators is to create situations in which you [children] need it [knowledge], and to create means by which you can find the knowledge when you need it for your purposes” (Papert, 1997, p.33).

Killen et al. (2020) examined the effectiveness of a science teacher CT inquiry group (STIGCT) as a paradigm for PD. By observing participants as they collaborated to create multiple CT-infused elementary science lessons, they determined how the model's structure, specifically the diversity of participants, the collaborative structure of activities, and the time for iteration and experimentation, leveraged and exceeded the benefits of a CoP model. The STIGCT is the first research effort to create a standard of practice for educating teachers about CT, filling a knowledge gap relevant to individuals involved in CT education (Killen et al., 2020).

2.14.2 Theory of Constructionism

Seymour Papert of the Massachusetts Institute of Technologies developed a theory of learning based on Piaget’s constructivism. Papert worked with Piaget in Geneva in the late 1950s and early 1960s. In his own words: *“Constructionism — the N word as opposed to the V word — shares constructivism’s view of learning as “building knowledge structures” through progressive internalization of actions… It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it’s a sand castle on the beach or a theory of the universe”* (Papert & Harel, 1991*, p.1).*

Since Papert’s approach places a greater emphasis on learning by creating than on total cognitive potentials, it enables one to comprehend how concepts are shaped and transformed when conveyed through various media, when actualised in specific ways, and when worked out by individual minds. The focus turns away from universals and toward individual students’ interactions with their preferred representations, artefacts, or objects to think with (Ackermann, 2001). Constructivism’s extension, constructionism is also a learning philosophy and an educational strategy (Kafai, 1996). Constructionism shares constructivism’s connotation of learning as constructing information constructs but emphasises that learning is enabled by the creation of observable artefacts or items that can then be exchanged and explored with others (Papert & Harel, 1991).

Constructionism views learners as active constructors of their own experience and believes that people learn more effectively while constructing individually significant artefacts, "whether it's a sandcastle on the beach or a theory of the universe" (Papert & Harel, 1991, p. 1). These artefacts, which Papert (1980) refers to as "objects to think with" (p.12), facilitate the creation of concrete models of thought and knowledge about phenomena. The ability to modify these objects, to change and refine them constantly, or to play with them to determine how they function, lends itself to a concrete form of thinking. Papert and Harel (1991) say this transforms learning into an iterative and cumulative process that incorporates both preparation and bricolage types. Turkle and Papert (1990) coined the term "epistemological pluralism" to refer to the "validity of multiple ways of knowing and thinking" (p.129).

Constructionism further emphasises the social nature of learning, emphasising that practices, such as making, constructing, or programming offers a rich framework for learning by allowing the learner to create artefacts that others can see and criticise. The artefacts allow others to participate in the thought process while benefiting the learner's thinking from different perspectives and discussions (Butler, 2007). In this way, artefacts or 'thinking things' serve as a bridge between sensory and abstract experience and between the human and social realms (Ostwald, 1996). By conversing about their or another's artefact, they may facilitate the creation of a mutual understanding and lay the groundwork for new understandings (Ackermann, 2001). Thus, constructionism means a method of construction, both of artefacts and modern understandings. When artefacts and mutual awareness are coupled through processes of representation and interpretation, shared information is created (Ostwald, 1996).

The constructivist approach is predicated on the notion that students actively participate in their educational process, and that knowledge is created through experiences. Each individual considers their experience and combines new ideas with their past knowledge as events unfold (Kurt, 2021; Loi, 2004). Although the constructivist method requires more effort or time to apply, the instructor must be dedicated to fostering a learning environment in which classes entail the co-creation of knowledge (Kenny & Wirth, 2009).

Theory of constructionism and the integration of CT into the professional development of teachers

From a constructionism viewpoint, CT can be interpreted similarly to how Papert viewed computer programming; that is, CT can be viewed as both a skill to acquire and a method of acquisition – “to create, discover, and make sense of the world, using digital technologies as extensions and reflections of our minds” (Angevine et al., 2017, p.21).

Undoubtedly, an integrated strategy must set up a suitable constructionist educational environment to foster CT abilities. A properly constructed constructionist environment should be the foundation for an integrated educational process (Dolgopolovas et al., 2019).

Teachers' initial ideas about CT and how CT could be occurring in their classrooms as tools to be leveraged during professional learning and other instructional opportunities for teachers were interpreted by Rich, Spaepen et al. (2019) from a constructivist viewpoint. To capitalise on the links teachers saw between CT components and their current mathematics teaching methods, they approached their work with them through unplugged activities. This relates to a critical characteristic of positive PD: *a content focused approach*.

Cetin (2016) and Marcelino et al. (2018) built and delivered education courses, specifically for elementary school teachers to teach CT concepts and Scratch using constructionist principles that emphasise the importance of the topic in knowledge creation.

Hickmott and Prieto-Rodriguez (2018) discussed their experiences developing and strengthening teacher professional learning and recognised that it was only after adopting a constructionist perspective on their work, they made positive improvements to their PD programme. The mechanism by which Hickmott and Prieto-Rodriguez (2018) identified the reforms emphasised the importance of pursuing such aspects of PD that are not prominent in recent computing education literature. Hickmott and Prieto-Rodriguez (2018) agree these types of PD are beneficial for training new teachers and assisting in the development of in-service teachers. These courses should be planned to be current, particularly for primary school teachers. Hickmott and Prieto-Rodriguez (2018) held seminars in Australia for pre-service and in-service teachers who integrated constructionist concepts. The sessions of these workshops have included activities with step-by-step instructions, collaborative problem-solving exercises and lesson-planning activities, and talks by academics and industry professionals. The findings of the 2016 polls seemed to show that incorporating constructionist methods increased overall satisfaction with the workshop (Hickmott & Prieto-Rodriguez, 2018).

Hickmott and Prieto-Rodriguez (2018) incorporated a growing array of hands-on tasks in the classrooms, as indicated by survey respondents. However, it was critical to implement hands-on exercises that included open-ended problem-solving, in which teachers constructed their skills in-depth rather than relying solely on straightforward step-by-step teaching. They considered recommendations made by many constructionist PD researchers and professionals when incorporating these constructionist practices. Martinez and Stager (2019), for example, contend that PD is often “too meta” (p. 200) and propose that teacher educators have PD through which teachers observe learning from the viewpoint of a student. Martinez and Stager (2019) conceptualised and incorporated Constructing Modern Knowledge, which provided teachers with four days of uninterrupted time to collaborate on a design project in the context of making, tinkering and engineering in the classroom.

A similar point about PD settings was made by Brennan (2015), saying that “teachers should have learning experiences comparable to those of their students, within a supportive community of fellow teachers” (p.293). Brennan (2015) designed and introduced ScratchEd, a variant of constructionist-based PD for computing. ScratchEd's primary goal is to promote the utilisation of technology to develop practical ventures rather than focusing only on the use of particular technologies. ScratchEd, like Martinez and Stager’s (2019) Constructing Modern Knowledge classrooms, is intended to immerse teachers in classroom environments similar to those of their pupils. This study is grounded on constructionism theory.

2.15 Participatory Design

Participatory Design is explored in this section as a structured way to actively involve teachers in the professional development programme.

2.15.1 Background

Participatory design emerged in the 1970s and 1980s in Scandinavia. This early Scandinavian work was inspired by a Marxist contribution to collective worker empowerment and workplace democracy. Participatory architecture involves customer involvement when designing for job experience. Participatory design is a collaborative mechanism for developing (social and technological) processes that include human labour. It is predicated on the belief that users should be interested in the interfaces they use and that all people, including and particularly users, should have fair input into interaction design (Muller & Kuhn, 1993).

The participatory design model is constructivist in the context described by Mirel (1998). It views information as a result of interactions between individuals, activities, and artefacts — knowledge does not exist solely in mind; it is a state of a particular context. One of the most distinctive and prominent concepts of participatory design is the language game (Ehn, 1989, p. 17): bridging the realms of researcher-designers and consumers by establishing a shared "language" or mode of interaction compatible with all parties.

The essence of participatory design is to empower users and foster communication and collaboration between designers and users. It makes a concerted effort to involve all stakeholders active in the design process to ensure that the product/outcome satisfies all stakeholders' needs and expectations. It is more concerned with the design process and procedures than with its appropriateness and perfection (Schuler & Namioka, 1993). Participatory design is a collection of theories, practices, and research that emphasise end users as active participants when developing artefacts or products (Kinley, 2015).

Participatory design has been applied in a variety of ways and with a variety of common interaction codes. In certain cases, participatory design restricts user involvement to providing feedback for expert planners to consider, a practice known as consultative design (Mumford, 1981). Other methods empower consumers to take full responsibility for the final result, a process Mumford (1981) refers to as consensus design.

Druin (2002) coined the term "design partner" after analysing how children participate in the design and defining four roles: user, tester, informant, and design partner. Users are the primary audience for existing technologies, and their behaviours are studied to develop the system. The testers use technology not yet approved for commercial use to expand the technology's reach. Informants participate actively in the design process and provide feedback prior to, during, and after the technology is created. Partners are recognised as legitimate decision-makers and given equal status with creators and researchers. This function typology was later extended to include two additional roles: co-researcher and protagonist (Iversen et al., 2017). Co-researchers assist researchers in collecting and analysing data to investigate the usage sense, and protagonists take control of the concept process and assume responsibility for its continuation (Bødker & Kyng, 2018; Duarte et al., 2018). Genuine involvement refers to this shift in participants' roles (Bødker, 2003).

According to the Routledge International Handbook of Participatory Design, Genuine participation is characterised as a fundamental transformation of the users' position from mere informants to valid and accepted participants in the design process (Simonsen & Robertson, 2012). This type of involvement occurs as participants express their experiences by more than answering questions. They do so by painting, sketching, and other means. This demonstrates how participatory design is about assisting participants in recognising the existence of alternate options, negotiating what is most important to them among these choices, and exerting control over how these choices are followed (Bødker, 2003; Bødker & Kyng, 2018; Iversen et al., 2012). To achieve genuine engagement, members must have access to relevant knowledge, an autonomous status, the ability to participate in decision-making, adequate design processes, and organisational versatility (Clement & den Besselaar, 1993; Kensing & Blomberg, 1998).

2.15.2 Constructing knowledge through design

Design activities during participatory design serve as pre-conditions for knowledge generation (Frauenberger et al., 2015). Practical work creates a dialogic environment conducive to constructing new knowledge through exchanges between diverse stakeholders (Bannon et al., 2018; Iversen et al., 2017). This new knowledge can include the following: 1) the social context in which the study was conducted; 2) the design outcomes; 3) the methodology for conducting and analysing design; and 4) the design principles and structures (Frauenberger et al., 2015).

Knowledge about the social context includes, but is not limited to, participant conceptualisations of their job and living habits. This understanding is created in a two-way process in which members focus on their methods of operation, and planners and analysts seek to comprehend the meaning. The objective here is collaborative learning, in which the planners and analysts want to consider the participants while the participants become mindful of their existing activities and imagine ways to enhance them (Kensing & Blomberg, 1998). In this sense, participatory design is analogous to ethnographic research: interpreting participants' behaviours requires an understanding of their meaning, and simply questioning participants is insufficient since what they say is not always synonymous with what they do (Falcão et al., 2018; Lindtner & Lin, 2017; Grönvall & Kyng, 2013).

2.15.3 Participatory Design Methodology

Participatory design has its own methodological perspective, procedures, and approaches, much like its underlying methodology, participatory action research. Implementations of participatory design differ in their attention to rigour and validity, according to Spinuzzi (2005), but they all indicate a commitment to continuous, rigorous study according to grounded methodological principles. As the name suggests, the technique is as much about design as it is about research, creating artefacts, systems, work structures, and practical or tacit knowledge. The participatory design utilises a variety of research methods (such as ethnographic observations, interviews, analysis of artefacts, and sometimes protocol analysis), but these methods are always used to iteratively construct the emerging design, which simultaneously constitutes and elicits the research results as co-interpreted by the designer-researchers and the participants who will use the design (Spinuzzi, 2005).

2.15.4 Participatory Design Stages

According to Spinuzzi, three basic stages are present in most participatory design research (2005, p. 167), and they are presented in Table 2.4 below.

Table 2.4 Participatory Design Stages (Spinuzzi, 2005)

Stage	***Description***
1. Initial exploration of work	Designers meet the users and familiarise themselves with how the workers work together.
2. Discovery process	Together, designers and users clarify the users' goals and values and agree on the desired outcome of the project.
3. Prototyping	Designers and users iteratively shape artefacts to fit into the workplace envisioned in the discovery process.

Participatory designers view themselves as facilitators who seek to enable users to make their own choices (Spinuzzi, 2005). To attain this objective, the participatory design promotes *co-research* and *co-design*: researcher-designers must arrive at findings with users. Participatory design covers the rethinking of workplaces, work organisations, and tools. Through its iterative form, it allows employees and academics to study critically the effects of these ongoing incremental redesigns (Spinuzzi, 2005).

The co-design approach is rooted in three interwoven theories: constructivism theory, participatory design theory, and experiential learning theory (Agbo et al., 2021). Participatory design theory, for instance, is grounded in constructivism theory (Spinuzzi, 2005). While constructivism theory posits that learning is an active, constructive process in which students create their mental representations of learning objectives, participatory design theory focuses on methodological approaches that ensure users of technological artefacts are involved in the entire design process (co-design) of those artefacts to produce more efficient and usable systems (Agbo et al., 2021). The experiential learning theory considers learning a process in which concepts are generated from and continually transformed by the experience, i.e., "ideas are not fixed and immutable elements of thoughts but are formed and re-formed through experience" (Kolb, 2014, p. 26).

The conventional face-to-face technique is frequently employed for co-designing educational or commercial products. In this technique, the researcher/facilitator meets with co-designers to create a new item through a collaborative process (Agbo et al., 2021).

While using the online co-design (OCD) approach in research, Agbo et al. (2021) presented insights on the viability and acceptability of a co-design process within a developing country, specifically the African context. Several important lessons were learnt, which might help researchers, designers, educators, and other stakeholders who wish to perform a comparative study in a similar environment. They explored these lessons in five stages: (i) planning and engaging, (ii) exploring, (iii) designing, (iv) discussing and deciding, and (v) changes and feedback, as depicted in Fig. 2.5 as a process flow that relates to the lessons learnt through the implementation of the OCD technique. Each stage's actions serve as input for the subsequent stage (Agbo et al., 2021).

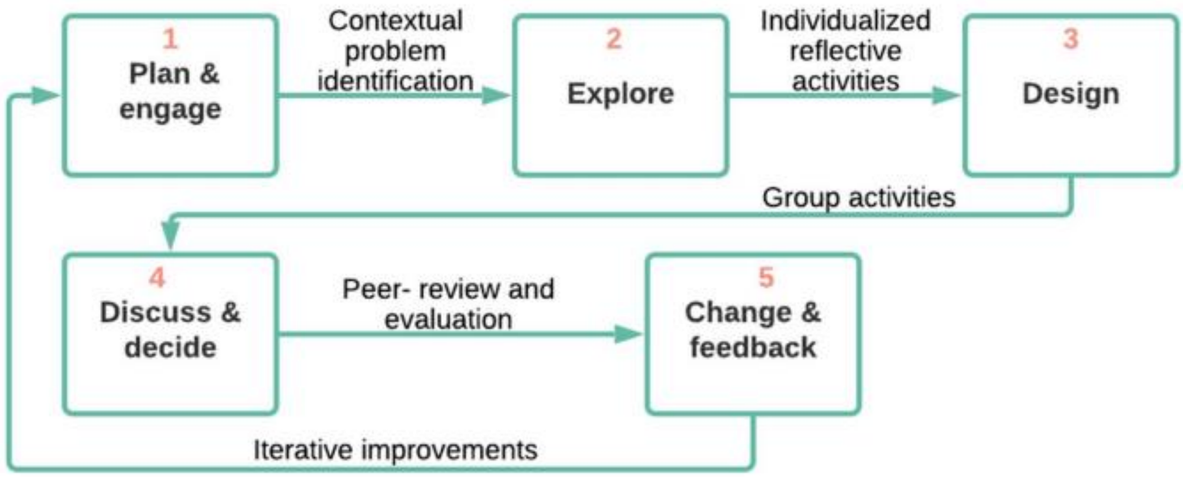


Figure 2.9 Process flow of the five-stage implementation of OCD process (Agbo et al., 2021)

2.15.5 Tools and techniques of participatory design

Brandt et al. (2012) have proposed a broader framework that can assist us in determining which tools and strategies best apply to certain circumstances. The framework overviews tools and approaches for involving non-designers in certain participatory design activities. It has three dimensions: *form, purpose, and context*.

- **Form** describes the action taking place between the participants in an activity and is described as *making, telling, and enacting.*
- **Purpose** describes why the tools and techniques are being used and are described along four dimensions: *probing* the participants, *priming to immerse* them in their domain of interest, better *to understand* their current experience and *generation* ideas for the future.
- **Context** describes where and how the tools and techniques are used. Context is described along these four dimensions: *group size & composition, face-to-face vs online, venue, and stakeholder relationships.*

The aim is to use these principles while choosing and developing tools for participatory design. Making, telling, and enacting while conducting workshops during the design process will help support participation (Brandt et al., 2012).

- **Telling** is mostly about existing practices and the telling of needs and dreams. It is about what is and what could be. The most important is that the participant is heard and gives the user a sense of commitment and participation (Brandt et al., 2012).
- **Making** activities often use physical artefacts and can roughly be divided into three categories: participatory prototyping, probes and the use of generative tools. All these may describe future objects or provide different views on future ways of living. We also embody the thoughts and ideas of physical artefacts (Brandt et al., 2012).
- **Enacting** refers to activities where one imagines and acts out possible futures. These can be based on scenarios or improvisation and experimentation in the situation being improvised (Brandt et al., 2012). One aim of enacting is to develop knowledge through practical exercises.

2.15.6 Participatory design and teachers' professional development

The participatory design approach in teacher development recognises teachers as critical agents of educational transformation, repositioning them from information transmitters to creators of students' learning (Mor et al., 2012). This methodology reduces the gap between consumers and system builders, according to Chin (2004), or, in our case, between teachers and policymakers, enabling teachers to gain input into and co-own change efforts. The participatory design allows consumers to adapt the design to their requirements and contexts, resulting in a high degree of product compatibility and adoption (Damodaran, 1996). Participatory design enables teachers as design partners to create instructional content more compatible with their students and their own teaching needs (Tuhkala, 2019).

Participatory design has been used in teacher PD, with two experiments focusing on PD improvement. During a month-long initiative, researchers and high school teachers collaborated to build CT-infused mathematics and science courses (Kelter et al., 2021). Teachers learnt constructionism through the co-design process, which produced a constructionist curriculum they could apply in their classrooms. They provided three case studies to demonstrate the many ways that teachers and researchers split the work of co-design and the effects of these various co-design methods on teacher learning and classroom enactment. Some educators created their computational tools through programming, while others only contributed to their conceptualisation and left the actual development to their co-design partners. The findings show that constructionist co-design is a fruitful dual strategy for curricular and PD, while occasionally, these two objectives are at odds (Kelter et al., 2021).

The Edukata Model

As a result of the work done by Toikkanen et al. (2015), a toolkit called Edukata teachers' toolkit was created to let teachers create their learning activities and close the gap between visionary ideas and classroom practice. For educators, Edukata is a collaborative, participatory design process model. The methodology contains five iterative phases: gather participants, set up a workshop, facilitate a workshop, design learning activities, and reflect on a workshop. The steps can be combined to make a special design approach appropriate for your school (Toikkanen et al., 2015).

A key component of Edukata is workshop facilitation since this is how stakeholder views are heard. The Edukata facilitator's guidelines include: being prepared as a workshop's goal is to identify design challenges, opportunities, useful resources, and share learning activity ideas with all participants, keep work organised and accessible

(Keune et al., 2014). Results are even greater when teachers are encouraged to create their learning activities with the right facilitation and direction (Lewin & McNicol, 2014).

The PROFILES Initiative

Kyza and Georgiou (2014) investigated the effect of participatory architecture on teachers' sense of ownership of inquiry-based learning modules. The research was conducted in the framework of Cyprus's PROFILES continuous professional development initiative, which used a collaborative and participatory design approach to facilitate science teachers' PD (Kyza & Georgiou, 2014). This study enrolled 26 teachers and 171 high school students. During one academic year, teachers organised four discipline-based groups (Biology, Chemistry, Elementary Science Education, and Physics) and collaborated on the creation and eventual implementation of inquiry science learning experiences using the PROFILES method in their classrooms.

Kyza and Georgiou (2014) used face-to-face and web-based communication in their participatory design process. Technologies were chosen based on the researchers' earlier PD experiences that may provide deeper communication and collaboration experiences and could supplement face-to-face sessions. Each discipline design group met at least seven times during after-school hours, with four meetings consisting of cross-group talks and group work sessions. Additionally, each discipline group held video-conference planning meetings that lasted at least an hour.

A mixture of synchronous and asynchronous communication methods was deployed to ensure continual access to information and boost teachers' capability to engage in creating each disciplinary team. These web-based technologies included an asynchronous communication platform, a video conferencing system, and the online STOCHASMOS learning and teaching platform utilised to create their PROFILES-based learning environments by the chemistry science educator group (Kyza & Constantinou, 2007). STOCHASMOS (http://www.stochasmos.org) is a scaffolded environment that enables instructors to take a more active part in the design process and provides computer-based scaffolding for students' reflective inquiry-based explorations. Thus, the platform enabled chemistry professors to upload web-based resources and create scaffolding structures, such as data pages and explanation frameworks to support teachers' inquiry and decision-making processes (Kyza & Georgiou, 2014).

The web-based, asynchronous communication platform enabled sharing of resources in a virtual space, while the web-based video conferencing system enabled deeper interactivity during synchronous video sessions. Additionally, the STOCHASMOS platform facilitated the creation of scaffolded inquiry modules based on the European project - Professional Reflection Oriented Focus on Inquiry-based Learning and Education (PROFILES). This combination facilitated idea sharing, aided in the coordination of the design process, and facilitated instructors' collaborative reflection (Kyza & Georgiou, 2014).

According to the teachers from Kyza and Georgiou's (2014) study, participatory design is a collaborative and supportive process that fosters sharing of diverse viewpoints, promotes critical constructivism, and supports adopting modern teaching methods and technology. But the time-consuming aspects of participatory design, coordination difficulties, and disproportionate contributions of participants were described as the primary drawbacks. About this, both teachers opted to create their modules rather than using pre-made ones. According to Kyza and Nicolaidou (2017), the iterative design encourages teachers' PD by allowing teachers to focus on inquiry-based learning and teaching.

Other studies that included the participatory design of educational programmes

Al-Eraky et al. (2015) collaborated with teachers to create a faculty learning curriculum for teaching professionalism in medical education. The Department for Medical Education prepared and delivered the curriculum, which was authorised by the Deanship for Educational Development at the University of Dammam in Saudi Arabia. The study's planning and execution occurred between September 2013 and May 2014. Two guiding principles guided the programme's creation: cultural awareness and situated learning. It must exemplify professionalism in the Arabian sense and demonstrate how the fundamental principles of professionalism are operationalised in the behaviours of chosen health practitioners. In a participatory learning approach, two parties share responsibility for curriculum development: writers and students. The authors studied the literature in three areas: professionalism in the Arabian sense, instructional standards for teaching professionalism, and preparation and assessment of faculty development programmes. The programme was developed in three stages. The workshop drew 28 teachers from several health professions' education perspectives in medicine. Teachers' participation in collaborative dialogue during the orientation workshop and subsequent phases of vignette production, and students' input on their learning experiences, suggested that the curriculum was feasible, positive, and fruitful (Al-Eraky et al., 2015).

Põldoja et al. (2014) discussed the design issues associated with developing a technical solution for self- and peer-assessment of teachers' digital competencies. In Estonia, research was conducted in pre-service and in-service teacher education, and the design approach was driven by a research-based technique (Leinonen et al., 2008). The design process was divided into four iterative phases, which can occur in parallel: (1) contextual investigation, (2) participatory design, (3) product design, and (4) hypothesis development. The first phase's primary objective is to identify the environment and design issues. The primary background is teacher education, emphasising new teachers completing an orientation year in classrooms. The second stage of the research-based design process is participatory design, which involves potential users in design sessions. To convey design concepts to customers requires non-technical communication methods. Product design, the third step of the research-based design process, culminated in usage cases and simple interaction.

So et al. (2009) created an online forum for Singaporean teachers to post vibrant photographs of their activities. They conducted a large-scale online study of 1605 teachers to ascertain existing PD activities. The quantitative data gathered in this study provided a macro-level snapshot of the current state of teacher PD in Singapore's public schools. Following that they held participatory design workshops with 11 teachers in two schools to better understand teachers' perspectives on professional learning experiences and to collect ideas for creating an online video-based community. These workshops aimed to include end users, who are Singaporean teachers, early in the design process to ensure that their needs and insights were incorporated into the final design. So et al. (2009) developed a conceptual framework for an online environment for a teacher group using the ideas collected during the participatory workshops. Throughout the design process for this area, a special focus was placed on Web 2.0 technology strategies that emphasise community engagement as a critical component. They concluded by discussing the problems and difficulties encountered during this project, such as what would drive teachers to develop and share video cases and what social support system should be put in place to boost intrinsic motivation and long-term participation (So et al., 2009).

Although not focused on teacher professional development, an interesting study examines the effects of teaching CT skills through participatory design, and Participatory Debugging (PDeb) approaches by creating and producing games in App Inventor while adhering to a participatory design process and finding bugs/failures in the game's code (Theodoropoulos, 2022). In PDeb, the goal is for students to solve a problem by finding errors in their code, which will help them develop new methods and concepts for developing mobile games. The study recommends efficiently integrating games into computer education using participatory design and PDeb techniques (Theodoropoulos, 2022).

Several authors used participatory design to develop the curriculum. For example, Tulinius et al. (2012) created a curriculum to assist teachers in developing essential assessment capabilities and academic ability, and Rodrigo and Ramirez (2017) created a master course in online teaching.

2.16 Conclusion

This chapter has reviewed the current literature on CT, teachers' PD and participatory design approach to set the tone for this study. It also looked at different theoretical frameworks on which this study is based. The most important and relevant findings arising from the literature review are tools and teaching approaches used to integrate CT; how teachers were prepared for CT integration; the challenges faced by teachers when integrating CT and that developing countries' context is under-researched. The next chapter will discuss the development of the proposed PD framework for Primary CT teachers.

3 CHAPTER 3: THE PROPOSED PROFESSIONAL DEVELOPMENT FRAMEWORK FOR PRIMARY CT TEACHERS

The main objective of this study is to develop the Professional Development for a Primary CT integration framework for teachers or the **PD4PCT** framework that training providers and researchers can use to integrate CT in teachers' professional development programmes. It is necessary to create a new framework because the three existing ones don't fit the combination of developing country context and primary education level. All three were developed for a developed country context where CTTD and 3Cs frameworks involve technology (programming) while ADAPTTER is developed for secondary schools (see Table 2.3).

The proposed framework is mostly based on Desimone's (2009) conceptual perspective (section 2.13.1). According to Desimone's (2009) model, PD is a relationship between the main properties of PD, teacher characteristics such as experience and values, classroom teaching practice, and student learning outcomes. Desimone focuses on measuring PD's impact and emphasises the core features of effective PD, change in classroom practice, and student outcomes. The proposed PD4PCT focuses only on the part of her framework, namely content focus, active learning, collective participation, coherence, duration, and context.

Some elements that are similar were considered from the three existing PD frameworks for CT, namely, 3C, ADAPPTER and CTTD (section 2.13.5) frameworks. The elements considered are content knowledge from 3C (Connect) and CTTD (CK), which in both frameworks involves CT knowledge, such as CT concepts, practices and perspectives, and falls under the proposed framework's content focus component. 3Cs Create and CTTDs Pedagogical Content Knowledge (PCK) were also considered as both elements involve teachers creating CT-infused learning material and fit in with the active learning component of Desimone's conceptual framework using either programming or unplugged learning strategies.

From the ADAPPTER framework, the application and reflection elements were considered because they involve students practising and applying their CT knowledge through unplugged activities and sharing the developed artefacts with others to get feedback, respectively. The reflection element falls within the collective participation component of the proposed framework implemented through participatory design principles in this suggested framework.

The code component from the 3C framework was not considered because the proposed framework does not require programming knowledge at the start of the intervention. This is because the framework is aimed at teachers in developing countries and considers the school's context the teachers, which usually means they have no access to computers and a programming environment (Ausiku & Matthee, 2021).

Figure 3.1 below presents the framework adapted from Desimone's impact framework discussed in the following sections.

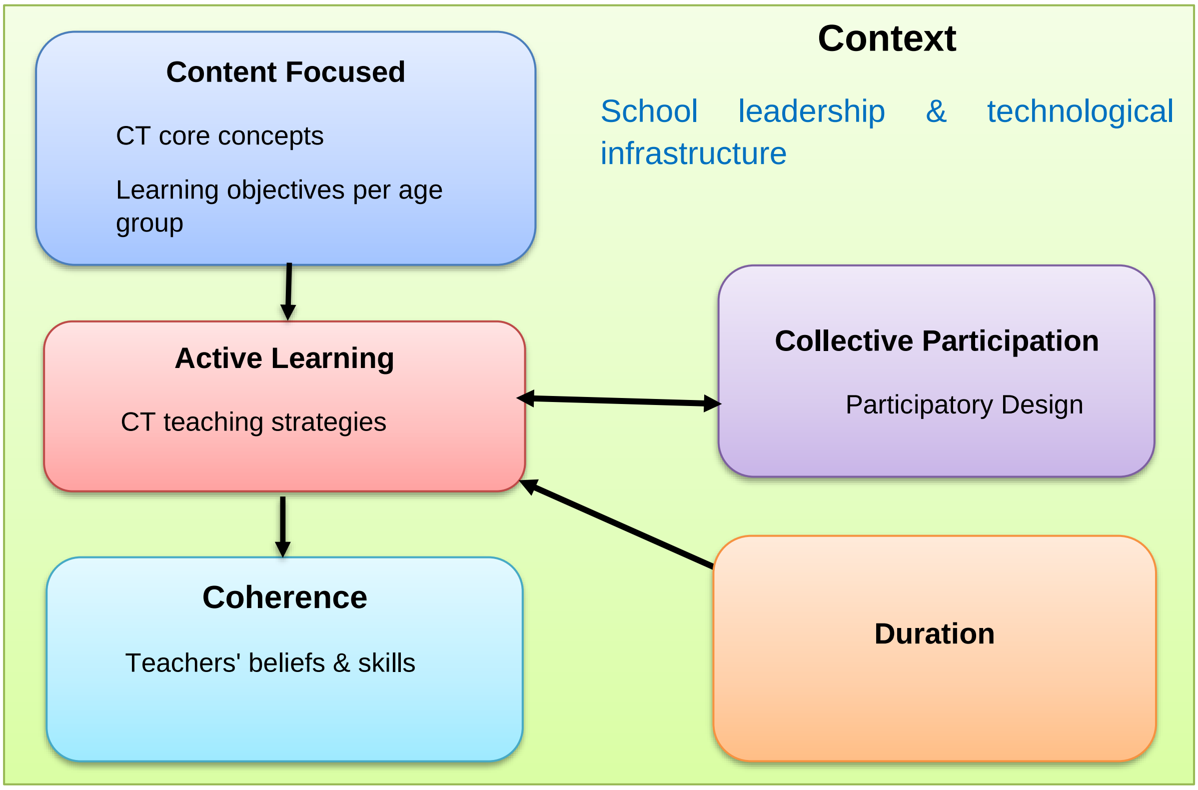


Figure 3.1 The proposed professional development framework (PD4PCT)

3.1 Pedagogical Content knowledge

Teachers should demonstrate an understanding of the CT definition, concepts, practices, and the ability to employ and modify instructional moves that facilitate student learning and application of CT across multiple subjects based on feedback from PD programmes and classroom observations. Teachers should be able to utilise a variety of materials — from paper to digital devices (where available) to aid students in understanding CT and the topics with which it is incorporated.

As indicated in the literature (see section 2.14.1) by Clement and den Besselaar (1993) and Kensing and Blomberg (1998), providing the teachers with access to relevant information and resources about the topic will establish genuine participation.

The content knowledge should be based on the core CT concepts defined in section 2.4. These concepts are algorithms, abstraction, decomposition, pattern recognition and debugging & evaluation. Some frameworks also mention data, parallelisation and simulation as CT concepts (see section 2.3).

Existing curricula for primary schools are discussed in section 2.8.

3.1.1 CT curriculum for primary schools

A suggested curriculum, based on the core CT concepts and existing curricula, is provided in Table 3.1. It is adapted from Computing At School curriculum in the UK as suggested by Angeli, Voogt et al. (2016) and Selby and Woollard (2014), proposed amendment to South Africa's CAPS Department of Basic Education of South Africa (2021) and from the 2016 Massachusetts Digital Literacy and Computer Science (DLCS) Curriculum Framework (Massachusetts Department of Elementary and Secondary, 2016).

The grades classification shown below is generally followed in developing countries that are more or less equivalent to elementary years in other parts of the world: Grades 0-2 (5-7 years), Grades 3-5 (8-10 years), and Grades 6-7 (11-12 years). The main differences between the different grades are the learning outcomes and tools used, e.g., the concept of algorithms in Grades 0-2; the learners are only expected to create a simple algorithm by putting the steps in a correct sequence. However, in Grades 3-5, they are expected to use iteration in their algorithm and be able to debug it, and in Grades 6-7, they should add conditionals and use logical reasoning to predict outputs by being aware of inputs (Table 3.1).

Table 3.1 CT Knowledge Content

CT Skills	Grades 0-2	Grades 3-5	Grades 6-7
Definition	Understand what CT is and how it can be used in the problem-solving process both with and without computers and conceptualise its integration across the curriculum.	Understand what CT is and how it can be used in the problem-solving process both with and without computers and conceptualise its integration across the curriculum.	Understand what CT is and how it can be used in the problem-solving process both with and without computers and conceptualise its integration across the curriculum.
Algorithms	• Understands what an algorithm is. • Define a series of steps to solve a problem. • Put these steps in the correct sequence. • Create a simple algorithm (e.g., getting ready for school). • Demonstrates care and precision to avoid errors.	• Understands what an algorithm is. • Define a series of steps to solve a problem. • Put these steps in the correct sequence. • Design simple algorithms using iteration. • Detects and corrects errors, i.e., debugging. • Understands that algorithms can be implemented on digital devices as programmes or paper as steps/instructions.	• Understands what an algorithm is. • Define a series of steps to solve a problem. • Put these steps in the correct sequence. • Design algorithms that use repetition and conditionals, i.e. if, then and else. • Uses logical reasoning to predict outputs, showing an awareness of inputs. • Recognises that different solutions exist for the same problem.
Abstraction	• Create a model/representation to solve a problem (i.e., using specific directional language - forward, left turn, right turn, back). • Identify key characteristics and attributes of objects, e.g. cars have a colour, type (e.g., pickup, van, sedan), number of seats, etc.	• Create a model/representation to solve a problem (i.e., create an object and assign properties). • Identify key attributes of various objects. • Use words, letters, numbers, symbols, or pictures to represent information in another form (e.g., secret codes, Roman numerals, abbreviations). • Organise information differently to make it more useful/relevant (e.g., sorting, tables).	• Create a new model/representation to solve a problem (i.e., create an object and assign properties). • Identify attributes of individual objects within a group that differ between. Members of the group and attributes that are similar). • Define a simple function that represents a more complex task/problem and can be reused to solve similar problems. • Use decomposition to define and apply a hierarchical classification scheme to a complex system, such as the human body, animal classification, or computing.

Decomposition	• Break a complex task into simpler subtasks (e.g., break a long path into a series of smaller paths that one can follow).	• Break a complex task into simpler subtasks. • Develop a solution by assembling collections of smaller parts (e.g., organising a school trip).	• Break a complex task into simpler subtasks. • Develop a solution by assembling collections of smaller parts. • Individually and collaboratively decompose a problem and create a sub-solution for each part (e.g., video game, robot obstacle course, making dinner).
Pattern Recognition	• Identifying patterns and commonalities in artefacts. • Identify common patterns and similarities between older and newer problem-solving tasks, and use sequences of instructions to solve a new problem.	• Identifying patterns and commonalities in artefacts. • Remix and reuse (by extending if needed) resources previously created. • Adapting solutions, or parts of solutions, so they apply to a whole class of similar problems. • Transferring ideas and solutions from one problem area to another.	• Identifying patterns and commonalities in artefacts. • Remix and reuse (by extending if needed) resources previously created. • Adapting solutions, or parts of solutions, so they apply to a whole class of similar problems. • Transferring ideas and solutions from one problem area to another.
Evaluation	• Recognise when instructions do not correspond to actions. • Remove and fix errors. • Assessing that an artefact or solution is fit for purpose.	• Recognise when instructions do not correspond to actions. • Remove and fix errors. • Assessing that an artefact is fit for purpose. • Assessing whether the solution is effective and efficient. • Shows awareness of tasks best completed by humans or computers.	• Recognise when instructions do not correspond to actions. • Remove and fix errors. • Assessing whether the solution is effective and efficient. • Identifying ways to improve solutions or information quality. • Selecting and justifying appropriateness, precision, or quality of "best" solutions and information sources.

3.2 Incorporates active learning

When participating in development activities, teachers should be constructive participants rather than passive recipients of information (Darling-Hammond et al., 2017). Active learning in this framework should involve teachers when designing and experimenting with instructional methods, allowing them to use authentic artefacts, immersive experiences, and other techniques to provide profoundly rooted and contextualised professional learning (Desimone, 2009). PD activities should actively engage teachers in meaningful discussions with other teachers or training specialists about the goal of a lesson, tasks, teaching strategies, and practice. Teachers should apply the different CT practices, such as tinkering with existing artefacts to make changes or creating new ones by collaborating.

Active learner participation is critical to constructionism instructional design, and task-based learning strategies are an excellent approach to keep learners engaged throughout the course (Loi, 2004). According to the constructionist learning theory, which underpins this study, instructors serve as facilitators. The facilitator should encourage collaboration and adapt the lessons to the teachers' prior level of comprehension (Kurt, 2021). This aligns well with constructionism; to personalise learning experiences, constructionists believe each learner should be engaged on both cognitive and emotive levels (Brennan, 2015).

Involving teachers as design partners during the participatory design workshop (see section 3.5), they are recognised as legitimate decision-makers and given equal status with the researcher as stated in the literature by Iversen et al. (2017), Clement and den Besselaar (1993), and Kensing and Blomberg (1998).

The following sections (as defined in section 2.9.2) discuss the suggested teaching strategies for active learning that can be used in teaching teachers about CT skills.

3.2.1 A list of active learning strategies

Unplugged computing is one teaching strategy suggested to develop teachers' CT skills without computing devices. This strategy will enable teachers to engage in CT activities regardless of their school context in terms of computing infrastructure. It is also an easier method for teachers without a computing background. Without the distraction of computer devices, the unplugged method broadens the reach of CT across different perspectives (Huang & Looi, 2020).

Programming or coding is the most researched strategy in training teachers on CT skills, according to the literature (Ausiku & Matthee, 2021). Educating primary school teachers about CT principles using block-based programming tools, such as Scratch and hands-on programming activities will enable them to produce instructional materials for their classrooms. Computational pedagogy is an example of a constructionist method of structuring the educational environment, and it is especially useful for primary level instruction. With the use of modelling and simulation tools, such teaching helps students to "cycle back and forth between the inductive and deductive approaches to learning" (Dolgopolovas et al., 2019, p.185).

Robotics is the second most researched method for teaching CT skills to primary school teachers, according to the literature (Ausiku & Matthee, 2021). Training teachers to integrate educational robotics into their classrooms should increase their confidence and CT skills by constructing and programming these robotic kits such as LEGO. Learning to teach CT concepts through robotics is a constructivist endeavour. Therefore, teachers should know how to design learning environments to support learners.

Project-based learning (PBL) is another teaching strategy in which teachers' learning of CT skills is organised around tasks and requires teachers to engage in design, problem-solving, decision-making, or investigation activities in collaboration with others. As a form of situated learning (Lave & Wenger, 1991), PBL is founded on the constructivist conclusion that students get a deeper knowledge of content when they actively construct their understanding by working with and employing concepts in real-world contexts. Engaging teachers in a design-centred approach can also help teachers improve their CT content knowledge and pedagogical practices (Du & Igwe, 2018).

Game-based activities should enable teachers to extend common ideas from their subjects to various dynamic activities using CT skills to generate models and representations using games. In constructivist teaching approaches, technology and games (which can range from digital cameras to complex simulations) are used to assist students in completing projects that help them learn how to recognise and solve issues, grasp new phenomena, create mental models of these phenomena and how to set objectives and govern their learning. As indicated by Simonsen & Robertson (2012), Bødker and Kyng (2018) and Tuhkala (2019), by actively teaching teachers about CT skills through the process of participatory design, genuine engagement is achieved.

3.3 Coherence

Designing for teachers' development coherence implies striving to promote a culture of learning throughout PD events and encouraging teachers to understand their learning as connected to a common set of ideas—about schools, students, teaching, and learning. Coherence helps teachers to develop the same thoughts regarding their teaching practice over time and influences the transformation of instruction. Coherence also includes delivering PD in a mode consistent with the teachers' skills and beliefs (Desimone, 2009).

An important outcome of the PD4PCT framework is that teachers should be convinced of the topic's relevance, have improved CT and programming abilities, be prepared to integrate CT into their lessons, and have increased confidence to teach the topic. This is in line with Mason and Rich's (2019) findings that training programmes must include coding experience, self-efficacy, and teachers' other views about coding and CT to prepare teachers to teach programming and CT. Without understanding the value that CT adds to their classrooms regarding technology usage; teachers will be hesitant to teach it (Kale et al., 2018).

The framework builds on a constructionist approach to teacher learning, assuming that educational experiences are most effective when they build on and exploit what teachers already know, as Rich, Yadav and Schwarz (2019) suggested.

The table below shows teachers' skills and beliefs relating to CT as identified in the literature (see also section 2.10) that should be considered and determined before and after the PD workshops. These skills and beliefs will influence the coherence of the PD programme. For example, suppose the teachers' skills and knowledge is not considered, or the training is not aligned to their current subjects. In that case, the PD workshop may not improve their content knowledge of CT. Likewise, suppose the teachers' beliefs are unknown. In that case, it will be difficult to change their attitudes towards CT and improve their teaching practice to incorporate CT, which is the workshop's objective (Desimone, 2009). Table 3.2 below shows what to measure.

Table 3.2 Teachers' Skills and Beliefs relating to CT

	What to measure	References
Skills	*Content knowledge about CT*	Rich, Yadav, and Schwarz (2019), Fessakis and Prantsoudi (2019), Ling et al. (2017), Llyod and Chandra (2020), Sentence & Csizmadia (2017), Mouza et al. (2017), Garvin et al. (2019), Mouza et al. (2017)
	CT Integration skills	Rich, Yadav, and Schwarz (2019), Fessakis and Prantsoudi (2019), Ling et al., (2019) Bati and Yetişir (2021), Garvin et al. (2019), Mouza et al. (2017), Hickmott (2020)
Beliefs	*Attitudes towards CT*	Kong and Wong (2017), Fessakis & Prantsoudi (2019), Llyod and Chandra (2020), Rich, Larsen and Mason (2020), Bati and Yetisir (2021), Mouza et al. (2017), Çoban et al. (2020), Ling et al., (2018)
	Self-efficacy	Çoban et al., (2020), Kaya et al., (2019), Rich, Larsen and Mason (2020), Hickmott (2020), Bean et al., (2015), Korkmaz et al. (2017)

3.3.1 Measuring instruments of the CT skills and beliefs of teachers

These skills and beliefs can be measured using the questionnaires provided in Appendices D & E. These questionnaires are based on the work of Bean et al. (2015). It is understood that measuring the outcome and effect of the PD of teachers is contentious (Desimone, 2009). However, it is necessary to assess the quality and impact of PD because without an understanding of what is effective and why, it is difficult to design and execute PD for teachers that is effective and considers teachers' needs (Darling-Hammond et al., 2017). These instruments can be used in pre- and post-assessment to determine the effect of the PD development programme on the teachers' CT skills and beliefs.

Through participatory design methods, the constructionist approach involves teachers from the start in co-designing units (Kelter et al., 2021). This process gives teachers a rich environment where they can develop their CT skills and confidence by creating computationally rich learning environments appropriate for their local educational contexts.

3.4 Duration

According to research, intellectual and pedagogical reform requires PD programmes to be sufficiently lengthy, both in terms of the time period covered by the practice (e.g., one day or one semester) and the number of hours expended on the activity (Desimone, 2009).

In anticipation of limited time for teachers, the workshops can be blocked in after-school sessions, each about 2/3 hours long, 2/3 days per week. Workshops should span at least four weeks. Another option is to conduct the workshop on Saturdays, e.g., two Saturdays per month over a semester. Another alternative is to train the teachers during school holidays through intensive sessions, e.g., 5-days covering at least 4 hours per day. The ideal total hours of contact should be at least 20 hours. Training needs to be continuous, not just a once-off activity. It can be spread over an entire semester or school year. Teachers should be asked to choose from available dates to establish a sense of self-motivation and control.

Constructivist interventions take significantly more time but are worthwhile when done well. This agrees with Kenny and Wirth (2009) that although it takes a little longer, the time investment pays off since it fosters an inquiry-based learning environment and culture of learning in the classroom.

3.5 Collective participation

Desimone (2009) refers to collective participation as she mentions that the characteristic of collective participation can be achieved by including teachers from the same school, grade, or department. High-quality PD provides opportunities for teachers to exchange insights and collaborate on their learning. By collaborating, teachers will build communities with a positive impact on the culture and instruction of their entire grade level, department, classroom, and/or area (Desimone, 2009).

Desimone's framework is enhanced by suggesting an explicit method to implement collective participation, namely PD techniques (see a detailed discussion of Participatory Design in section 2.15). During the training workshops, teachers will be provided with opportunities to collaborate, e.g., in grade level and subject-specific groups, to build new lesson plans and activities or enhance existing ones to integrate CT skills. This should allow them to control the creative process and develop authentic artefacts they can use in their classrooms. Spinuzzi (2005) explained that participatory design enables teachers to redesign their working tools. The design process is in line with constructionism, predicated on the notion that the most beneficial educational experiences are those that involve the active construction of a variety of things; particularly those that are significant on a personal or social level, produced through interactions with other people such as an audience, collaborators, coaches. This encourages reflection on one's way of thinking (Brennan, 2015; Papert, 1980).

Throughout this collaboration period, facilitators/researchers would 'step down' into direct working connections with teachers. This structure enables groups to ask facilitators questions throughout preparing lesson plans and activities, acquiring just-in-time CT knowledge. Unlike traditional expert-to-novice PD approaches, the participatory design presents teachers as knowledgeable collaborators with the agency to share their experience with children and the classroom environment hence contributing to the learning of others. At the conclusion, teachers will share, discuss, and reflect on their lesson plans and activities (Agbo et al., 2021).

While some characterise co-design as having strong facilitation and well-defined responsibilities, different teams can build their approaches following the constructionist value of methodological and epistemological pluralism (Kelter et al., 2021). According to the constructionist perspective, sharing a constructed artefact within a community is as critical as its construction, as it impacts a learner's comprehension and generation of meaning (Kurt, 2021).

Besides lesson plans and activities, different teaching strategies should be shared and discussed among teachers. Teachers should choose the teaching strategies that fit in with their school or classroom context. A few aspects of PD are discussed below as applied to this context:

3.5.1 Roles of participants

- Teachers are design partners, are acknowledged as legitimate decision-makers, and are promoted with an equal role with designers and researchers (Iversen et al., 2017).
- The facilitator guides and provides scaffolding during the design process. The role is a neutral position that encourages and empowers the group to actively engage in the design process (Tuhkala, 2019).

3.5.2 Tools and techniques during the PD process

According to Brandt et al. (2012) (section 2.15.5), their framework's three dimensions were applied to determine the best tools and methods for this study. For this study, the first dimension of *form* is applied, and it is decided that participating teachers will create CT-infused lessons and activities. The second dimension of *purpose* is applied by immersing the teachers in the CT domain, and the third dimension of *context* enables us to choose face-to-face training at a school with small groups of teachers.

3.5.3 The suggested Participatory Design process

The design process for lesson plans follows the participatory design methodology, stages, tools, and techniques described in sections 2.14.3 - 2.14.5 of the literature review (Agbo et al., 2021; Brandt et al., 2012; Spinuzzi, 2005). The design methodology is an iterative construction of an artefact, and the design stages are plan and engage, discover, design, evaluate and feedback.

To better present these stages, the figure below shows the process flow and connects it to participatory design.

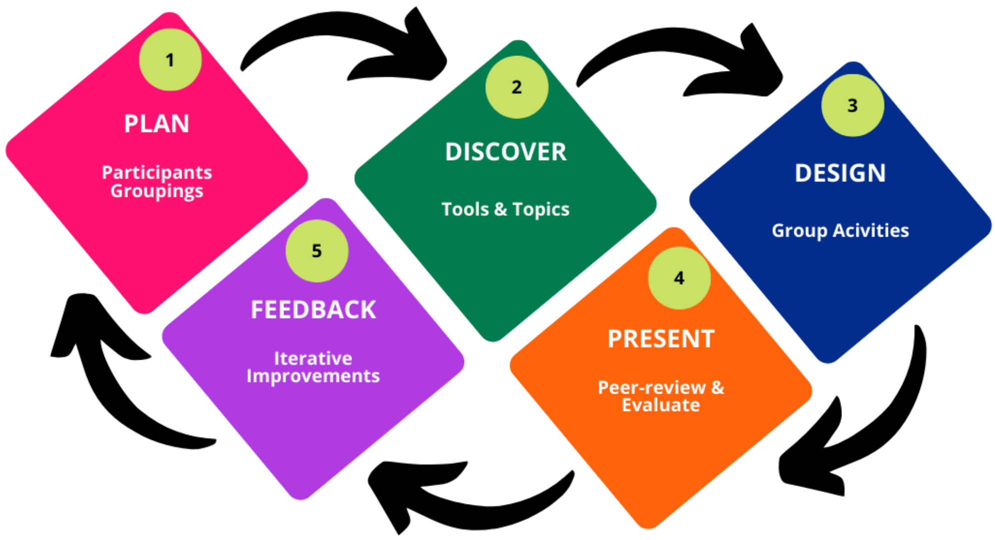


Figure 3.2 Participatory design process flow diagram (adapted from Agbo et al., 2021; Spinuzzi, 2005)

Planning stage

The first stage of planning and engaging is where the researcher and teachers interact most heavily, typically involving group interactions. Teachers can be grouped into small teams per subject or grade level at this stage. The participatory design positions teachers as knowledgeable collaborators with the agency to draw from their experience in a classroom environment with learners, hence contributing to the learning of others (Killen et al., 2020). Because of participatory design's orientation toward design, the goal is to make meaning out of work cooperatively rather than to simply describe it (Spinuzzi, 2005).

Agbo et al. (2021) suggest that to mitigate challenges such as Internet connectivity and motivated participants, it is recommended that during the planning and engaging stage, researchers should identify and recruit co-designers willing to participate. They also recommend that researchers provide basic facilities, such as an Internet connection that participants might need during the intervention.

Discovering stage

During the discovery stage, the topics and tools are assessed to ensure they fit for CT integration. The tasks are also distributed among team members, and all other materials are needed, such as lesson plan templates (Agbo et al., 2021). The existing lesson plan (Appendix H) is the work of Mason (2018), and the blank lesson plan template (Appendix I) is adapted from the works of Computing at School (2014), Mason (2018) and Hickmott and Prieto-Rodriquez (2018). The tools that can be provided at this stage will depend on the topics, CT concepts and teaching strategies chosen. For example, if a group that teaches Maths wants to focus on the topic of shapes, the CT concept of algorithms can be applied through programming. So, to do this, they will need a pen, paper, a computer with Scratch software, a lesson plan template and a reference sheet to Scratch code blocks. Before building a Scratch application to draw their designs, teachers can design algorithms to draw patterns formed of basic shapes in this exercise. They gain knowledge of the repetition concept this way. Choices of CT concepts to apply to subject topics can also be done based on the suggested curriculum (Table 3.1).

Designing stage

Experience shows that small groups can achieve quality collaboration, and inclusive collaboration can be enhanced if every member contributes to the group tasks (Agbo et al., 2021). Hence, it is recommended that researchers begin by conducting a brief seminar where participants are introduced to the activities' concepts, goals, and objectives. Afterwards, participants should be grouped to allow for effective collaboration (Spinuzzi, 2005). To make the designing stage a collaborative experience, the participants are grouped with each group limited to four members as guided by the context dimension of the Brandt et al. (2012) framework. Group activities within the designing stage include brainstorming on topics, ideation about contextual scenarios and stories, creation of lesson contents, and paper and mock-up designs of CT lesson plans.

Presenting stage

During the presenting stage, co-designers present their concepts at the group level. The groups act out the future lesson plans through enacting (Brandt et al., 2012). The peer review tries to obtain a user's perspective regarding what they consider suitable by playtesting the paper prototype. Therefore, we should allow co-designers to peer review themselves at the group level based on their expectations (Agbo et al., 2021). This way, they could learn more from one another's ideas and contribute by presenting their point of view.

Feedback stage

Each group evaluated another group's design and provided feedback as they presented their artefacts. In the feedback stage, comments from co-designers form part of the ongoing iteration of the design process. The process allows for feedback at any stage. Although the output from one stage could serve as the input of the next stage, it is recommended that the implementation of the process should be flexible enough to allow for scalability and changes that may arise (Agbo et al., 2021).

Summary of PD implementation

Table 3.3 below applies the Brandt et al. (2012) framework to help the training designer understand the importance of using all these different aspects during the workshops as these have different qualities, which help you get the right design by giving the participants many possibilities for idea generation, making of solutions and acting them out.

Table 3.3 Summary of the participatory design process

Stage	Steps	Form	Purpose	Context
Plan	Group teachers in small teams per subject or grade level. Have the workshop material guide that includes the timetable and tools needed (Appendix J).	Telling	Probing	Groups size of not more than four participants. Grouped per subject taught. Face-to-face mode. School as a venue.
Discover	Start by showing the teachers an existing example of a CT-infused lesson plan (see, for example, Appendix H). Explain what tools are available (see, for example, the list provided in Table 3.1). Provide teams with a blank template of the lesson plan (see, for example, Appendix I).	Telling	Priming participants to immerse them in CT.	
Design	Each team selects a unit topic, defines learning objectives, and brainstorms methods to 'CT-infuse' the topic. The teams develop an overview and unit plan and then develop the unit's contents.	Making	Get a better understanding of their current experience.	
Present	Teams present their lesson plans to the rest of the participants.	Enacting	Generation of ideas or design concepts for the future.	
Feedback	Teachers and the researcher discuss and critically examine the designed lesson plans and provide feedback. Teams incorporate the feedback and improve the lesson plans.	Telling & Enacting	Generation of ideas or design concepts for the future.	

3.6 Context

The effectiveness of the PD will depend on the school's context, including the school leadership and technological infrastructure.

3.6.1 Technological infrastructure

Teachers' pedagogical competencies in CT can also be increased if the necessary technology infrastructure is supplied and supported (Bower et al. 2017). It was also discovered that while adopting coding and CT classes, K-12 teachers face numerous challenges due to a lack of adequate instructional resources (Bower et al., 2017; Kadirhan et al., 2018; Ketelhut et al., 2019; Rich, Yadav & Schwarz, 2019).

CT integration can take various forms, depending on the needs of the particular teacher, the school, and the resources available. Integration may occur through a plugged activity (i.e., an activity that involves a computational device, such as a computer or robotics) or an unplugged activity (i.e., an activity completed using only paper and pencil, or other non-computational hands-on materials), depending on the precise learning goals you are pursuing (Sherwood et al., 2020).

Technological infrastructure needs for integrating CT into the classroom

Table 3.4 below shows the ideal technology infrastructure and activities that can be done using programming, robotics or game-based learning strategies (Esteve-Mon et al., 2019; Gleasman & Kim, 2020; Jaipal-jamani & Angeli, 2017; Leonard et al., 2016). It also shows how teachers can use the unplugged teaching strategy at schools where there are no computer labs or power connections (Brackmann et al., 2017; Rich, Yadav & Larimore, 2020).

Table 3.4 Technological infrastructure needs for integrating CT into the classroom (Brackmann et al., 2017; Jaipal-jamani & Angeli, 2017; Rich, Yadav & Larimore, 2020)

IT infrastructure	Computational thinking activities
• power connection • functional and easily available computers or tablets with the necessary software, e.g., Scratch • Internet connectivity • a school-based IT team • educational robotic kits, e.g., mBots, LEGO, etc.	• Automate algorithms and create programmes using Scratch and applying these fundamental programming concepts: ○ Sequencing ○ Repetition ○ Variables ○ Conditionals • Create interactive stories, animations and games. • Programme the robotic kits to perform tasks.

No computers nor Internet connectivity	• Create algorithms (step-by-step instructions on paper). • Playdough programming - one person (the programmer) instructs the other (the human computer) to create a playdough model based on verbal instructions alone. • Computational word games - converse with your partner without breaking the flow. • Paint by pixels - create your pixelated graphics using a spreadsheet or piece of squared paper.

3.6.2 School leadership

The school culture and leadership play a vital role in ensuring that CT integration happens in the classroom. School leadership is viewed among the stakeholder relationship through the dimension of context (Brandt et al., 2012). The school's leaders should be aware of and involved in the integration plan; otherwise, as in the study by Rich et al. (2017), a lack of cooperation from school management could impede CT integration. School administrators assist in articulating and communicating the school's vision and priorities, manage teaching and learning expectations, create and authorise instructional schedules that allow for CT integration efforts, and actively support their staff's PD (Boulden, 2020; Sherwood et al., 2020).

School strategies to enhance the integration of CT

Below are strategies that school leaders can use to increase the uptake of CT integration into instructional practice (Sherwood et al., 2020).

- Explicitly share a commitment to integrating CT within the school.
- Provide opportunities for ongoing professional development.
- Create a plan for access to resources and technological support.
- Begin small: While school-wide integration is the primary goal, it is a process that takes time and sustains effort, and integration need not happen all at once.
- Encourage teacher experimentation with integrating CT into instruction.
- Provide clear communication about how integrated CT will be approached in teacher evaluation.

- Amplify teacher and student successes both within the school building, with parents, and in the community.
- Develop mechanisms for promoting sustainability.

3.7 The PD4PCT framework

Figure 3.3 and Table 3.5 below provide the framework, including a graphical presentation of the components and the instruments developed for implementing the framework.

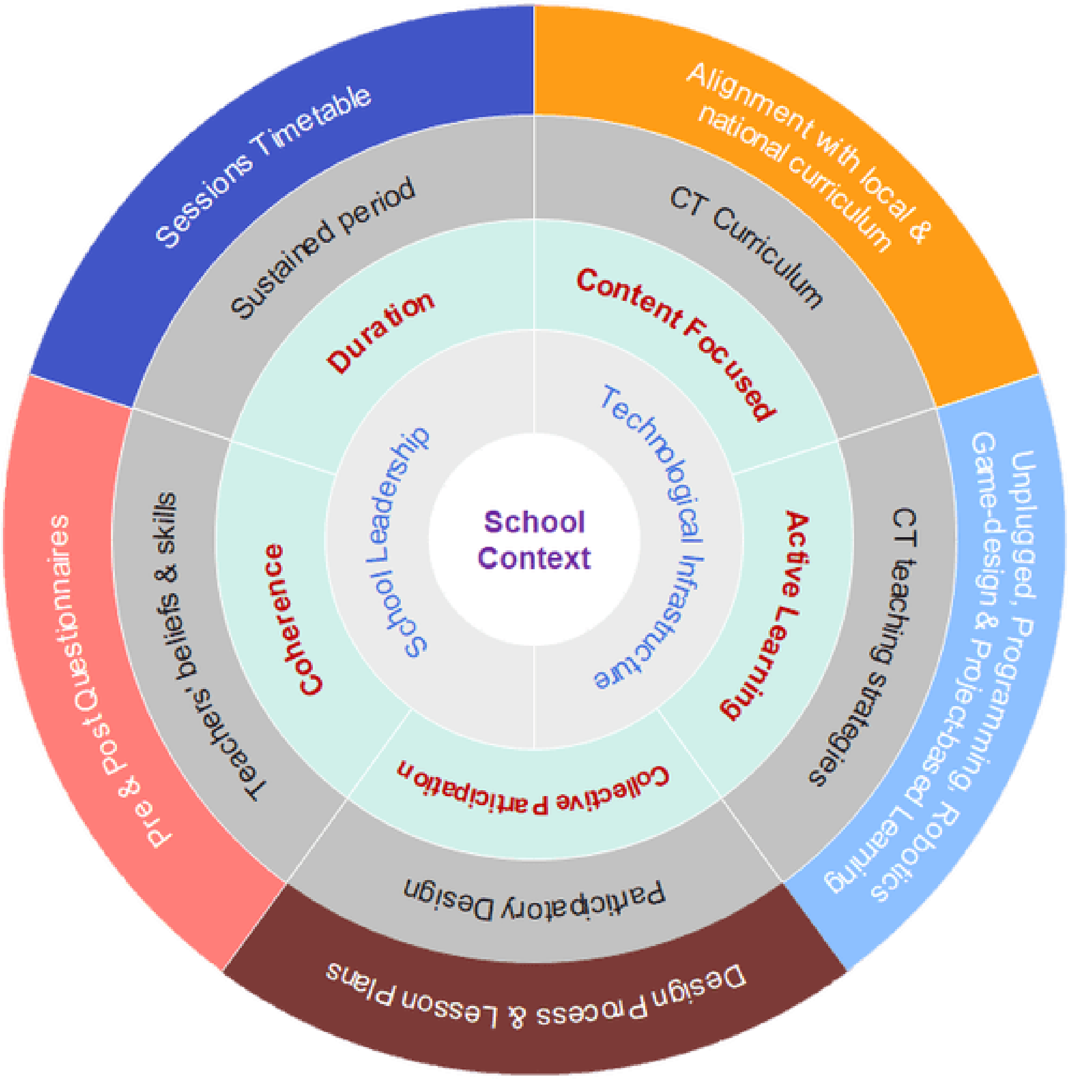


Figure 3.3 Initial Framework graphical presentation

Table 3.5 Components and the tools & techniques developed for the implementation of the framework.

Framework Component	***Tools & techniques developed for the implementation of the framework***
Content Focused	CT Curriculum (section 3.1).
Active Learning	List of active learning strategies (section 3.2.1).
Coherence	Pre & post-questionnaires to measure teachers' skills, beliefs, and attitudes (section 3.3.1).
Collective Participation	Participatory design process (section 3.5.3).
Duration	Workshop sessions timetable (section 3.4).
Context	Mapping between technological infrastructure and CT strategies (section 3.6.1.1), and strategies schools can follow to integrate CT (section 3.6.2.1).

Conclusion

This chapter discussed the proposed PD framework, which is the main object of this study. It highlighted the framework components and how training providers and researchers can use them to help teachers integrate CT into their classrooms. The next chapter looks at the methodology for this study.

The next chapter presents the methodology and research design applied to this study, including the research philosophy and approach, the study's strategies, methodological choices, and the data collection and analysis.

4 CHAPTER 4: METHODOLOGY

4.1 Introduction

This chapter discusses the adopted methodology for this study and is divided into sections. First, the research philosophy and approach are described. Next, the researcher explains the study's strategies, methodological choices and time horizons. After that sections describe the data collection, processing, and analysis to answer the research questions. The chapter finishes with a discussion of the study's validity and reliability.

In research, a methodology is viewed as a guiding premise. Under methodology, we cover the philosophical underpinnings of the research, the strategy, and the data gathering and analysis methodologies (Creswell et al., 2013). The chapter is guided by Saunders, Lewis and Thornhill's (2007) research 'onion' (see Figure 3.1).

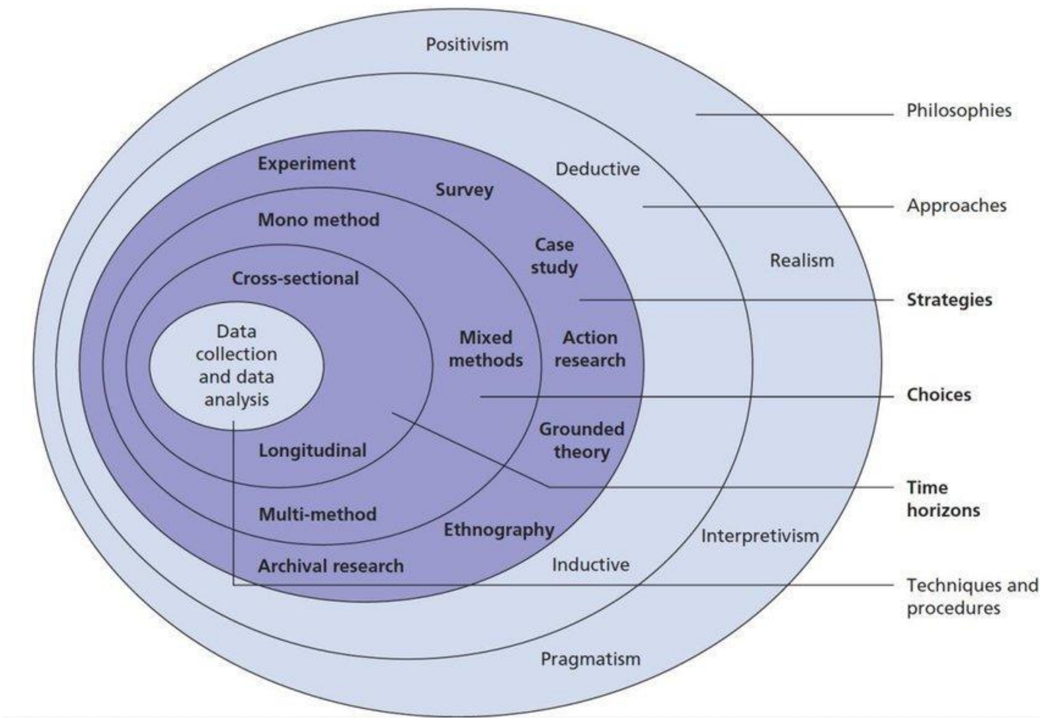


Figure 4.1 Research onion (Saunders et al., 2007)

4.2 Research Philosophy

The world-view assumptions (research philosophy) of the researcher are important as they will guide the method and strategies used for the study (Saunders et al., 2007). Easterby-Smith et al. (2008) claimed that if the researcher fails to think carefully about philosophical issues, then research quality can be greatly compromised as it is vital for research design.

The three main assumptions (*ontology, epistemology, and axiology)* about research philosophy are explained below.

Ontology concerns the nature of being or reality (Crotty, 1998). Ontology questions our assumptions of how the world works (Saunders et al., 2009). Ontology comes with two assumptions: realism and nominalism concerning the nature and structure of society. Realists assume that there are a pre-determined nature and structure to the outside world, and they are independent of human cognition and consciousness. Meanwhile, nominalists assume that reality differs for different people depending on their experiences and surroundings (Burrell & Morgan, 1979; Johnson & Duberley, 2000).

Epistemology is about what is considered to be valid knowledge (Collis & Hussey, 2009; Saunders et al., 2007).

Epistemology is a term used in research to describe how we come to know something, such as the truth or reality, or as Cooksey and McDonald (2011) put it, what counts as knowledge in the world. It is concerned with the foundations of knowledge - its nature, forms, and acquisition and how it might be transmitted to other humans. It focuses on the nature of human knowledge and comprehension that you, as a researcher or knower, may gain to extend, widen, and deepen your understanding in your field of study. It is defined by Schwandt (1997) as the study of knowledge and justification. When thinking about the epistemology of your research, you pose questions like: Is knowledge something that can be gained or something that must be directly experienced? What is the nature of knowledge, and how do the knower and the would-be known interact? What are the connection between me, the inquirer, and the known? These questions are crucial because they assist the researcher in situating oneself within the research context, allowing them to find what more is new, given what is already known (Kivunja & Kuyini, 2017).

Axiology deals with the value system of the researcher in terms of beliefs, personal values, feelings, etc. Axiology argues that reality can be value-free or value-laden (Bryman & Bell, 2003). When putting together a research proposal, axiology refers to the ethical considerations that must be examined. It entails identifying, analysing, and comprehending concepts of wrong and right behaviour concerning the research (Finnis, 2011). It evaluates how much value we will give to various parts of our research, such as participants, data, and the audience to whom we will present our findings. It seeks to answer the following question: What is the nature of ethics or ethical behaviour? In answering this question, think about how you respect the human values of everyone who will be involved in or participate in your research study (Kivunja & Kuyini, 2017).

Several research philosophies guide research with different assumptions regarding ontology, epistemology and axiology. Positivism, interpretivism, and pragmatism are some examples of research philosophies (Myers, 2013; Olivier, 2009).

4.2.1 Research paradigms

According to Guba et al. (1994), a paradigm is a basic set of beliefs or worldviews that leads to research action or investigation. The gurus of qualitative research define paradigms as "human constructions" that deal with fundamental principles or ultimates that indicate where the researcher is coming from to generate meaning from data (Denzin, 2000).

Positivism

The positivist paradigm, first proposed by a French philosopher, Auguste Comte, defines a viewpoint based on what is known in research procedures as the scientific investigation process. It is utilised in nature to look for cause-and-effect linkages. It attempts to explain observations in terms of facts or quantifiable entities (Fadhel, 2002). Researchers in this paradigm use deductive logic, hypothesis development, hypothesis testing, operational definitions and mathematical equations, computations, extrapolations, and expressions to arrive at results. Its goal is to give explanations and make predictions based on observable results (Kivunja & Kuyini, 2017). According to the positivist paradigm, there is only one true and objective reality, which can be measured, researched, and understood through methodologically rigorous studies whose outcomes are independent of the researcher. Positivist research attempts to provide testable and repeatable discoveries, which should eventually lead to theories that allow researchers to make predictions or gain insight into the objective reality (Mcbride et al., 2021). Positivist research is primarily deductive in practical attempts, relying on quantitative approaches to evaluate hypotheses and ideas. In other words, "positivists assume that reality is fixed, directly measurable, and knowable and that there is just one truth, one external reality" (Rubin & Rubin, 2011, p.14).

Interpretivism

In reaction to positivism, there are a variety of "anti-positivist" and "postmodern" philosophies, one of which is interpretivism. In contrast to positivism, interpretivism favours inductive rather than deductive research, sees reality as socially constructed, and focuses less on identifying an "objective" reality and more on understanding and reconstructing how the study's subjects arrived at their conclusions. Importantly, "interpretive approaches do not merely study beliefs, ideas or discourses, they study beliefs as they appear within, and even frame, actions, practices and institutions" (Bevir & Rhodes, 2003, as cited in Hay, 2011, p.17). This implies that an interpretivist researcher would believe that "to understand this world of meaning, one must interpret it. The inquirer must elucidate the process of meaning construction and clarify what and how meanings are embodied in the language and actions of social actors" (Schwandt, 1993, p. 222).

Thus, the interpretivist paradigm's central assumption is that reality is socially constructed (Bogdan & Biklen, 1998). This is why this paradigm is occasionally referred to as the *Constructivist paradigm* (Kivunja & Kuyini, 2017). In this paradigm, theory follows research rather than precedes it, as it is founded on the evidence acquired during the research act. When this paradigm is used, data are obtained and analysed in a way consistent with grounded theory (Strauss & Corbin, 1990).

This paradigm presupposes a subjectivist epistemology, relativist ontology, a naturalist methodology, and a balanced axiology. These components are discussed briefly below.

- ***Subjectivist epistemology*** implies that the researcher makes sense of their data by their reasoning and cognitive processing of data-informed by their interactions with participants. There is an assumption that the researcher would construct knowledge socially because of personal encounters in the natural settings under investigation (Punch, 2005). The researcher and their subjects are assumed to be engaged in collaborative procedures in which they interact, dialogue, question, listen, read, write, and record research data.

- **Relativist *ontology*** implies that you believe that the situation under study has various realities, which can be examined and given meaning through human interactions between the researcher and the research subjects and among the research participants (Chalmers et al., 2009).
- The researcher employs a ***naturalist methodology***, gathering data through interviews, discourses, text messages, and reflection sessions, serving as a participant observer (Carr & Kemmis, 1986).
- A ***balanced axiology*** presupposes that the research outcome will represent the researcher's values, attempting to offer a balanced account of the findings.

Pragmatism

The concept of pragmatism lies somewhere between positivism and interpretivism, but it is closer to the latter. For pragmatists, "a theory for a pragmatist is true if and only if it is useful. Pragmatists are not looking for the essential and timeless truths of the positivists and logical empiricists" (Marshall et al., 2005). Pragmatism can be traced back to philosophers like Charles S. Peirce and John Dewey, according to Wicks and Freeman (1998); it is empirical in nature but "goes beyond a pure orientation to the observation of a given reality" and is instead focused "towards a prospective, not yet realized world" (Goldkuhl, 2004. p.13). Pragmatism differs from other philosophies because it emphasises the importance of action over theory, knowledge, and understanding, as these can only be obtained by action. Ontologically, pragmatism concerns "actions and change; humans acting in a world that is constantly changing" (Goldkuhl, 2012. p. 139). In this sense, pragmatism spans the ontological divide between positivism and interpretivism. Pragmatics typically shows up in applying mixed methods research when it comes to technique. Regarding epistemology, pragmatism is most concerned with knowledge that enables it to effect change, and it believes this knowledge can be achieved only by action. Pragmatic research is methodologically oriented on doing and would lend itself to methods that stimulate change and action, such as action research, design science, systems architecture and systems dynamics, and other methodological approaches specifically focused on enacting change.

Table 4.1 highlights the fundamental qualities and makeup of each research philosophy to explain their distinctions more clearly.

Table 4.1 Comparison of Positivism, Interpretivism, and Pragmatism

	Ontology	**Epistemology**	**Methodology**
Positivism	Objective reality	Knowledge is real and objective, obtainable via measurement and statistics (reductionism).	Surveys, experiments, statistical analysis.
Interpretivism	Subjective reality	Knowledge depends on beliefs, values, and lived experience (constructivism).	Field studies, case studies, hermeneutics, phenomenology.
Pragmatism	Objective/ subjective	Knowledge is obtained by doing and acting.	Mixed methods research, action research, design science.

Adapted from Mcbride et al. (2021).

4.2.2 Research paradigm choice – Interpretivism

Although qualitative research is frequently connected with interpretivism, qualitative research in information systems can also be conducted in accordance with a pragmatist paradigm since it is associated with action, intervention, and constructive knowledge (Goldkuhl, 2012). The two paradigms share an orientation towards understanding, but there is an important difference: In interpretivism, understanding is seen as a value of its own; in pragmatism it is seen as instrumental in relation to the change of existence (Dewey, 1931).

The current research aims to generate knowledge by constructing knowledge through participation (the intervention); thus, the empirical work aims to better understand how primary school teachers can be supported when transforming teaching practices and which contextual factors facilitate or obstruct this process. The interpretivism paradigm informs this research study, as it employs meaning-oriented (rather than measurement-oriented) approaches, such as interviewing or participant observation, that rely on a subjective relationship between the researcher and subjects. The interpretive approach does not place the researcher above or outside the activities but positions the researcher as a participant observer who participates in the activities and deduces the meanings of actions as they are expressed, in particular social contexts (Carr & Kemmis, 1986).

In terms of this research, empowering teachers to teach CT is without a pre-determined nature, and a structure and the truth can be different based on the researcher's time, place and point of view. Thus, between the two ontological assumptions, the current research is geared more towards nominalism.

The main objective of this research is to develop a PD framework that can guide primary school teachers in teaching CT accomplished by enhancing the capabilities of teachers. In this study, the researcher influences the collection and analysis of data because there is no external reality and objective truth (Crotty, 1998). Interpretivism is deemed a suitable way of probing into the research question. As far as this research is concerned, it is value-laden as all aspects of the methods used; research question formulation would be imposed by values held by the researcher (Bryman & Bell, 2003).

Role of the researcher

In terms of training teachers to enable them to teach CT, the researcher is not just an observer, but part of the training as a facilitator. Real people's experiences determine a participant of the research and the reality, and this is a fundamental part of social constructionism (Easterby-Smith et al., 2008). Before this study, the researcher worked in Namibia's education sector as an education officer (advisory teacher) for ICT under the PD division. The mandate was to train and advise teachers on ICT-related subjects. The researcher was privileged to have attended a Training Workshop on Computer Science Education at the International Science, Technology, and Innovation Centre for South-South Cooperation, where she was introduced to CT in education. Given the researcher's background in CS, she was interested in exploring the integration of CT into compulsory education, which led to this study. The researcher intended to develop a framework that can be applied to the PD of teachers for teaching CT in developing countries.

Table 4.2 below summarises the research philosophy undertaken in this study.

Table 4.2 Summary of the research philosophy undertaken

Philosophy Assumption	**Interpretivism**
Ontology	*Socially constructed*
Epistemology	*Subjectivism*
Data collection methods	*Literature review* *Pre-post-questionnaires (Likert scale)* *Semi-structured interviews* *Journals*

4.3 The research Approach followed in the study

Using a theory is part of the research approach, and Saunders et al. (2007) examined abduction, deduction, and induction as three distinct research approaches. An inductive research approach enables the researcher to understand the reality of the research context (Saunders et al., 2007). The three approaches of abduction, deduction, and induction differ in logic in that abduction flows between data and theory, whereas deduction flows from theory to data, and induction flows from data to theory or conceptual framework (Saunders et al., 2007). "Data collecting is utilised inductively to investigate a phenomenon, find themes and patterns, and develop a conceptual framework" (Saunders et al., 2007, p.144). This research used all three approaches to solve the research problem, which aims to understand the experiences of primary school teachers in teaching CT using the theory of constructionism as a lens. The abductive approach was used to suggest a framework, after which a deductive approach was used to gather data (since the framework guides it). Inductive reasoning was used to identify themes from the interviews.

4.4 Research Strategy followed in this study

Research has several strategies that can be used, such as case studies, grounded theory, surveys, and action research, to name but a few (Myers, 2013). Not all strategies are suitable for any research. For example, experiments look at relationships between two or more variables by trying to observe what manipulation of independent variables has on dependent variables. This does not apply to this study (Saunders et al., 2007). A case study strategy was selected for this study because of its appropriateness and is explored in the next section.

This research followed a two-phased approach: a case study to implement and evaluate the framework and then a survey to validate it.

4.5 Phase 1: Implementation and evaluation of the framework

4.5.1 The case study strategy

A case study is an empirical investigation that examines a contemporary phenomenon in its real-world setting, particularly when the distinction between phenomenon and environment is not readily apparent (Yin, 2009). One significant strength of case studies is their ability to investigate a phenomenon in its context; hence, case studies are a helpful method for examining the world (Rowley, 2002). Due to the nature of this research, which necessitates a thorough grasp of the context in which it is conducted, and the processes being enacted, as indicated by Saunders et al. (2009), the case study technique was the best strategy for this research.

According to Walsham (1995, p.14), "case studies provide the main vehicle for research in the interpretive tradition." The case study technique has been claimed to be especially advantageous for practice-based problems where the actors' experience, and context of action are relevant (Galliers, 1991). Montealegre (1999) asserts that case studies (particularly in-depth case studies) enable a complete examination of complicated historical and social events. By investigating exposure to the phenomenon of interest, the interpretive researcher strives to draw their concepts from the field. This strategy results in the emergence of categories and themes preferably closely related to the experiences of the relevant study's participants (Orlikowski & Baroudi, 1991).

Yin (1994) cautions against conflating case studies with qualitative ethnographic approaches. Yin (1994) contrasts ethnographies with case studies by stating that the former requires extensive observational evidence and takes a long time to undertake. Case studies are undertaken over a specific period and do not always necessitate employing ethnographic approaches. Researchers conducting case studies may not even need to visit the organisation being studied; they can collect data by reviewing secondary sources or conducting telephone or e-mail interviews with respondents. Yin (1994, p. 13) believes that "the case study method enables an investigation to maintain the holistic and relevant qualities of real-world events such as individual life cycles, organisational and managerial processes, neighbourhood transformation, international relations, and industry maturation."

Thus, the case study approach is advantageous in cases where the contextual circumstances around the events being investigated are essential, and the researcher has little control over how the events occur. As a research strategy, the case study should incorporate specific methodologies for data collection and analysis guided by clearly stated theoretical assumptions. Additionally, data should be gathered from various sources, and its integrity should be maintained (Yin, 1994).

Additionally, Yin (1994) identifies three types of case studies: exploratory, causal, and descriptive. In an exploratory case study, data gathering happens before formulating hypotheses or particular research questions; this is followed by data analysis, which results in more systemic case studies. The first step in this case study is defining the research topics. The causal case study will investigate cause-and-effect linkages and explanation ideas for the observed phenomena. According to Yin (ibid.), this situation creates the optimal conditions for using the case study as the primary research technique. A descriptive case study will require a theory to guide the data gathering process. This theory should be articulated openly in advance, open to discussion and debate, and eventually serve as the descriptive case study's plan (Yin, 1994).

Walsham (1995) believes that the legitimacy of a case study approach developed from an interpretive epistemological position is predicated on the plausibility and cogency of the logical reasoning used to describe and convey the results of the instances and to draw inferences from them. Similarly, Yin (1994) believes that case studies are employed for analytical generalisations, in which the researcher seeks to generalise a specific set of findings to broader theoretical assumptions. Walsham (1995) expanded on Yin's approach with four categories of generalisation derived from interpretative case studies: concept development, theory creation, theory inference, and contribution of deep insight. Additionally, the case study method enables 'rich descriptions' of the studied phenomena (Yin, 1994).

4.5.2 Motivation for Case Study

The PD4PCT framework developed in Chapter 3 will be implemented through a case study and evaluated. The following aspects will be evaluated: *Beliefs, Self-efficacy and CT knowledge.*

Given the interpretive position taken in this research and the nature of the research questions, the case study methodology was chosen because it enables the systematic collection of data, analysis of information, and reporting of results, allowing for a thorough understanding of a particular problem or situation (Merriam, 1998).

According to Creswell (2003), a topic can be studied as a single or multiple case study within a bounded structure. Case study methodologies enable the researcher to examine the phenomenon under investigation in its natural setting, bounded by space and time (Hancock & Algozzine, 2006). In this study, the case study is contained within the Khomas Region of Namibia. When a researcher is attempting to answer a "how" or "why" research question, a case study is the most appropriate method of inquiry (Yin, 2009). The fundamental research question for this study is: How should in-service teachers be prepared to teach CT in primary schools in a developing country context? Thus, the case study method is appropriate for this research since it enables the collection and analysis of data from various sources, including interviews, surveys, documents, and observations, enhancing the findings' credibility (Yin, 2009).

The study's main objective is to develop a framework and evaluate it in a single case deemed sufficient.

4.5.3 Unit of analysis

The term 'unit of analysis' can be simply defined as "the entity that is being analysed in scientific research" (Dolma, 2010). The unit of analysis is the major entity you are analysing in your study. For this research, a school will be a case, and the study is concerned with the support for teachers in integrating CT into the curricula. Here, the unit of analysis is teachers. The research is intended to conduct a single case study.

4.5.4 The Case Study

The identified case study school, FCR Primary School (pseudonym), is in Windhoek, Khomas Region, Namibia. The school has about 1200 learners from Grades 0-7, 40 teachers, a principal, four heads of department, three institutional workers and two secretaries. The school is in an informal settlement (a place in an urban setting used for residential purposes without formal planning approval) and does not have a computer lab for learners yet; however, the construction of the computer lab is at an advanced stage. The school has five desktop computers and three laptops for the teaching staff. The school principal has a positive attitude towards computing education and was keen for her school to be part of the study. The principal was instrumental in fundraising for constructing the computer lab. CT is not yet part of the curriculum in Namibia. However, it is included in the draft revised policy for ICT in education passed soon by the cabinet (Ministry of Education, 2019).

The researcher chose FCR Primary School due to the school's proximity to her workplace, thus giving her convenient access to the school, saving her time and travelling expenses, which gave her enough time to do in-depth research. The case study school is shown in Figure 4.2 below.

Figure 4.2 Case study school

4.5.5 Participatory design teacher groups

In this study, participatory design was used as a methodological approach for the intervention, which was the professional development programme. For the participatory design workshop, teachers were divided into five groups based on the subject taught. This resulted in a group for Social Studies, Natural Science, Mathematics, English and Afrikaans language. Some groups consisted of fewer teachers than others. Each group had to decide on the topics to work on. The teachers collaborated in creating and reviewing lesson plans and activities, with the researcher acting as a facilitator. Table 4.3 below show how the teachers were grouped according to the subjects they teach while Figure 4.3 show teachers working in a group.

Table 4.3 Participatory groups

Group	Subject	Participants
Group 1	Social Studies	Teacher A, Teacher E, Teacher M
Group 2	English	Teacher B, Teacher G, Teacher L
Group 3	Natural Science	Teacher C, Teacher D, Teacher I, Teacher J

Group 4	Mathematics	Teacher K, Teacher N
Group 5	Afrikaans	Teacher F, Teacher H

Figure 4.3 Teachers working in groups

4.6 Methodological choices for this study

This section provides an overview of the methodology used in the study. The research method is an investigation strategy that progresses from fundamental assumptions to research design and data collection (Myers, 2009). While there are further distinctions across research modes, the most prevalent categorisation of research methodologies is qualitative and quantitative. On one level, qualitative and quantitative allude to distinctions concerning the nature of knowledge: how one perceives the world and the research's ultimate goal. On a more fundamental level of discourse, the phrases refer to research methodologies, specifically how data are acquired and analysed and the types of generalisations and representations formed from the data (Myers, 2009).

Quantitative research methods originated in the natural sciences to study natural events. Qualitative research methodologies were established in the social sciences to enable researchers to explore social and cultural phenomena. In education, both quantitative and qualitative research is undertaken. Neither approach is intrinsically superior to the other; their usefulness is determined by the context, goal, and nature of this research study, depending on the type of study, one may be preferable to the other. Depending on the type of study and its methodological base, some researchers prefer to utilise a mixed methods approach, capitalising on the distinctions between quantitative and qualitative approaches and combining them in a single research project (Brysman & Burgess, 1999).

Although the phrases mixed methods, and multimethod have been used interchangeably in the social and behavioural sciences, including IS, the two have important conceptual distinctions (Venkatesh et al., 2013). Researchers in multimethod research use two or more research methodologies but may (or may not) limit the research to a single perspective (Mingers & Brocklesby, 1997; Tashakkori & Teddlie, 2003; Teddlie & Tashakkori, 2009). For example, a researcher may analyse a new IS deployment in an organisation using participant observation and oral history. Another researcher may employ ethnography and case studies to better comprehend the same problem. In both circumstances, the researchers are limited to a single viewpoint (i.e., qualitative) yet use different data gathering and analysis approaches. Mingers and Brocklesby (1997) distinguished two types of multiple methods research: methodology combination (combining two or more methodologies, e.g., survey and interviews in a research inquiry) and multi-methodology (partitioning methodologies and combining parts (e.g., two methodologies within qualitative paradigms). They proposed that multi-methodology research might utilise a single paradigm or numerous paradigms.

This study is an interpretive case study where both quantitative and qualitative data were collected. The viewpoint remains qualitative but uses quantitative data to enhance the interpretation. More information on the data collection methods is given in section 4.8.

4.7 Time Horizon of this study

Time horizon is about how long the study will take, and this can be determined through two perspectives longitudinal, where the phenomenon is investigated over a longer period or cross-sectional, where the researcher carries out the study over a particular shorter period (Collis & Hussey, 2009; Saunders et al., 2007). Due to the time constraints associated with academic studies, most research studies are cross-sectional; however, those that are not time-constrained may take a longitudinal approach (Saunders et al., 2007). This study was conducted over 15 sessions in four weeks and is considered a cross-sectional study.

After establishing the study's time horizon, the next layer describes the techniques and procedures used in the research. The following section discusses data collection and analysis.

4.8 Techniques and Procedures used in this study

This section discusses the techniques and procedures used to carry out the research. The following sub-sections explain the target population and then data collection and analysis.

4.8.1 Population

A population is a group of possible people or units that can be part of the study (Denscombe, 2010). Castillo (2009) regards a distinct collection of individuals with related or parallel characteristics as the research population. For this study, the population is primary school teachers in Namibia.

4.8.2 Sample

Since case study research was conducted, one school was chosen to understand the value of the developed framework in a specific context. Within the case study, a sample was chosen from the senior primary phase, Grades 4-7 and science, social studies, and language departments. Eighteen teachers (18) from the three departments made up the sample of this study. However, due to last-minute cancellations, two teachers pulled out.

The researcher was aided in identifying potential participants for the study by the school principal and heads of departments. To prospective participants, a document outlining the study's purpose was distributed. Self-selecting sampling led to willing and eager participants. This made it so much easier and possible to conduct the study successfully. The participating teachers are shown in table 4.4 below.

Table 4.4 Participating Teachers

Participant	Grades Taught	Subject
Teacher A	6-7	Social Studies
Teacher B	4	Social Studies
Teacher C	5	Natural Science
Teacher D	6-7	Natural Science
Teacher E	5, 7	Social studies
Teacher F	4, 6	Afrikaans
Teacher G	6	English
Teacher H	5-7	Afrikaans
Teacher I	6	Natural Science
Teacher J	6-7	Natural Science
Teacher K	4	Mathematics
Teacher L	4	English
Teacher M	4-6	Social Studies
Teacher N	4-6	Mathematics

4.9 Data Collection methods used in this study

Teddlie and Tashakkori (2003) state that a method of data collection is simply a technique that a researcher employs to collect first-hand research data. McMillan (2012) mentions that quantitative data collection methods are used to measure, document, and provide numerical values, while qualitative methods of data collection are relatively unstructured and often open-ended through interviews.

In this study, the researcher mainly used qualitative methods to collect primary data through self-reporting journals and semi-structured interviews but also used quantitative methods to collect some data through pre-post-questionnaires. Secondary data was collected through a literature review. A systematic literature review was done preceding this study and published in the Conference Proceedings of the 5th International Symposium on Emerging Technologies for Education, 2020 (Ausiku & Matthee, 2021). In concurrent data collection, the quantitative and qualitative data are collected at roughly the same time (Teddlie & Tashakkori, 2003).

4.9.1 Pre - Post-Questionnaires

A questionnaire is a list of printed questions given or posted to participants who will complete the questionnaire in their own time (Laws et al., 2003). The researcher used a questionnaire in the study because it is a flexible instrument. By using a questionnaire, I organised closed-ended questions. Closed-ended questions allowed the respondents to select an answer from among a list provided by the researcher. This provided uniformity of responses that can be processed to obtain the quantitative data (Babbie, 2013).

The next sub-section explains the design of the instrument (questionnaires) used pre-workshop and post-workshop to collect quantitative data.

Instrument Development

To answer SRQ 1: What is the change in teachers' beliefs and attitudes towards CT resulting from a professional development program in the Namibian context? the researcher adapted the TSECT instrument developed by Bean et al. (2015) to measure the participating teachers' perceived CT knowledge, beliefs, and self-efficacy towards teaching CT. Bean et al. (2015) developed the TSECT instrument by drawing on Bandura’s theories of self-efficacy and his suggestions for constructing instruments that measure self-efficacy. The TSECT items are also aligned with the dimensions and elements from the CT frameworks developed by Brennan and Resnick (2012), and Angeli, Voogt, et al. (2016) and the definitions of the five computational concepts were used in the construction of the items (see Appendix D & E).

Item Development

Questionnaire items were adapted from existing self-efficacy instruments and CT scales found in the literature (Bean et al., 2015; Korkmaz et al., 2017; Rich et al., 2020; Weese & Feldhausen, 2017). Following a review of the literature, I developed positive and negative statements to which teachers could answer using a five-point Likert-type scale to indicate their level of agreement with the stated opinion. The options were: Strongly Agree (SA), Agree (A), Neither Agree nor Disagree (N), Disagree (D), and Strongly Disagree (SD).

Both pre and post-questionnaires had similar items; the demographics section and the CT scales; however, the post questionnaire has extra items. The demographics section of the questionnaires helped to collect demographic information about the teachers in the programme: *grade taught, subjects taught* and *years of teaching experience.* The CT scales section had three sub-sections with the following constructs: *CT knowledge, Value Beliefs towards CT, Self-efficacy for CT* and *Teaching Self-Efficacy for CT (TSECT).*

Table 4.5 below shows the number of items per section for the pre- and post-questionnaires.

Table 4.5 Number of Questionnaire Items

Section	Pre-workshop Items	Post-workshop Items
Demographics	3	3
CT Knowledge	15	15
Value Beliefs towards CT	9	9
Self-efficacy for CT	10	22
Teaching Self-Efficacy for CT	0	10

i. CT knowledge items

CT is characterised in my study using the framework proposed by Angeli et al. (2016). Angeli et al. (2016) highlighted five CT skills described in the preceding chapter: *abstraction, pattern recognition, decomposition, algorithmic thinking, and debugging*. The study focused on all five ideas and three coding concepts from Brennan and Resnick's (2012) framework that was most relevant to the teachers that participated in my study: *sequences*, *loops*, and *conditionals*.

ii. Value beliefs items

As illustrated by Pajares (1992), value belief questions relate to the importance and utility value of learning CT and coding for young children. In that study, the *Expectancy x Value* hypothesis investigated learners' expectations for success and the utility of the significance, usefulness, and enjoyment of executing a certain job. Because expectancies are related to self-efficacy (which would require a distinct set of answers), this study concentrated on value-related questions regarding statements about the significance and utility of computing education.

iii. Computational thinking self-efficacy items

With numerous definitions of CT, the study sought to look for elements shared by all of them. The study also wanted to pay tribute to Wing's original proposal of CT as a basic 21st-century literacy comprised decomposition, abstraction, and automation. I tried to see which concept measured in the review of existing scales appeared in these frameworks. Finally, I settled on six constructs that comprise CT and must be reflected in the scale: decomposition, pattern recognition, algorithms, abstraction, problem-solving, and evaluation. Because most programmes aim to build CT by teaching some coding or programming, I also included many items designed to assess teachers' self-efficacy for foundational coding topics.

iv. Teaching self-efficacy for computational thinking items

Teaching efficacy is multifaceted. Teachers' beliefs about their competence to teach a subject to differ from their beliefs about their ability to do that subject well. To better understand the PCK required for teachers to teach CT, coding, or programming, I drew on recent research on teachers’ practices in teaching coding, CT, or programming in Grades K–8 (Rich, Browning et al., 2019). In that study, teachers of young children (ages 5–14 years) willingly answered various survey questions regarding their current coding teaching strategies, including their resources, accomplishments, obstacles, and advice for other teachers. I derived a set of tasks requiring coding teachers to be comfortable teaching CT or coding to young children.

Pre-workshop Questionnaire Items

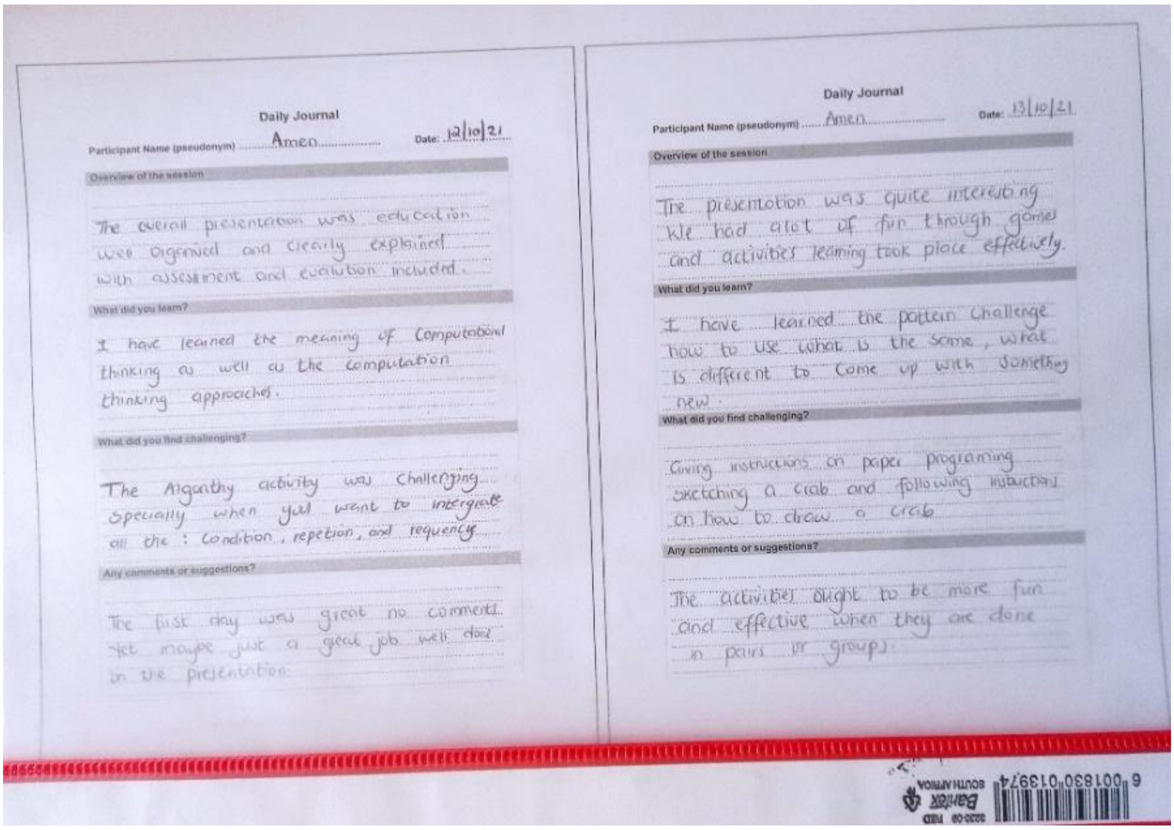
A subset of the items discussed above was used in the pre-workshop questionnaire, which was divided into two parts and included 34 items. I conducted a pilot study ahead of the programmes' initial session and appropriately projected that it would take between 20 and 40 minutes for participating teachers to complete the questionnaires, as they would not be as familiar with the concepts. The first sub-section of the CT scales was used to assess teachers’ prior knowledge of CT concepts. The second sub-section assessed teachers’ value beliefs of CT, while the third sub-section of the pre-workshop questionnaire assessed teachers' self-efficacy of CT before the workshop. Refer to Appendix D for the questionnaire.

Post-workshop Questionnaire Items

The post-workshop questionnaire had 57 items and is included in the appendices. It included the same sections as the pre-workshop questionnaire, and the survey items were identical to those used in the demographics, CT knowledge, and value beliefs sections of the pre-workshop questionnaire. The section on self-efficacy for CT had eight extra items connected with programming principles and four additional items aligned with CT practices. A new construct titled “Teaching Self-Efficacy for Computational Thinking” (TSECT) has been included. Refer to Appendix E for the questionnaire.

4.9.2 Weekly Journals

It was critical to ask teachers about their learning during the workshop’s four weeks to comprehensively understand their learning during the programme. The gathering of journal entries done by teachers, as used by Reding and Dorn (2017), is one way to collect data about teachers' experiences during PD programmes. Collecting weekly journal entries provided a way to collect qualitative data from teachers on a regular and timely basis throughout the programme. Teachers were instructed to write a journal entry during each workshop session’s last 5 to 10 minutes. The weekly journal entries were gathered via a booklet, which is included in Appendix G. The questions in the weekly journals were designed to elicit replies from teachers about their experiences during each session, what they had learnt in each session, and which ideas from each session, they were having difficulties understanding. The weekly journal entries as shown in Figure 4.4 were also utilised to gather input from teachers on how the sessions were run. I examined the journal entries of the participating teachers after each of the weekly sessions and altered how I ran sessions or explained particular concepts the next week where I thought it was appropriate. Thematic analysis of these journal entries was used to aid in answering the research questions.


Daily Journal

Participant Name (pseudonym) Amen Date: 12/10/21

Overview of the session

The overall presentation was education well organised and clearly explained with assessment and evaluation included.

What did you learn?

I have learned the meaning of Computational thinking as well as the computation thinking approaches.

What did you find challenging?

The Algorithm activity was challenging specially when you want to intergrate all the: condition, repetion, and frequency

Any comments or suggestions?

The first day was great no comments yet maybe just a great job well done on the presentation

Daily Journal

Participant Name (pseudonym) Amen Date: 13/10/21

Overview of the session

The presentation was quite interesting. We had alot of fun through games and activities learning took place effectively.

What did you learn?

I have learned the pattern challenge how to use what is the same, what is different to come up with something new.

What did you find challenging?

Giving instructions on paper programing sketching a crab and following instuctions on how to draw a crab

Any comments or suggestions?

The activities ought to be more fun and effective when they are done in pairs or groups.

Figure 4.4 Example of a journal used

4.9.3 Semi-structured Interviews

Glesne (2011) states that an interview is face-to-face communication between at least two persons to collect data on a specific topic. Interviews are required for case study research (Myers, 2013). Interviews allow the researcher to acquire rich, insightful data detailed enough to reveal the participants' "experiences or behaviours, opinions or beliefs, feelings, and factual knowledge" (Esterberg, 2002, p.45).

Qualitative data was gathered through semi-structured interviews to explain and understand the context of teachers' experiences relating to CT during the training workshop (Hennink et al., 2011).

Interviews were guided by pre-prepared questions as included in the appendices (Appendix F), and these prepared questions are meant to serve as a reference for extra questions or difficulties that may arise throughout the interview process (Creswell, 2013; Myers, 2013; Oates, 2005). They allowed the researcher to ask to follow-up questions as needed. Interviews were conducted a day after the workshop ended in groups at the school. Due to time constraints, there were two individual interviews for the teachers who could not fit their schedules into group slots. All interviews were voice-recorded and transcribed verbatim. The qualitative data collected were analysed to answer RQ3, RQ4 and RQ5.

4.9.4 Literature review

The literature review protects the researcher from reinventing the wheel (Hall & Hall, 2008). However, it also allows the researcher to profit from theoretical and intellectual advancements in the field in which the study will be undertaken. A literature review examines information and research-based material to provide a full, accurate portrayal of the available knowledge and research-based theories on a certain issue (Dawidowicz, 2010).

The literature review assesses what is known about CT at the primary level, ensuring that knowledge of key issues in the research is not assumed. The literature review is an important element of the research process since it includes the evaluation and examination of relevant research papers and other documents and policies. It also gives the researcher a thorough understanding of past research work (Hartas, 2010).

According to Rocco et al., (2011), a critical literature review kicks off the theoretical and methodological foundations of research. It serves as a basis for knowledge construction while elucidating the topics under consideration. A fresh viewpoint on the issue is also generated, providing greater investigative and predictive potential. Literature reviews can help with knowledge of the subject being examined (Mertens & McLaughlin, 2004). When developing a volume of recognised information on any education issue, challenges may arise if necessary, and literature evaluations are not conducted. A review of the literature can also give further insights into the goal and findings from research. Some research questions, such as RQ2 and RQ3, will be answered with the help of the literature review.

Table 4.6 below summarises the type of instruments and methods used to collect data in this study.

Table 4.6 Summary of instruments used

Instrument	Method
Pre - post-workshop questionnaires	Quantitative
Self-reporting journals	Qualitative
Semi-structured interviews	Qualitative
Literature review	Qualitative
Online questionnaire	Qualitative

4.10 Data Analysis techniques used in this study

“Data analysis consists of examining, categorising, tabulating, testing or otherwise recombining evidence to produce empirically based findings” (Yin, 2004, p. 132).

Out of 16 participants, one teacher did not complete the pre-questionnaire, and another one did not complete the post questionnaire. These participants’ data were excluded from the analysis as their data was incomplete.

4.10.1 Quantitative Analysis

This section describes the data processing and analysis utilised for the quantitative aspects of the mixed methods analyses performed to answer the research questions.

To correctly interpret Likert data, one must first comprehend each item's measuring scale. The numbers assigned to Likert-type items indicate a "more than" relationship but no indication of how much greater. Likert-type items fall into the ordinal measuring scale because of these criteria. A mode or median for central tendency and frequency for the variability is recommended descriptive statistics for ordinal measurement scale items (Boone & Boone, 2012). Because a Likert item involves ordinal data, parametric descriptive statistics like mean and standard deviation are not the best measure to use when analysing individual Likert items. It is preferable to report the mode, median, range, and skewness (Schrum et al., 2020).

The quantitative data of pre-post-questionnaires were entered into SPSS software and coded for analysis. The scores on the instrument’s statements were not normally distributed, which is common when measuring outcomes in social science studies

(Bono et al., 2017). Because the questionnaires' response data did not fulfil the assumptions of normality of popular parametric tests, such as a t-test, it was most appropriate to employ non-parametric tests for hypothesis testing (Field et al., 2012). When parametric test assumptions cannot be satisfied or the sample size is small, like in this study (n=14), standard non-parametric tests have been utilised. The Wilcoxon rank sum test, Wilcoxon signed-rank test, and Kruskal-Wallis test are the most used non-parametric tests (Dwivedi et al., 2017). For the quantitative analysis of questionnaires' responses, a non-parametric test, the *Wilcoxon signed-rank test,* was used. The Wilcoxon signed-rank test is a dependent non-parametric test used to compare teachers' responses before and after participating in the training workshop (Field et al., 2012).

Rosenthal's ***r***, which Rosenthal (1994) introduced as an alternative measure of effect sizes appropriate for non-parametric research, is recommended as a measure of effect size by Field et al. (2012). Rosenthal and Rosnow (1984) established the following standards for reporting the ***r*** value: *-.10* for a small effect, *-.30* for a medium effect, and *-.50* for a big impact. The absolute (positive) standardised test statistic ***z*** divided by the square root of the number of pairings yields the effect size. The results of the Wilcoxon signed-ranks tests are reported using the median (*Mdn*), the ***z*** statistic, ***p*** value and ***r*** value in Chapter 5.

4.10.2 Qualitative analysis

Data analysis can start with a qualitative approach during the data collection stage (Cohen et al., 2011). When attempting to comprehend a group of experiences, thoughts, or behaviours present across a data collection, thematic analysis is a suitable and effective technique (Braun & Clarke, 2012).

The first step in qualitative data analysis is to organise and prepare the data (Braun & Clarke, 2006; Creswell, 2013; Nieuwenhuis, 2016). Voice recordings must be transcribed. After systematically structuring the data, the researcher should familiarise herself with it by reading it several times (Braun & Clarke, 2006; Creswell, 2013; Nieuwenhuis, 2016). Then comes coding, which might be emergent (open coding) or a priori (where constructs of a theory are applied). The coding process will reveal themes in the data. That depends on whether its content, discourse, or other analysis.

The selected themes must be synthesised into a coherent narrative to communicate the findings (Braun & Clarke, 2006; Creswell, 2013; Nieuwenhuis, 2016).

The qualitative data included weekly journal entries and interview transcripts. The interviews were voice-recorded and transcribed. The data collected for the qualitative element's analysis were entered into ATLAS.ti software and coded. As per Braun and Clarke (2006), thematic analysis was performed to answer RQ3, RQ4 and RQ5. To begin this thematic analysis, I looked through the journal entries and interview data to see how the teachers articulated their CT knowledge and self-efficacy. These extracts were then categorised under codes, and themes were generated, provided in Chapter 6.

4.10.3 Quantitative and Qualitative Results Integration

To acquire a more comprehensive understanding of the data, the findings from the quantitative and qualitative analyses were combined. Creswell & Clark (2017) suggested presenting quantitative and qualitative results in a joint display table when integrating findings from those analyses to indicate whether the quantitative and qualitative results confirm or contradict each other. Because mixed approaches were used to answer the research questions, joint display tables are included in Chapter 6.

4.11 Quality assessment

For the data to be included in the analysis, the participant should have completed the pre-post-questionnaires and attended the training workshop. Both quantitative and qualitative studies need to prove their credibility (Golafshani, 2003). Quantitative studies prove this by using instruments, whereas qualitative studies mostly rely on the researcher to put effort into the credibility of the study (Golafshani, 2003).

4.11.1 Quantitative Data

A pilot study was conducted with ten teachers from three primary schools in Windhoek for the researcher to evaluate the sampling frame, evaluate participant response level, acquaint with the research setting, estimate length, and test the research method. Two teachers found one statement to be confusing, and it was revised. Another teacher suggested for the questionnaires to be shortened to avoid lengthy completion times. This resulted in some similar statements combined and the introduction section was shortened.

Reliability

Reliability relates to the consistency and stability of the measuring instrument utilised throughout time. In other terms, the capacity of measuring instruments to produce identical results when used at different times (Sürücü & Maslakçi, 2020). Cronbach's alpha is the most widely used test for determining an instrument's internal consistency. The average of all correlations in each combination of split-halves is determined in this test. This test can be used with instruments that include questions with over two answers. A value between zero and one is Cronbach's result. A reliability score of 0.7 or above is considered satisfactory.

The pre-post instrument was tested for internal consistency during the pilot study by using Cronbach's Alpha test. Each construct was tested separately, and the results were satisfactory, scoring above 0.7. Cronbach's α internal consistency coefficient was *0.706* for CT knowledge, *0.701* for CT beliefs & values and *0.709* for CT self-efficacy.

Validity

Validity measures how well a measuring instrument fulfils its purpose by determining if it measures the behaviour or quality it measures (Anastasi & Urbina, 1997). The understandable and suitable interpretation of the data produced from the measuring instrument because of the analysis determines validity (Heale & Twycross, 2015). Construct validity refers to whether or not you can deduce test scores from the idea under study. How much the instrument assesses the notion, behaviour, idea, or quality—that is, a theoretical construct—that it promises to measure is called construct validity (Sürücü & Maslakçı, 2020).

The researcher relied on the already established validity of the original instrument. Our sample size was too small to carry out appropriate exploratory and confirmatory factor analysis to assess the validity of the modified instrument. Komperda et al. (2018) highlight that when an instrument is modified (i.e., changing its words), it may influence its structural validity, and researchers should conduct a rigorous psychometric analysis of the modified instrument. However, the researcher did not employ any factor analysis due to the small sample size in this exploratory research. But STEBI instrument was modified for disciplines such as math by Enochs et al. (2000) and engineering (Kaya, Newley, et al., 2019). This cited research evaluated the instrument's validity on a large-scale and found similar factor loadings reported by (Enochs & Riggs, 2002). The researcher relied on the already established validity of the original instrument.

4.11.2 Qualitative Data

Qualitative studies commonly rely on “credibility, neutrality, consistency and applicability” rather than the notion of validity and reliability, which are more applicable to quantitative studies (Golafshani, 2003, p.601). Triangulating data from multiple sources and methods will strengthen the credibility of the findings (Cohen et al., 2000).

Guba (1981) contrasted rationalistic and naturalistic inquiry and provided four criteria for assessing the reliability of naturalistic inquiries. Naturalistic investigations frequently employ qualitative research methodologies because they can uncover a subjective reality of the phenomenon under investigation. Guba (1981) identified four criteria: *credibility, transferability, reliability, and confirmability*.

Researchers must ensure that study participants are appropriately identified and described to build *credibility*. To ensure that the results are credible, assess how well the categories cover the data and determine whether there are similarities and variances within and between categories (Elo et al., 2014).

The interpretive equivalent of generalizability, *transferability,* refers to how much qualitative research results can be applied to other contexts with different respondents (Bitsch, 2005; Tobin & Begley, 2004). The “researcher facilitates the transferability judgement by a potential user through ‘thick description’ and purposeful sampling,” according to Bitsch (2005, p. 85).

Reliability, Confirmability and Transferability

In a qualitative investigation, the goal of *reliability* is to back up the claim that the findings are “worth paying attention to” (Lincoln & Guba, 1985). This is especially critical when utilising inductive content analysis, because without a theory-based categorisation matrix, categories are formed from raw data.

The degree to which the findings of an investigation could be validated or corroborated by other researchers is called *confirmability* (Baxter & Eyles, 1997). Confirmability is “concerned with establishing that data and interpretations of the findings are not figments of the inquirer’s imagination but are clearly derived from the data” (Tobin & Begley, 2004, p. 392).

The researcher has attempted to assure the reliability and confirmability of the qualitative analysis by merging the qualitative findings with the quantitative findings when analysing the data from the study. To address transferability, the researcher has given information on the Namibian setting and the participating teachers that researchers and training providers can use to determine if the findings apply to their context. To ensure dependability, the researcher has also informed the methods used throughout each theme analysis she has conducted.

4.12 Phase 2: Validation of the framework

4.12.1 Method

The validation phase of the framework used a survey method to collect responses from expert reviewers. Survey research is gathering data from a sample of people based on their answers to questions. This kind of study permits several techniques for participant recruitment, data collection, and instruments. Survey research can make use of quantitative research techniques (such as utilising numerically scored items on questionnaires) and qualitative research techniques (e.g., using open-ended questions) (Brant et al., 2015).

4.12.2 Sampling

Purposive sampling was used in the study's second phase to pick study participants for the framework validation. Purposive sampling involves selecting participants based on the researcher’s judgement and because the researcher believes the participants possess the necessary traits and expertise (Cohen et al., 2011). Five experts were carefully chosen to obtain a variety of viewpoints. According to one definition, experts have “institutionalised authority to construct reality” (Hitzler et al., 1994, as cited in Meuser & Nagel, 2009, p. 19). Experts in this study were persons who work in the industry or education sector and have important roles in the concerned organisations (Bogner & Menz, 2009).

4.12.3 Participants

The validation process for the framework involved presenting it to five experts from the education sector and the ICT industry for assessment. These reviewers were not involved in the study's initial data gathering. This was done to ensure they could honestly assess the study and the suggested framework while still being critical.

Table 4.7 below shows the details of the reviewers and the rationale behind choosing them.

Table 4.7 Validation of expert reviewers

No	Position	Rationale
1	Head of Department (Senior Primary Phase): Mathematics	To get views on applying CT concepts to existing subjects.
2	Senior Education Officer (Senior Primary Phase): Humanities	To get views on PD programmes for teachers using the framework.
3	Primary School Principal	To gain insight into school context and teacher support at school.
4	Chief Education Officer: Professional Development & Advisory Services (Directorate of Education, Arts & Culture)	To understand how PD for CT can be conducted at a regional level.
5	Retired Senior Education Officer: ICT (now Africa Code Week Trainer for teachers)	Get views on training teachers on programming and technology integration.

4.12.4 Data collection

To collect data during the validation phase of the framework, an online questionnaire (Appendix N) was used to collect qualitative data from experts using Google Forms. The link to the questionnaire was emailed to the experts after they consented to participate in the validation process. The online questionnaire consisted of four (4) open-ended questions to get an in-depth expert view of the framework. The questionnaire design used to collect the data for this phase was built around open-ended inquiries that produced qualitative data that could be examined from an interpretative perspective.

4.12.5 Data Analysis

During the study's second phase, where the framework was validated, the collected data was thematically analysed as per Braun and Clarke (2006). Thematic analysis is a qualitative data analysis approach that involves searching through data collection to locate, evaluate, and report on repeating patterns. It is a data description approach but requires interpretation in picking codes and generating themes (Braun & Clarke 2006). The qualitative data from the online questionnaire was entered into ATLAS.ti software for analysis and coding where after themes emerged. Findings from these themes led to the refinement of the initial PD4PCT framework.

4.13 Ethical considerations

Consideration of ethics in research is important, especially when it involves people as participants in the study and to maintain the reputation of the concerned learning institution (Myers, 2013). Ethics in research entail how the study is conducted with consideration of morals and responsibility (Blumberg et al., 2005). Myers (2013, p.45) outlines some aspects of “honesty, plagiarism and informed consent” about ethics in research. The researcher enforced honesty measures to ensure that the study was credible. The University of Pretoria, where the researcher is enrolled for the doctoral study, enforces very strict measures on plagiarism, and this research study has tried to adhere to the policy on plagiarism.

The participants were not forced to take any part in this study. The participants were informed of the intent and benefits of the study, and a consent letter (Appendix C) was given to the participants for signing, and this was an undertaking by the researcher to maintain credibility with the participants. The interviews were conducted on time schedules suitable for the individual participants of the study. All recordings and notes were taken with the informed consent of the individual interviewees, and their rights to privacy, confidentiality and anonymity were carefully considered.

Ethical principles are maintained throughout the research to guide integrity and quality.

- Ethical clearance to collect data was obtained from the University of Pretoria (Appendix A).
- The researcher obtained a permission letter for access to the site and participants from the Khomas Directorate of Education, Arts & Culture (Appendix B).
- The researcher informed the participants that their participation in this study is voluntary.

- The researcher ensured the physical and emotional safety of participants during the study.
- The researcher ensured transparency and trust with study participants.
- All participants and site anonymity are maintained on all data records and reports by using pseudonyms.

4.14 Conclusion

This chapter explained how the research was conducted. It discussed the research paradigm, approach, methodological choices, data collection and analysis techniques used for the study. The next chapter discusses how the developed framework was implemented.

5 CHAPTER 5: IMPLEMENTATION OF THE FRAMEWORK: CASE STUDY

5.1 Introduction

This chapter discusses the implementation of the PDF4PCT through the case study. The objective of implementing the framework is to establish if it meets the objective of assisting teachers in integrating CT into their teaching through PD programmes.

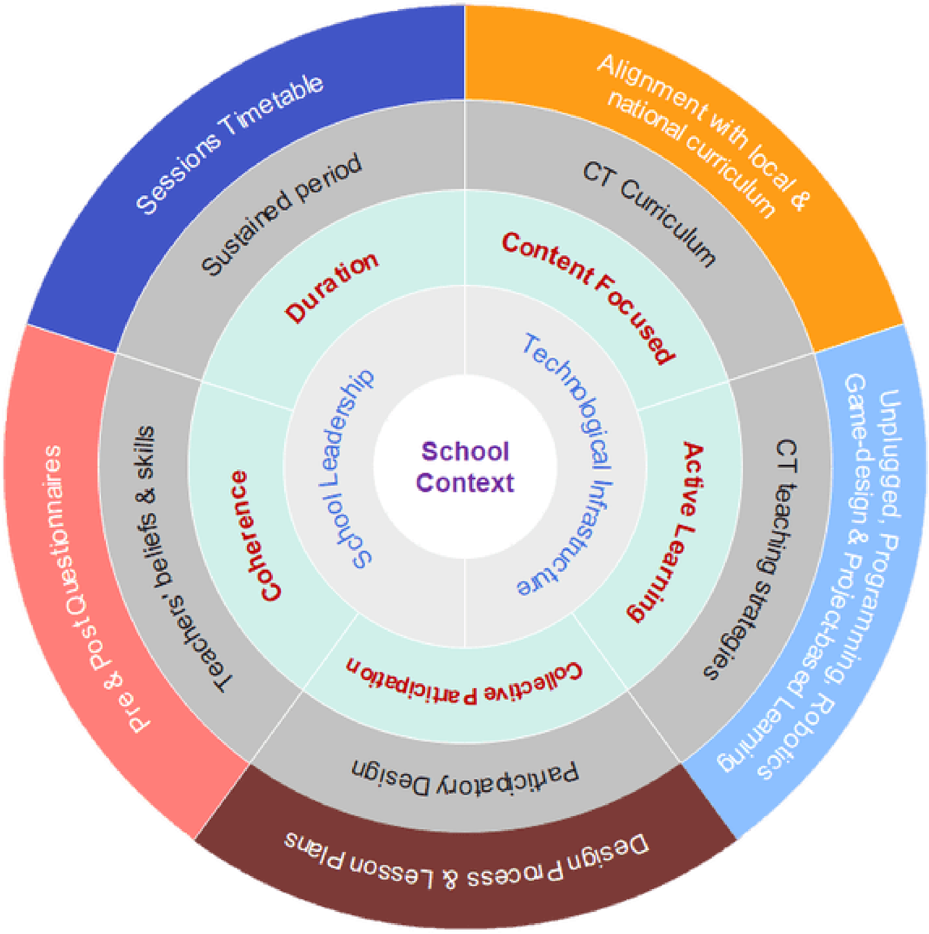


Figure 5.1 The initial PD4PCT framework

To ensure the effectiveness of PD, the framework was applied through a participatory design training workshop for the teachers, as discussed in the following sections.

5.2 School Context

The school, which was the case, is a primary school in the informal settlement of Windhoek in Namibia. The school leadership has a positive outlook towards CT and supports being part of the study. The school principal and head of departments (HoDs) encouraged teachers to be participants, making it easier for teachers to be keen participants in the study.

The school does not currently have a computer lab for learners but has a few desktops and laptops for teachers' use. According to Table 3.4, teachers can choose activities based on their context and technological needs. Using an unplugged teaching strategy is much preferred because of the school context, and teachers created activities like paper programming (Figure 5.4). However, the programming strategy was also tried by teachers with the few computers they had as chosen during stage 2 of the design process (Figure 3.2), and teachers created interactive activities and games using Scratch software (Figure 5.6).

5.3 Duration of the training workshop

A detailed workshop timetable was developed to guide the sessions (Appendix J). The training workshop was four weeks long, from 11 October 2021 to 3 November 2021. The training sessions were two hours long and conducted after-school for three days per week. The sessions were 24 hours of contact. Teachers could choose suitable dates based on their availability, which established a sense of self-motivation and control (Desimone, 2009).

5.4 Collective participation through participatory design

The table below gives the summary of the design process implementation with a detailed discussion following after.

Table 5.1 Summary of participatory design process implementation

Stage	Activities	Form	Purpose
Plan	• Group teachers in small teams per subject. • Provide timetable and tools needed. • CT poster, pens, blank papers, computers with Scratch programme and lesson plan templates.	Telling	Probing
Discover	• Teachers, select topics from the local/national syllabus they feel are suitable for training (Trade, Sequences, Ecosystem, Grammar, Action Words). • Provide advice on a topic selection with CT linkage.	Telling	Priming participants to immerse them in CT.
Design	• Practical lesson development considering classroom context. • Work with teachers to define key ideas of subject matter (CT).	Making	Get a better understanding of their current experience.
Present	• Modelling and practising activities & teaching methods. • Working through CT activities with teachers in the role of learners.	Enacting	Generation of ideas or design concepts for the future.
Feedback	• Ask teachers to lead the review of material and instructions.	Telling & Enacting	Generation of ideas or design concepts for the future.

To achieve the characteristic of collective participation, teachers were grouped according to the subject taught during the participatory design workshop to work collaboratively as planned in **stage 1 (Planning)** of the design process (Table 5.1). This resulted in five groups of Social Studies, Natural Science, Mathematics, English and Afrikaans. The Social Studies and English groups consisted of three teachers each; the Natural Science group had four teachers, while Mathematics and Afrikaans groups had two teachers each. Each group was given pens, blank papers, CT Poster, a computer with Scratch software, a session timetable, etc.

In the true participatory design approach, the design process enabled the teachers to have control over the design process by letting them choose their unit topics to work with and what tools to use during **the Discovery Stage** (stage 2) of the design process (Table 5.1). They could also develop authentic lesson plans and activities for their specific subjects that can be used in their classrooms. The Social Studies group chose the topic of trade (import & export), while the English group chose to work with grammar and usage (verbs), and both used Scratch for their quiz activities. Meanwhile, the Natural Science group chose the topic of the food chain, and the Afrikaans group chose to work with action words for their lesson plan and activity. Teachers in the mathematics group worked with sequences topics. The participatory workshop allowed teachers to ask questions about CT knowledge and integration as design partners through this process. As a facilitator, I shared with them examples of how CT skills are integrated across different subjects and ensured that they had access to all the tools needed by providing lesson plan templates (Appendix H & I).

During the Design stage (stage 3), participants were grouped with a maximum of four individuals per group to make the designing phase a collaborative experience. At the design stage (Table 5.1), group activities included brainstorming on lesson plan components, paper and mock-up drawings, and concept presentations. The groups that used the programming approach for their activities had to first use pen and paper to design their activities before writing the programme on the computer using Scratch.

During the **Presentation stage** of the design process (Table 5.1), the groups presented their artefacts and peer-reviewed each other. At this stage, the teachers modelled both the unplugged and programming teaching strategies with other teachers in the role of learners. Ensuring that each member contributes to the group's responsibilities enhanced inclusive participation (Agbo et al., 2021). The final **Feedback Stage** (Table 5.1) saw teachers giving other teams feedback on their artefacts, leading to iterative improvement of the lesson plans and activities. Some groups had to include new CT concepts they did not think of applying to their lesson plans and activities because of the advice other groups gave during reviewing artefacts.

Below is an example of a lesson plan in Figure 5.2 designed by English teachers. The lesson plan was based on the use of Scratch programming language where a sprite (character, e.g., cat or person) on the screen performs an action when clicked, e.g., jumping, and learners should identify what that action was and type in the answer. Learners will need to work in groups hence applying the CT collaboration approach.

Lesson Plan

Grade: 4C	**Which key learning topic is this lesson plan for?**
Subject: English second language	Grammar and usage : Verbs

Syllabus Outcome(s): *What do students learn and are able to do as a result of this lesson?*

Write with progressively more accuracy in spelling, punctuation and refencing using appropriate, vocabulary, idioms and parts of speech in a range of sentences structures.

Introduction: *How will you get the students motivated, curious and ready to learn?*

The learners will observe the action being done by the sprites on the computer, and discuss the actions in groups of four.

Metalanguage: *What are the key concepts or procedures that you want students to understand as a result of this lesson?*

They should be able to identify the action being done by the specific or by the sprites on the computer.

Computational Thinking: *Which of the computational concepts, practices and perspectives will students have the opportunity to learn about in the lesson?*

Algorithm, Collaboration and perservering

Teaching Activities: *What strategies will you use to teach the content and skills? How long will you spend on each of those strategies and with the content? How would you address different levels or prior knowledge?*

When the sprite is clicked, it will perform a certain action. The learners will then discuss the actions done by each sprite.

Lesson Conclusion: *How will you bring the lesson to a conclusion?*

The computer will display the answers.

Assessment: *How will you know whether the students achieved what you wanted them to achieve?*

The computer will provide the scores.

Resources: *What materials do you need for this lesson? Have you used ideas from elsewhere?*

Computer with Scrach

Figure 5.2 A CT-infused lesson plan for English prepared by teachers

5.5 Enhancing Teachers' Knowledge

For the teachers to understand CT and its integration, the CT curriculum used in training focused on CT definition, CT concepts and approaches.

The first day of the training was devoted to CT knowledge, whereby teachers were introduced to the definition of CT, the CT concepts of algorithms, abstraction, decomposition, pattern recognition and evaluation as per Table 3.1 for Grades 3-5 and Grade 6-7. To apply the constructionist approach in ensuring the teachers use their prior knowledge to combine with the new CT knowledge, at the planning stage, each CT concept was explained with an example and its association as indicated in the learning material (see Appendix J) (Kurt, 2021; Loi, 2004). The CT approaches of tinkering, creating, debugging, persevering, and collaborating were introduced once the teachers were familiar with CT and its skills. Each teacher received a CT poster at the beginning of the session, as Figure 5.3 shows below, to help visualise the meanings of CT concepts and approaches.

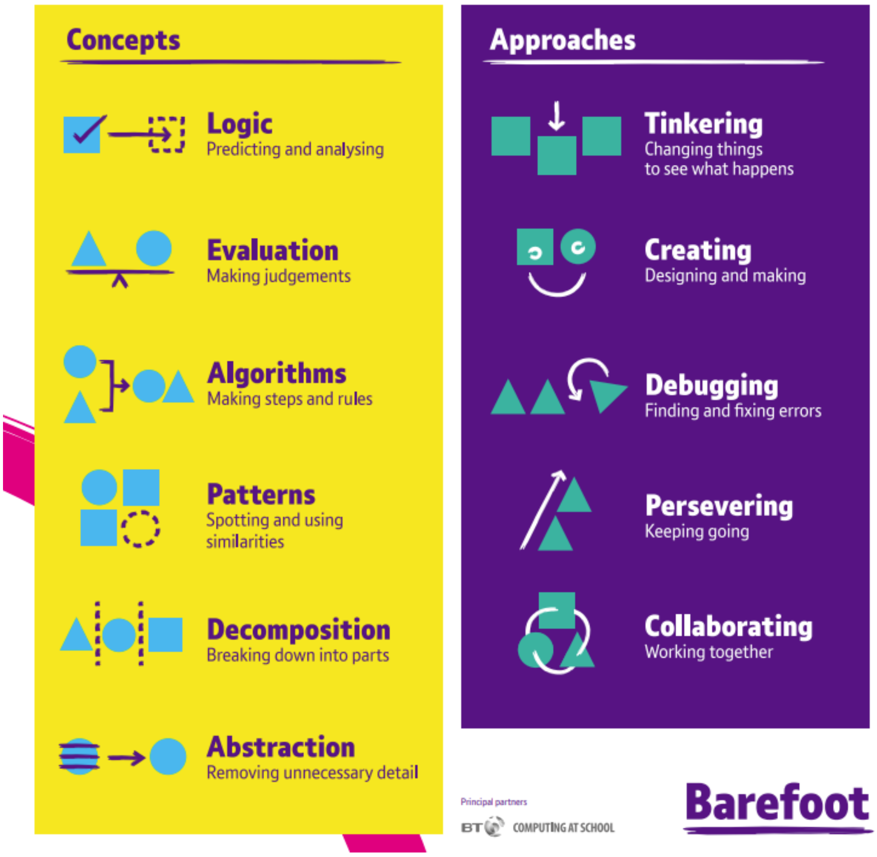


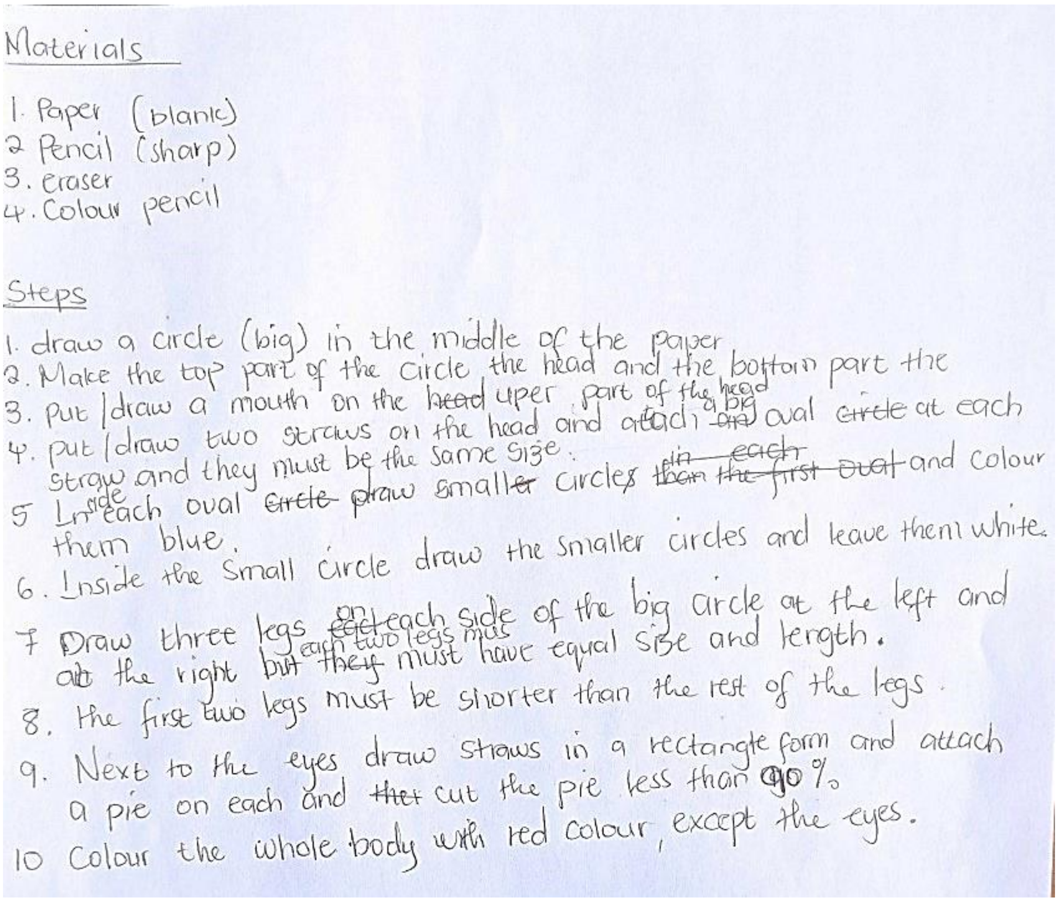
Figure 5.3 CT Poster (Source: http://barefootcas.org.uk)

5.6 Active learning through Constructionism

During the training, teachers were introduced to different teaching strategies for CT, such as unplugged, programming, robotics, game design and project-based learning. However, only unplugged programming strategies were used during the workshop. This session later introduced the teachers to the programming concepts of sequences, loops, and conditionals ahead of the hands-on activities in the following session.

As the study is rooted in constructionism, teachers were active participants during the training sessions by doing hands-on activities using unplugged and programming strategies. Teachers discussed and designed the tasks and activities during the training sessions with the help of the researcher as a facilitator. The process was a constructionist learning experience for teachers due in part to their participation in developing artefacts as “public entities,” as per Papert and Harel (1991), to be shared with their colleagues. As a facilitator, the researcher emphasised constructionist design principles to ensure that the resulting CT-infused units would also be constructionist (Kelter et al., 2021).

For the unplugged strategy, teachers used pen & paper to design and carry out the tasks without computing devices. This was decided in the design process's second stage (discover), as shown in Figure 3.2. based on the school context component of the framework. For example, in one activity called “Draw a crab,” a group of teachers was shown a simple picture of a crab and had to write instructions for how to draw it. These instructions were then given to a different group to follow. This activity tested the skills of abstraction, algorithms, rules, cause, and effect. Figure 5.4 below showcases an output of the activity as carried out by teachers. This showed the importance of giving clear and concise instructions to someone or a computer. The result of this activity is shown in Figure 5.5 below.


Materials

1. Paper (blank)
2. Pencil (sharp)
3. eraser
4. Colour pencil

Steps

1. draw a circle (big) in the middle of the paper
2. Make the top part of the circle the head and the bottom part the
3. Put/draw a mouth on the ~~head~~ uper part of the head
4. put/draw two straws on the head and attach a big ~~an~~ oval ~~circle~~ at each straw and they must be the same size.
5. Inside each oval ~~circle~~ draw smaller circles ~~than the first oval~~ ~~in each~~ and colour them blue.
6. Inside the small circle draw the smaller circles and leave them white.
7. Draw three legs on ~~each~~ each side of the big circle at the left and at the right ~~but they~~ each two legs must have equal size and length.
8. The first two legs must be shorter than the rest of the legs.
9. Next to the eyes draw straws in a rectangle form and attach a pie on each and ~~then~~ cut the pie less than 90%.
10. Colour the whole body with red colour, except the eyes.

Figure 5.4 Unplugged Algorithm activity to draw a crab created by teachers

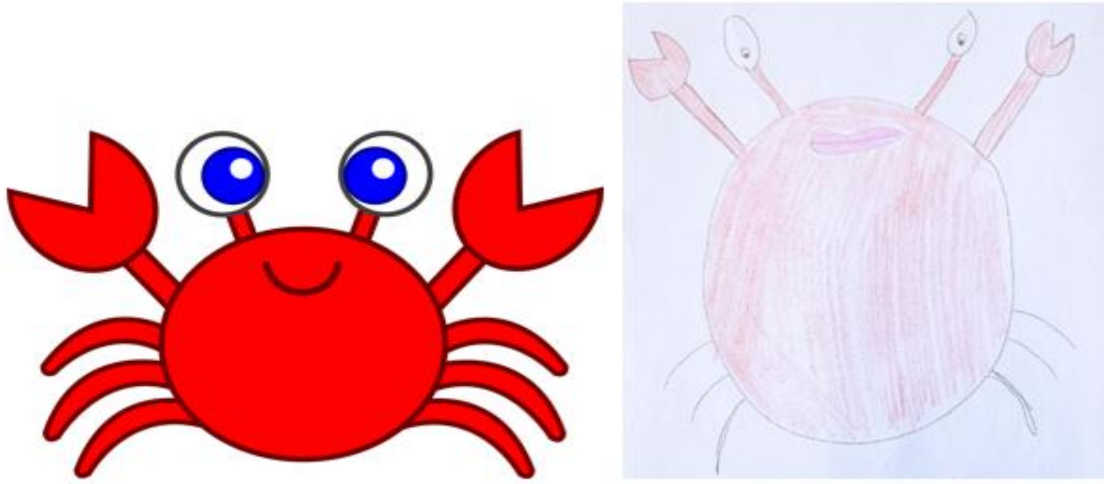


Figure 5.5 Original Crab picture on the left (Source: http://clipart-library.com/) Crab drawn by teachers on the right based on instructions given in Figure 5.4

Meanwhile, for the programming strategy, the teachers used computers with Scratch software to create programmes for the activities. This session taught teachers how they could teach CT concepts through the programming/coding strategy using Scratch, which is a drag and drop visual programming language. Before the hands-on Scratch activities, teachers were introduced to Scratch concepts such as sprites, blocks, costumes, move, turn, etc. During the hands-on session with Scratch, teachers worked in groups formed in the planning stage (Table 3.3) to complete an activity called "Dancing cat." In this activity, they needed to apply the loop concept to make the cat talk and move. This needed algorithmic thinking, collaboration and debugging from team members to complete the task successfully. Figure 5.6, shown below, is a screenshot of the code from the teachers' activity.

Participants frequently interacted between teams. After each session, teachers and the researcher provided each other with comments on their respective artefacts. During stage 4 (present) of the design process (Figure 3.2), all the participants gathered to discuss their successes and difficulties. As 'thinking and talking' about what you do is a central principle of constructionist learning, discussion opportunities inside and between co-design teams were essential to transforming the design process into a constructionist learning experience (Papert & Harel, 1991).

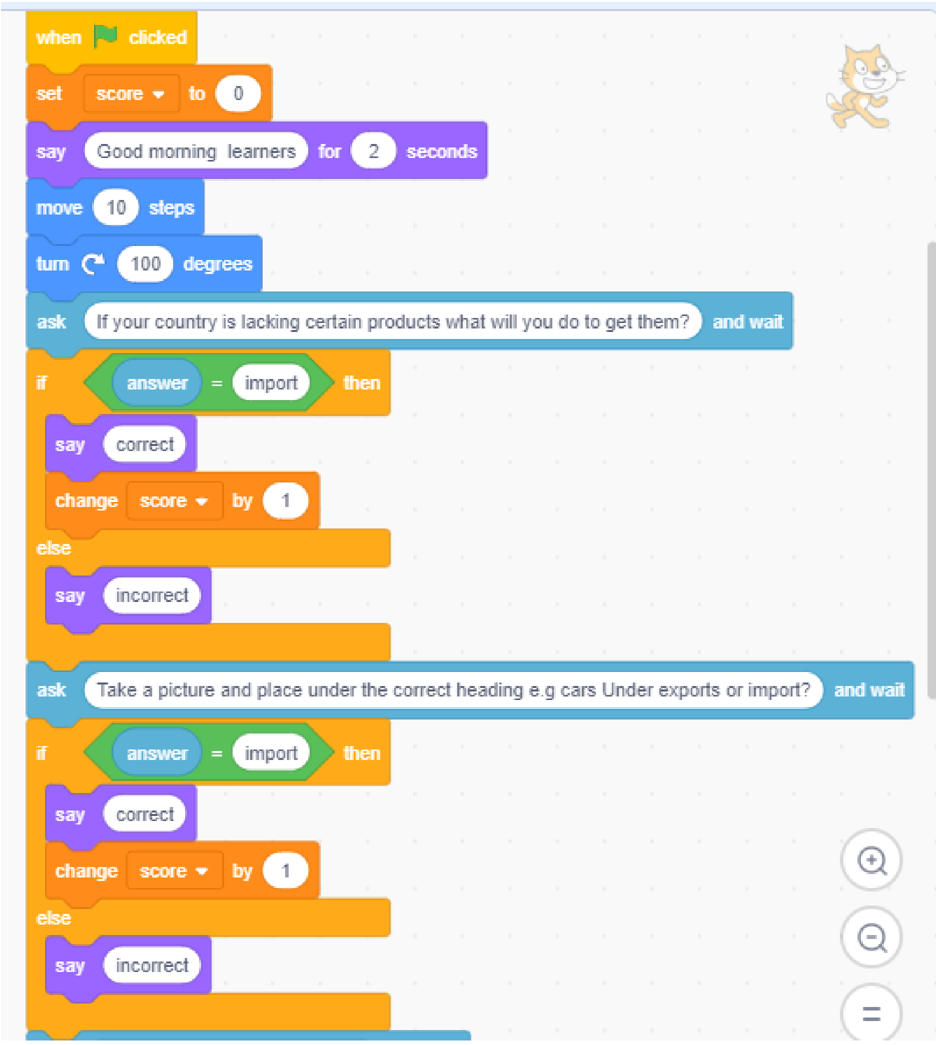


Figure 5.6 Social Studies Teachers artefact from hands-on activity with Scratch

As analytical tools, constructionism's foundational theoretical components, such as personally meaningful artefacts and debugging, were utilised to analyse design and learning (Kelter et al., 2021).

5.7 Teachers' self-efficacy and curriculum features

After the training, the training provider needs to know the skills and beliefs of teachers towards CT, its importance, and the value it adds to their classrooms (Rich, Larsen & Mason, 2020). These were measured through a questionnaire before and after the participatory training workshop (see Appendix D & E). CT knowledge comprehension, beliefs and self-efficacy were measured to indicate if the PD programme influenced these constructs and if teachers are likely to integrate CT into their classrooms.

As the participatory approach was used with a constructionist teaching approach, teachers produced new or adapted current curricular materials to incorporate a focus on CT into numerous subjects and customised instruction according to their needs during the Design stage of the participatory design process, stage 3. The training content-related to relevant topic areas, and school curricula fostered both CT skills and subject matter expertise as the framework builds on the constructionist approach (Rich, Rich et al., 2019). Based on Table 3.1, the scope and sequencing represented a multi-grade (or multi-unit) sequence that gives various chances for learners to interact with concepts and activities and attain the intended learning outcomes (Sherwood, Yan, et al., 2020).

5.8 Teachers' Support

The school management supported the teachers by offering them access to a CT-integrated PD programme demonstrating five essential characteristics of effective PD: topic concentration, active learning, coherence, sustained duration, and group engagement (Sherwood, Yan et al., 2020).

5.9 Summary of CT knowledge, subject knowledge, and active learning strategies for each teacher group

Table 5.2 below summarises what each group did during the participatory training workshop. It shows what CT concepts and approaches were applied during their lesson plans and activities, and which learning strategy was used.

Table 5.2 Summary of CT knowledge, subject knowledge, and active learning strategies for each teacher group.

Group	Subject	CT knowledge focus	CT thinking practices	Subject curriculum focus	Active learning strategy	Lesson plan
Group 1	Social Studies	Algorithms Decomposition	Collaboration Persevering	Trade (import-export)	Unplugged, Programming	See Figure 5.6
Group 2	English	Algorithms	Collaboration Persevering	English grammar	Programming	See Figure 5.2
Group 3	Natural Science	Algorithms, Pattern recognition	Collaboration	Ecosystem (food chain)	Unplugged, Programming	Append ix: L
Group 4	Mathematics	Algorithms, Decomposition Pattern recognition	Collaboration Persevering Debugging	Sequences	Unplugged	Append ix: L
Group 5	Afrikaans	Debugging	Creating Tinkering Collaboration	Action Words	Programming	Append ix: L

5.10 Conclusion

This chapter looked at implementing the PD4PCT framework. It detailed the steps followed during the participatory training workshop through a single case study. It also detailed all six components of the framework and what was done under each. As constructionism was applied during the active learning component of the framework, teachers produced artefacts provided as figures in the chapter. The next chapter discusses the findings from implementing the PD4PCT framework.

6 CHAPTER 6: FINDINGS FROM THE CASE STUDY

6.1 Introduction

This chapter presents findings from the case study analysed to test and refine the PD4PCT framework. The chapter is divided into sections according to the relevant components of the framework, namely CT content knowledge, coherence, active learning, and collective participation. The discussions of how the findings from the analyses relate to findings from related research are presented in Chapter 7.

Qualitative data were analysed using both inductive and deductive approaches following the most widely accepted methodology for thematic analysis by Braun and Clarke (2006), which entails a six-step procedure: becoming familiar with the data, creating preliminary codes, looking for themes, reviewing themes, defining, and naming themes, and preparing the report.

After the interview scripts and journal entries were transcribed and imported into Atlas.ti software program, 20 initial codes were created from the data. As a result of the initial codes, themes were generated inductively, and some were merged into one after the review as they were not distinct enough to merit separation (see Appendix K). After review, seven themes emerged and are reported in the following sections. But the components of the framework were used to deductively analyse the interview results and the journal inscriptions. The results from both these qualitative analyses are provided below.

The participants were labelled to make analysis easier. Table 6.1 below shows which teachers belonged to what group and summarised the CT concepts, practices and learning strategies used per group.

Table 6.1 Summary of CT concepts, subject knowledge, and active learning strategies for each teacher group

Group	Subject	CT knowledge focus	CT thinking practices	Subject curriculum focus	Active learning strategy	Lesson plan
Group 1 ***Teachers:*** A, E, M	Social Studies	Algorithms Decomposition	Collaboration Persevering	Trade (import-export)	Unplugged, Programming	See Figure 5.6
Group 2 ***Teachers:*** B, G, L	English	Algorithms	Collaboration Persevering	English grammar	Programming	See Figure 5.2
Group 3 ***Teachers:*** C, D, I, J	Natural Science	Algorithms, Pattern recognition	Collaboration	Ecosystem (food chain)	Unplugged, Programming	Append ix: L
Group 4 ***Teachers:*** K, N	Mathematic s	Algorithms, Decomposition Pattern recognition	Collaboration Persevering Debugging	Sequences	Unplugged	Append ix: L
Group 5 ***Teachers:*** F, H	Afrikaans	Debugging	Creating Tinkering Collaboration	Action Words	Programming	Append ix: L

The discussion below provides the findings from the qualitative and quantitative data analysis per components of the PD4PCT framework.

6.2 CT content knowledge

This section includes the analysis of items from the pre-post-questionnaires conducted to measure the teachers' CT knowledge before and after the implementation of the framework.

6.2.1 Quantitative Analysis

When comparing the responses before and after the training, there were substantial increases in the median values to all CT knowledge statements, with large effect sizes. The teachers who attended the training event had a perceived deeper knowledge of CT. This comparison's statistical findings are summarised in Table 6.2. As described in Chapter 5, the data distribution was not normal; hence, the Wilcoxon signed-rank test was used to compute the *p* value. The effect sizes (*r*) were determined using *Rosenthal's r* formula.

Table 6.2 Summary of the CT Knowledge items responses (N=14)

Item No	Item Description	Pre-Median	Post-Median	*P* value	*r* value
1	I can define what computation thinking (CT) is.	2.00	4.00	0.002	-0.59
2	I can describe fundamental computational thinking concepts (e.g., algorithms, abstraction, decomposition, pattern recognition & evaluation).	2.00	4.50	0.002	-0.58
3	I can describe fundamental coding/programming concepts (e.g., loops, variables, conditional logic).	0.00	4.00	0.001	-0.61
4	I can look at a process and figure out how to make it more efficient.	2.00	4.00	0.009	-0.49
5	I can suggest different solutions to solve problems.	2.50	4.00	0.003	-0.56
6	I can generalise solutions that can be applied to many problems.	2.50	4.00	0.002	-0.59
7	I am good at finding patterns in data.	4.00	5.00	<.001	-0.67
8	I am good at solving puzzles.	4.00	5.00	0.012	-0.48
9	I can read a formula (e.g., algorithm, equation, input/output process) and explain what it should do.	2.50	4.00	0.003	-0.56
10	When I'm presented with a problem, I can easily break it down into smaller steps.	2.00	4.00	0.003	-0.57
11	When solving a problem, I work with others to solve different parts of the problem simultaneously.	4.00	4.50	0.002	-0.58
12	When solving a problem, I look how information can be collected, stored, and analysed to help solve the problem.	2.00	4.00	0.002	-0.58
13	When solving a problem, I create a solution where steps can be repeated.	2.00	4.50	0.003	-0.56
14	When solving a problem, I create a solution where some steps are done only in certain situations.	2.00	4.00	0.001	-0.61
15	When solving a problem, I try to simplify the problem by ignoring details not needed.	2.00	5.00	0.003	-0.56

Before the training, the teachers' responses to the questions assessing CT knowledge were generally low; however, all median responses increased following the training. Item number 3, "*I can describe fundamental coding/programming concepts (e.g., loops, variables, conditional logic),*" had the lowest median replies of 0.00 prior to training and considerably climbed to 4.00 after training. After the training, the teachers' self-reported CT knowledge had grown dramatically, with large effect sizes across the board.

Overall, the teachers who participated in the training responded highly to item 7, "I *am good at finding patterns in data*." before and after the programme. Therefore, a Wilcoxon signed-rank test revealed that responses were significantly higher at post-training workshop (*Mdn = 5.00, n = 14*) compared to pre-workshop training (*Mdn = 4.00, n = 14), z = -3.56, p = <.001*, with a larger effect size, *r = -0.67*.

6.2.2 Qualitative analysis

Neither the interviews nor the writings in the teachers' journals had any explicit queries concerning the CT concepts; nonetheless, some teachers referenced CT concepts they found difficult to grasp.

In the inductive qualitative analysis, two themes arose regarding the CT content knowledge component of the framework, which are described in the following sub-sections.

Teachers' challenges and needs

Teachers were asked to write in their journals what they found difficult during sessions, and below is what some teachers wrote.

Most teachers stated they had a strong grasp of algorithms, decomposition, and pattern recognition, whereas some indicated in their diaries that they had difficulty understanding programming concepts.

> **Teacher B**: *"The activity was challenging when I wanted to integrate conditionals, repetition and sequencing."*

> **Teacher H:** *"The most challenging thing was making sure that the set of rules or sequences are in a correct order."*

> **Teacher K**: *"The new concepts that I have never encountered before were overwhelming."*

> **Teacher M:** *"Starting up Scratch activity on my own without help due to lack of computer literacy."*

Some teachers indicated through the interviews and journal entries that more time and training are needed as they felt that the allocated hours for this intervention were not long enough to grasp all the concepts and deepen their understanding. Teachers E, F and G, had this to say:

> **Teacher E:** *"I just feel like I still need more, more of this workshop, or let me say more classes of this, just so I can be exposed to more knowledge."*

Teacher F: *"So I just, I just need more of training on how to use, like creating activities, using the computers, like creating with Scratch. That's where I'm not more confident. I need more training as time goes on."*

Teacher G: *"So we need more training for us to do these things".*

Teachers' understanding of CT skills

The results showed that teachers were not aware of how some of their lessons were already aligned with CT skills. During the interviews, some teachers mentioned that after the training, they now realised that they had been using some of the CT concepts in their existing lessons without knowing they are part of CT skills.

Teacher I: *"After the workshop, looking at the CT concepts of logic, evaluation, algorithm and patterns and decomposition, looking at these things, we are already using these things, these concepts. So, if you look at logic in mathematics, we do the estimations and then after that we have to analyse our answers, whether the answers are correct. And then I think we are doing these concepts in our lessons without us knowing."*

The results also revealed that most teachers did not know or have heard anything about CT before the training.

Teacher B: *"At the beginning of the training, I did not understand anything."*

However, after the training, there were still a few who felt they still lacked CT knowledge.

Teacher E: *"I just feel like I still don't have enough information or enough knowledge, although currently I'm really exposed to a lot of things."*

When teachers were asked what they understood by CT at the end of the training, most defined CT as a problem-solving process that can be used in any subject and mentioned its concepts and approaches.

Teacher D: *"computational thinking is a problem-solving process that includes a number of characteristics and dispositions, like, including evaluation, algorithm, patterns, decomposition, abstraction. And also, we have approaches like tinkering, creating, debugging, persevering and collaboration; that's my own view."*

The results suggest that most teachers have improved their CT knowledge after the training with abstraction the least understood concept.

6.2.3 Integration of Quantitative and Qualitative Findings

The quantitative findings and quantitative findings were combined to see where they converged and differed. As illustrated in Table 6.3, three key findings arose. First, the training enhanced the teachers' comprehension of the terminology, concepts, and approaches of CT. This was obvious from the quantitative data, demonstrating that teachers' responses to the CT knowledge questions on the pre- and post-questionnaires have shown considerable improvement. Additionally, in the qualitative data, only a few teachers mentioned having trouble learning programming concepts. In their journal notes, teachers had difficulties comprehending loops and conditions, and in the interview, one teacher mentioned trouble executing sequences. Second, some teachers found it difficult to comprehend computational concepts even after receiving training, which appeared most frequently in the qualitative data when teachers discussed what they found difficult. Third, some teachers failed to give and follow clear directions for completing a task, which was recorded mostly in the journal entries.

Table 6.3 Joint findings for CT knowledge

Conclusion from Integration	Quantitative Findings	Qualitative Findings
Teachers enhanced their comprehension of the terminology, concepts, and approaches of CT.	Teachers' responses to the CT knowledge statements greatly improved, with larger effect sizes.	The majority of teachers stated that they understand what CT is and how to apply its principles and practices.
Some teachers found it challenging to comprehend the programming concepts.		In journal entries and interviews, few teachers admitted to struggling with sequences, loops, or conditionals.
Even after finishing the training, some teachers believe they still lack enough knowledge of CT.		In journal entries and an interview, some teachers stated that they do not yet have enough CT expertise.

6.3 Coherence

This section includes the analysis of items from the pre-post-questionnaires conducted to measure the teachers' beliefs and attitudes towards CT.

6.3.1 Teachers' beliefs and attitude towards CT

Quantitative Analysis

When comparing responses before and after training, the median of participants' responses to all but one of the teachers' beliefs and attitudes towards CT increased significantly, with significant effect sizes. Overall, the teachers who participated in the training programme improved their views and attitudes towards CT. This comparison's statistical findings are summarised in Table 6.4. As described in Chapter 5, the data distribution was not normally distributed; hence, the Wilcoxon signed-rank test was employed to compute the p value. The effect sizes (r) were calculated using *Rosenthal's r* formula.

Table 6.4 Summary of teachers' beliefs and attitude towards CT responses

Item No	Item Description	Pre-Median	Post-Median	*p* value	*r* value
1	Computing should be taught in primary schools.	4.00	5.00	0.004	-0.54
2	Learning about computing can help primary school learners become more engaged in school.	4.00	5.00	0.005	-0.54
3	Computing is like art—you are born with the ability to think that way or you are not.	2.00	2.00	0.915	-0.02
4	Computing content and principles can be understood by primary school children.	3.00	5.00	<.001	-0.64
5	My current teaching situation does lend itself to teaching computing concepts to my learners.	2.50	4.00	<.001	-0.65
6	Knowledge of computer programming is needed in most careers.	1.00	4.00	0.001	-0.62
7	Providing more computational thinking activities is necessary to enrich my learners' overall learning.	1.50	4.00	0.001	-0.61
8	Computational thinking is an important 21st-century skill.	4.50	5.00	0.017	-0.45
9	My current primary school learners will need to know how to apply computing concepts to remain competitive for jobs by the time they are adults.	3.50	5.00	0.005	-0.54

The results show that the teachers' responses to the statements about their beliefs and attitudes towards CT were generally low before the training. As it can be seen from Table 6.3 above, for item 1, "*Computing should be taught in primary schools,*" the *Wilcoxon signed-rank test* revealed that responses were significantly higher at post-training workshop (*Mdn = 5.00, n = 14*) compared to pre-workshop training (*Mdn = 4.00, n = 14), z = -2.85, p = 0.004*, with a larger effect size, *r = -0.54*. The training has positively changed the beliefs and attitudes of teachers significantly on this item.

The statement that said, "*learning about computing can help primary school learners become more engaged in school,*" a *Wilcoxon signed-rank test* revealed that responses were significantly higher at post-training workshop (*Mdn = 5.00, n = 14*) compared to pre-workshop training (*Mdn = 4.00, n = 14), z = -2.84, p = 0.005*, with a larger effect size, *r = -0.54*. The training has positively changed the beliefs and attitudes of teachers significantly on this item.

On the belief that "*computational thinking is an important 21st-century skill,*" a *Wilcoxon signed-rank test* revealed that responses were significantly higher at post-training workshop (*Mdn = 5.00, n = 14*) compared to pre-workshop training (*Mdn =*

4.50, n = 14), z = -2.39, p = 0.017, with a medium effect size, *r = -0.45*. The training has positively changed the beliefs and attitudes of teachers significantly on this item.

After the programme, the participating teachers' responses had a significant positive change with large effect sizes across most items on beliefs and attitudes towards CT.

Qualitative analysis

There were no explicit questions on teachers' beliefs and attitudes in journals and interviews. When answering questions regarding applying coding and CT teachings in their classrooms and how they would use what they had learnt in the programme, teachers shared their confidence in teaching CT throughout the interviews.

6.3.2 Self-Efficacy for Computational Thinking

This section includes the analysis of items from the pre-post-questionnaires conducted to measure the teachers' self-efficacy under the coherence component of the framework.

Quantitative Analysis

When comparing the median responses before and after the training, there were significant increases in participants' replies to all CT teacher self-efficacy statements, with effect sizes more than –0.47. Therefore, the total self-efficacy of the teachers who attended the training session increased regarding CT. This comparison's statistical findings are summarised in Table 6.5. As described in the Methodology chapter, the data distribution was not normally distributed; hence, the Wilcoxon signed-rank test was employed to compute the p value. The effect sizes (*r*) were calculated using *Rosenthal's r* formula.

Table 6.5 Summary of teachers' self-efficacy on CT responses

Item No	Item Description	Pre-Median	Post-Median	*p* value	*r* value
1	I feel confident using computer technology.	4.00	5.00	0.013	-0.47
2	I feel confident writing simple instructions for another person on paper.	4.00	4.50	0.003	-0.56
3	I know how to teach computing concepts effectively without a computer.	2.00	4.00	<.001	-0.63
4	I know how to teach programming concepts effectively without a computer.	2.00	4.00	0.004	-0.55
5	I can promote a positive attitude towards computing education to my learners.	2.00	4.00	0.001	-0.61
6	I can guide learners in using programming as a tool while we explore other topics.	2.00	4.00	0.001	-0.61
7	I feel confident using programming as an instructional tool within my classroom.	2.00	4.00	0.002	-0.59
8	I can adapt lesson plans incorporating unplugged activities as an instructional tool.	2.00	4.00	<.001	-0.64
9	I can adapt lesson plans incorporating programming as an instructional tool.	2.00	4.00	0.003	-0.57
10	I can identify how computational thinking concepts relate to the syllabus.	2.00	4.00	0.002	-0.59

Before the training, teachers' responses to questions on teaching CT concepts through unplugged or programming were poor. The median score for the teachers' responses to the items before the training programme was 2.00. The replies to these questions revealed that teachers lacked confidence in teaching these concepts before the training.

The self-efficacy of the participating teachers rose significantly after the training, with large effect sizes across most items, except for the confidence in utilising technology items, for which the answer median was less than -.50.

Qualitative Analysis

During the interviews with the groups, teachers were questioned about their confidence in teaching their students CT skills. Teachers addressed their confidence levels and what prevents them from feeling confident to teach CT concepts and how to overcome these obstacles.

Three themes related to teachers' confidence in teaching CT were uncovered in the qualitative analysis for coherence and are presented in the following sub-sections.

Teachers' low confidence about CT after training

When it comes to feeling confident to teach CT to learners, few teachers mentioned in the interviews, they do not feel confident after training to teach CT skills to their learners.

> **Teacher D**: *"I'm not so well confident because I haven't yet grasped all the approaches and concepts. So maybe I will have to, I will need more maybe training or hands-on activities so that I can practice."*

> **Teacher E:** *"Um, I would say I'm still not confident enough to pass on this, uh, to develop the student computational thinking capabilities yet."*

Teacher C felt confident using the unplugged strategy to teach CT skills to learners as she struggled with Scratch and programming strategy.

> **Teacher C: "***Yeah, just maybe a little bit, I'm confident on using the unplugged way, uh, than the computer system."*

Some teachers discussed that they felt unable to teach CT skills via programming independently, or that they lacked the skills to learn coding after the training. Teacher G discussed a challenge he had encountered during training on his journal entries:

> **Teacher G:** *"Starting up Scratch activity on my own without help is a challenge due to lack of computer literacy."*

Few participating teachers had low confidence in their ability to teach CT through programming even after the training. Data from the interviews suggest these low confidence levels will prevent teachers from integrating CT into their lessons without further support.

Teachers' plans for gaining confidence before teaching CT concepts

Some teachers discussed their plans to gain confidence in teaching coding and CT before they taught coding lessons in their classes. Teacher C discussed that she had wanted to feel confident about teaching the concepts in an interview before introducing Scratch in her classes:

> **Teacher C:** ***"****So I just, I just need more of training on how to use, like creating activities, using the computers, like creating scratches. That's where I'm not more confident. I need more training as time goes on."*

In the weekly journal entries, teachers commented that they need more training and to practice more with hands-on activities to get comfortable with Scratch. For example, Teacher J and others said the following:

> **Teacher J:** *"With more training, I am sure I can do these activities."*

> **Teacher K**: *"Yeah, I think if we have, if we just get the computers, I'll be more confident, because even if I'm teaching my learners, I will be able to show them what I want them to do or to know and then that will actually increase my confidence in making my learners believe what I'm telling them is something that I know."*

> **Teacher H:** ***"****I would say maybe, I'm not so confident, I can say maybe we need a little bit more time I wish we, we were having at least even for a full month, we are being trained about these things, just for us to, for the information to be broken down for us, and then maybe we will be presented with some notes and questions, whereby we, yeah, and then, whereby we are just doing practical things."*

> **Teacher F:** ***"****Okay, so be, in order for me to feel comfortable, I only need more training because I'm having that confidence that if, uh, I, I had more time, I can do it perfect."*

The teachers' trust in their abilities to teach CT improved throughout the programme and was typically high by its conclusion. Nonetheless, qualitative data shows that some teachers need further time and practice to obtain the confidence to implement the CT concepts into their lessons.

Teachers' improved confidence about CT after training

In their journal entries, survey responses, and interviews, a few teachers mentioned that their confidence in teaching CT concepts had increased due to their involvement in the programme. Some teachers stated that before the programme, they lacked comprehension of CT and confidence in their ability to teach, but after completing the programme, they felt confident in their ability to teach CT.

Teacher A explained her confidence after the training during the group 1 interview:

> **Teacher A: "***I think I'm very confident in developing, uh, computational thinking capabilities in the students, because from what we learnt, I just realised that we actually already apply computational thinking in our teaching. It's just that we're not aware of it."*
>
> **Teacher I: "***As for me looking at the computational thinking, talking of the concepts, I think I will say more of 50 Confident 50 not so confident. So, because for now, I understand the concept of computational thinking, which I can say we have been using them in our classes, though we cannot really tell which one is which."*
>
> **Teacher J:** *"I'm confident in being able to teach these concepts to my learners."*

Qualitative results suggest that teachers who feel more confident in teaching CT concepts could recognise that some of their topics are already aligned with CT concepts.

Integration of Quantitative and Qualitative Findings

The coherence component of the framework involved examining teachers' confidence in their ability to teach CT. The quantitative and qualitative data were combined to determine where they converged and differed. As indicated in Table 6.6, integrating these observations led to three primary conclusions. Before completing the training, the teachers who participated in the programme had low confidence in their abilities to teach CT. There was an indication in the qualitative data that some teachers lacked confidence in their ability to teach CT after receiving training.

Participation in the programme resulted in a substantial rise in teachers' confidence in teaching CT. The quantitative study of the comparison of teachers' responses to all questions before and after training demonstrated that their self-efficacy grew substantially. In their interviews, some teachers stated that their confidence in teaching CT greatly increased after participating in the programme. According to the qualitative data, some teachers wished to build more confidence in teaching CT before implementing Scratch programming into their lectures.

Table 6.6 Joint findings table for answering coherence related questions

Conclusion from Integration	Quantitative Findings	Qualitative Findings
Before attending the programme, teachers reported poor self-efficacy in teaching CT.	Before the programme, the bulk of the questionnaire items received low responses overall.	In interviews, some teachers said they lacked confidence in teaching coding and CT before the programme.
As a result of engaging in the programme, teachers' confidence in their ability to teach CT grew.	All questionnaire items showed teachers making great progress, with large effect sizes.	In their journal entries and interviews, some teachers claimed that participating in the programme had given them more confidence.
Teachers sought to acquire confidence in teaching CT before implementing coding sessions.		

6.4 Active learning and Collective participation

This section presents the results of the analyses of CT integration during the PD training using the proposed framework. During the interviews, teachers were asked how they integrated CT during the training and how they plan to use the workshop experience in the future.

Qualitative Analysis

During the participatory training workshop, teachers were regarded as design partners and invited to select their subject topics and construct activities and lesson plans incorporating CT concepts and practise.

The inductive qualitative analysis (see Appendix K) revealed two themes relating to the framework's active learning and collective participation components, described in the following sub-sections.

Integrating CT as design partners

The results demonstrated that teachers used algorithms most, followed by decomposition and pattern recognition. Most teachers who designed activities utilised the collaboration and perseverance approaches because they believed that activities were more enjoyable and productive when completed in pairs or groups. Teacher E discussed how they incorporated CT into their Social Studies lesson:

> **Teacher E:** *"Um, the computational skills, like what teacher A said, because we were doing the same subject, social studies, we used algorithm and collaborating. So those are the skills that, um, that we developed for our lesson and the activities also using computational thinking aspects, such as algorithm and decomposition."*

Another teacher from the Natural Science department also added how their group integrated CT concepts into their lesson.

> **Teacher C:** *"Okay. Uh, like in my case, we developed a lesson plan for science whereby the topic was on, uh, ecosystem whereby we used algorithms and, uh, patterns and also logic. Whereby algorithm will allow um, the teacher to make up steps, come up with steps and patterns so that learners will know the difference between the omnivores, the herbivores and the carnivores."*

Teacher A and G also discussed their group's experiences about integrating CT concepts and approaches in their lessons in during the interviews:

> **Teacher A:** *"We used, um, collaborating whereby the learners are going to work together in groups so that they can complete, uh, the task given to them. And also, the persevering whereby they have to continue or to keep going if they did not get the right answer. We also used, um, algorithm, whereby um, they have to make the steps that they have to follow."*

> **Teacher G:** *"Um, nothing much, really, maybe just to explain the, uh, approach that we used in the other lesson plan, which involved the computational thinking, which is the persevering."*

As design partners, teachers indicated that they learnt how to design their lesson plans and activities incorporating CT concepts. The data also showed that teachers appreciated the feedback received from peers as part of a collaboration with others teaching different subjects. Teacher I described the concepts and approach they used when designing their Natural Science lesson plan.

> **Teacher I:** *"For us, we used the pattern recognition, or concept that is enabling us to, to find the similarities and the differences, because we were talking about food chain. And so the learners have to know the differences and similarities between the food web and the food chain, and also use their collaboration, that allows the learners to work in groups."*

Meanwhile, Teacher F explained how her group had to refine their lesson plan based on the feedback from other teachers when presenting their lesson plans.

> **Teacher F**: *"Okay, I'm a language teacher. So the concept that I found fit that I used in my lesson plan was debugging. But then, when we did the review of a lesson plan, my colleagues advise that I use the tinkering method and the collaborating method, which are also a great way because it has also opened my eyes to get the learners more involved or to make it a more learner-centred lesson plan of teaching approach."*

When analysing the interview data, it shows that Social Studies and mathematics teachers used at least two concepts when incorporating CT into their lessons and activities compared to other teachers. Teachers A and M described how their groups integrated multiple CT concepts into their Social Studies lesson plans.

> **Teacher A:** *"Okay, with our lesson plan, we did the lesson plan of social studies; we incorporated the algorithm, making steps and rules into our teaching and decomposition, breaking down the topic into different parts so that the child can understand it better."*

> **Teacher M:** *"We have used decomposition, that is breaking down their lesson into different parts, because we chose a topic that was on import and export,*

and that is a complex topic. So, we have to break it down for the learners to make it easier for them. And we also used algorithm, that is making steps and rules."

The mathematics teacher mentioned how he would use CT concepts in their lesson plans.

> **Teacher N:** *"To incorporate CT in the lessons, for example, like in mathematics where we have problem-solving as a topic. So, here I could use the CT a concept like for example, the algorithm and decomposition concepts."*

Future plans to integrate CT

After training, all teachers agreed that given their school context, they would use the unplugged strategy in their classroom to incorporate CT skills through posters, workbooks, flashcards, etc., as Teacher F and Teacher C explained below how they plan to deliver CT teaching through unplugged activities:

> **Teacher F:** *"Yeah. To add to the strategy, uh, as teacher G said, uh, I just want to add the, uh, unplugged is the best way to use at our school because we don't have the computer lab and the learners themselves don't have knowledge about using the computers yet. So, I think we only use to have the posters, the flashcards, all those, uh, strategies that we can apply instead of having the computer."*

> **Teacher C:** *"Okay. In this case, as, um, at, our school, we don't have enough computers, I would use the unplugged way of, um, creating activities whereby we can use posters, flashcards, and other activities whereby like in mathematics, where learners are to solve or to work out the number patterns, we can use the flashcards, and also because we use, um, the, the methods on how to find the next missing number in the pattern. So here we can use our own, uh, unplugged way. And we can also use, uh, include a lot of approaches, like tinkering, whereby learners have to change things or the formulas."*

Teachers discussed integrating CT approaches into their activities and lessons in the qualitative data. A common approach teachers used for integrating CT was collaboration, so learners work in groups and experience perseverance in the process. Below is the excerpt from Teacher L discussing during the interview how she plans to teach the learners using CT skills.

> **Teacher L**: *"By that, I will have to give learners a problem. So that they will have to work together that is now to collaborate. Because when they are working together, when the learners are working together, it's very easy for them to keep going rather than just one learner, and then she, or he will have to give up because she cannot. She cannot fix the errors on her own or his own."*

One teacher indicated that they would also use Scratch for creating quizzes, and learners can use the few computers available through group work.

> **Teacher C:** *"When I have access to the computers here, we can use the, um, like the Scratch whereby we create activities within the computer, and then we create more activities like quizzes for them to solve."*

Teacher K was optimistic that soon they will get computers at the school and will then use programming with Scratch teaching strategy with the learners too.

> **Teacher K**: *"Okay, since we are in, what's the word? okay, where we find ourselves right now, I will usually use the unplugged method because currently, we do not have a lot of computers at school. And we are actually building a computer lab. So, for now, the best computational thinking strategies that I can integrate into my lesson plan might be unplugged, but obviously in future, I will use the Scratch method, then I think the lab will be done by next year. And then I will be more than happy to pass on the skills to my learners using the Scratch method because we'll be able to have a lot of computers at school."*

Some teachers mentioned that the school management is positive and feel that they will not hesitate to buy more computers if money becomes available. As the school is busy building the computer lab, this has heightened hope that soon they will get the computers.

> **Teacher C:** *"Okay. Looking at my school management and the way I know it, having money won't be a problem for us to purchase computers."*

Overall, the results from the interviews indicate that most teachers can design lesson plans and activities that incorporate algorithms, decomposition, and pattern recognition concepts. They also prefer to apply collaboration to their classroom activities when integrating CT skills. Looking at the data, teachers were uncomfortable integrating concepts such as abstraction and debugging into their lessons, suggesting they did not grasp the concepts and could not align them to their topics. Regarding their future plans of integrating CT, they all agree that the best-suited teaching strategy for their school is unplugged due to the lack of computers at the school.

6.5 Context

When doing a PD programme for teachers, one should consider the context of their school. This study's findings showed that the teachers were aware of their context when choosing the material and teaching strategies for their CT-infused lesson plans. All teachers agreed that they would prefer to use unplugged strategies because of the lack of computers at their school. This agrees with the findings of Sherwood, Fancsali, et al. (2020) that CT integration can take different forms depending on the needs of the teachers or school. In addition, schools where there is no electricity can also use unplugged activities (Brackmann et al., 2017; Rich, Yadav et al., 2020).

> **Teacher N** - *"I just want to add that, uh, unplugged is the best way to use at our school because we don't have that computer lab, and the learners themselves don't have knowledge about the computers yet. So, I think we only use to have the posters, the, the, the, uh, flashcards, all those, uh, strategies that we can apply instead of having the computer."*

> **Teacher G** - *"Looking at the facilities that we have at our school, uh, would say the unplugged will work best because we don't have the computers at school.*

We don't have the library, the computer lab as well. So, um, I, I think the unplugged will work because then the learners will have to work with the pen and paper and worksheets."

Another context that appeared in the findings is that of school leadership. The teachers who mentioned it indicated that their school leadership is positive towards technology and CT. They believe that when money becomes available, they will not hesitate to buy more computers because they are already busy building the computer lab. This is supported by Leonard et al. (2017), Rich et al. (2017) and Sherwood, Yan et al. (2020) that school leaders need to be aware of and involved in CT integration.

Teacher J: *"Looking at my school management and the way I know it, having money won't be a problem for us to purchase computers."*

Teacher F: *"I do trust our management if there are funds available for that they will, they are very; we are very positive. We are a positive team. They will buy because we do everything for the interest of our learners, even though, uh, we don't purchase a computer for each learner."*

6.6 Other findings - Subject Matter Knowledge

During the training, while teachers debated which subject areas to focus on for lesson plan preparation, the Natural Science group brought up the solar system. However, a less experienced teacher with only one year of experience interrupted and stated that she was not yet comfortable with the solar system topic. She was afraid that if they applied CT principles to a lesson plan and had to deliver it to learners, she would *"either fake my way through it or just skim the surface of things."* The teacher then advised that they chose the topic of ecosystems, and the group agreed.

Following this, informal interactions with teachers revealed that a lack of subject expertise would make discovering CT conceptual links to the subject material difficult or impossible. Teachers discussed that they believe teaching entailed building knowledge in their pupils rather than simply completing tasks with them. They should be able to organise a fruitful discussion and answer learners' questions.

But, as one teacher put it, *"I'm one of those individuals who doesn't mind expressing to the learners; I don't know, and I'll find out for you, or let's find out together, but some teachers will never confess to learners that they don't know.*"

These teachers' remarks highlight the importance of subject matter expertise as a precondition for effective integration of CT with the specific subject's topics.

6.7 Conclusion

This chapter presented the study's findings and highlighted the key constructs that significantly affect the effective preparation of teachers to teach CT.

To summarise the main findings, Table 6.7 shows in green what CT concept, practice or learning strategy the teachers have improved on or feel comfortable with, while red indicates less improvement or not comfortable.

Table 6.7 Summary of CT knowledge, subject knowledge, and active learning strategies for each teacher group

Group	Subject	CT knowledge focus	CT thinking practices	Subject curriculum focus	Active learning strategy	Lesson plan
Group 1	Social Studies	Algorithms Decomposition	Collaboration Persevering	Trade (import-export)	Unplugged, Programming	See Figure 5.6
Group 2	English	Algorithms	Collaboration Persevering	English grammar	Programming	See Figure 5.2
Group 3	Natural Science	Algorithms, Pattern recognition	Collaboration	Ecosystem (food chain)	Unplugged, Programming	Append ix: L
Group 4	Mathematics	Algorithms, Decomposition Pattern recognition	Collaboration Persevering Debugging	Sequences	Unplugged	Append ix: L
Group 5	Afrikaans	Debugging	Creating Tinkering Collaboration	Action Words	Programming	Append ix: L

The next chapter will discuss the findings and the components of a PD framework for primary school teachers for teaching CT.

7 CHAPTER 7: DISCUSSION

7.1 Introduction

This chapter discusses the relationship between the case study's findings and the literature. Insights from quantitative and qualitative studies have enabled giving a comprehensive view of issues connected to preparing primary school teachers to integrate CT. This chapter offers a critical narrative of the findings, incorporating viewpoints that arose throughout the investigation, and where applicable, comparing the research findings with those of previous studies.

The chapter is structured according to the components of the framework: six sections based on sub-research questions 1, 2, 3 and 4.

SRQ 1: What is the change in teachers' beliefs and attitudes towards CT resulting from a professional development program in the Namibian context?

SRQ 2: What are the findings from previous research on professional development of teachers to teach CT?

SRQ 3: How can in-service teachers participate in the design of learning material for CT in primary schools?

SRQ 4: What are the components of a framework for the professional development of primary school in-service teachers for teaching CT?

7.2 CT content knowledge

7.2.1 Teachers' understanding of CT

Participation in the training significantly increased the effect of teachers' self-reported comprehension of the taught computational concepts and approaches. Intervention studies in K-6 computer education have typically focused on the effects of interventions on kids' grasp of coding and CT rather than teachers' (Boulden, 2020; Bower et al., 2017). Teachers who adopt curricula incorporating coding and CT must have adequate content knowledge (CK) to teach these concepts. The results reported in the previous chapter confirm Hickmott's (2020) opinion that a CPD programme can have a favourable effect on teachers' comprehension of CT concepts and can thus be utilised to increase K-6 teachers' CK for CT (Hickmott, 2020).

Before the training session, most teachers in this research had little or no knowledge of CT. However, the strategy employed to increase the teachers' CT knowledge contained topics that teachers may relate to everyday life. Similar to existing studies, this has boosted the instructors' CT knowledge following the training (Çakır et al., 2021; Corradini et al., 2017; Mason & Rich, 2019; Uzumcu & Bay, 2020).

Most teachers defined CT as a procedure for problem-solving that can be applied in every scenario, as demonstrated by the findings about their knowledge of what CT is. The teachers' understanding of CT is consistent with definitions from researchers in the literature who all included problem-solving in their definitions and agreed that CT includes at least algorithmic thinking, decomposition, and abstraction (Barr & Stephenson, 2011; CSTA & ISTE, 2011; Selby & Woollard, 2014; Sysło & Kwiatkowska, 2013; Wing, 2006).

7.2.2 Preparation of teachers to teach CT

This study's qualitative analysis revealed that teachers grasped the concepts of CT but found programming/Scratch concepts challenging. There are studies that taught teachers to apply the programming technique and found that their comprehension of fundamental programming concepts was enhanced (Cetin, 2016; Falkner et al., 2018; Gleasman & Kim, 2020; Kong et al., 2020). These studies with pre-service teachers proved the benefit of introducing them to programming activities while they are still in college, as opposed to the teachers in this research who are already in the classroom, and some are unfamiliar with computers.

The results also suggested that following the training, except for a few, teachers related the concepts of CT-enhanced learning to their previous classes, making it easier for them to comprehend and implement, particularly when employing the unplugged technique. This is backed by studies by Ausiku and Matthee (2021), Huang and Looi (2020) and Muñoz del Castillo et al. (2019) that the unplugged method can have a better theoretical basis and allows teachers to integrate CT concepts into any topic without the pressure of mastering a particular technology first. Due to their familiarity with mathematics concepts, teachers who taught mathematics were able to readily integrate several CT concepts into their lessons. The perspectives indicated by the interviewed teachers are comparable to those revealed in research on teachers' incorporation of coding and CT in K–6 Mathematics. According to Rich, Yadav et al. (2019), primary school teachers who participated in their study perceived substantial linkages between CT and Mathematics. This is also corroborated by Nordby et al. (2022) and the Rich, Yadav and Schwarz (2019) results that offering direct teacher preparation through unplugged and presenting CT as a general problem-solving procedure improves teachers' instructions.

Summary

This component is focused on the teachers' understanding of the CT content and their ability to apply it across subjects using different materials. The analysis results have shown that the participating teachers have gained an understanding of what CT means, what the concepts and approaches are and how to integrate them into their lessons which is similar to the findings by Marksbury (2017) and Uzumcu and Bay (2020). All teachers have integrated at least one CT concept in their lesson plans, mainly algorithms, decomposition, and pattern recognition. The integration was done across subjects, such as Mathematics, Social Studies, Natural Science, English and Afrikaans. Teachers modified their lesson plans and activities based on feedback from others which is similar to the study by Kaya, Yesilyurt, et al., (2019).

7.3 Coherence

7.3.1 Teachers' beliefs and attitude towards CT

As a result of the training, teachers' opinions and attitudes about CT have shifted in a positive direction. Comparing before and after responses, the analysis of variables obtained by the questionnaire instrument adapted from Bean et al. (2015) revealed substantial changes in the beliefs and attitudes of teachers regarding CT learning. Similar to findings reported by Zha et al. (2020), teachers exhibited apprehension and scepticism towards CT in the classroom prior to receiving training.

Teachers' views on whether "computing is like art – you are either born with the ability to think that way or not" show that they believe it can be taught. Past studies have indicated that teachers can effectively teach CT skills through detailed workshops (Bower et al., 2017; Curzon, McOwan et al., 2014; Falkner et all, 2018).

The results indicate that after undergoing the CT integration training, teachers' perceptions that CT concepts can be taught at the primary level in any subject and be understood by the learners have positively improved after they have experienced the CT integration training. Recent research on the training of in-service teachers has indicated good effects and changes in teachers' attitudes or beliefs (Boulden et al., 2021; Jocius et al., 2020; Rich, Larsen & Mason, 2020).

7.3.2 Self-Efficacy for Computational Thinking

Participation in the programme substantially increased teachers' self-efficacy about CT. The examination of data acquired via the questionnaire instrument developed by Bean et al. (2015) revealed significant increases in teachers' confidence in their ability to teach CT.

The qualitative findings also show that teachers' self-efficacy improved due to the programme and that their confidence in teaching CT assisted them in constructing CT-infused lesson plans. Teachers' confidence and self-efficacy are a frequently studied issue in the field of computing education, since there is an obvious need to increase teachers' confidence and self-efficacy as a means of assisting teachers in implementing curricula that integrate CT (Bower et al., 2017; Kadirhan et al., 2018).

This study revealed that the self-efficacy of the participating teachers was low before the intervention, in contrast to the findings of surveys with over 100 respondents. For instance, Sentance and Csizmadia (2015) discovered that K-12 teachers from the CAS community in England expressed moderate confidence in their ability to teach the Computing curriculum. In a global study, Rich, Browing et al. (2019) discovered that K-8 teachers were, on average, somewhat confident in their ability to do CT-related activities. Falkner et al. (2018) reported on the self-efficacy of Australian K-12 teachers in teaching and assessing concepts from the Australian DT curriculum. Their findings suggested that teachers were confident in teaching and evaluating these concepts. As revealed by these studies, the respondents' past participation in a CT PD programme or research initiatives may have made them more confident than other teachers. Another explanation could be that all these studies were conducted in developed countries where the respondents might have better digital literacy.

Authors of small-scale intervention studies have often measured self-efficacy as a concept and found that pre-service and in-service teachers' levels of self-efficacy about the teaching of CT were low before the intervention or ranged from low to high after the intervention (Bean et al., 2015; Rich et al., 2017). This study confirms these findings. Studies of self-efficacy in coding and CT that reach bigger and more representative samples of the worldwide teaching community are required so small-scale intervention studies may be compared to the broader teacher population.

A few recent studies have demonstrated that teachers' self-efficacy in teaching coding and CT may be enhanced with a PD intervention, even if the session lasts less than a day (Bean et al., 2015; Bower et al., 2017; Tankiz & Atman Uslu, 2022). This study differs from these cases in that it involves a CPD programme, and the results of the questionnaire instrument give a picture of the self-efficacy views of in-service primary school teachers towards teaching particular computational concepts or strategies. For instance, the findings of my study indicate that following the training, teachers were typically still not confident about teaching computational concepts using programming.

Insufficient knowledge of CT prevents teachers from integrating CT into their classes (Kaya, Yesilyurt, et al., 2019; Kong & Wong, 2017; Sentance & Csizmadia, 2017; Zhang, 2020). The shortage of time, whether in terms of instructional time or the duration of the intervention, has been identified as one obstacle teachers experience when teaching CT (Bower et al., 2017; Kadirhan et al., 2018; McGinnis et al., 2019; Rich et al., 2017). In this study, teachers also highlighted the time constraint in training as the main reason they do not feel comfortable teaching CT yet. They suggested that the intervention be extended and get more time doing hands-on activities to grasp all concepts and increase their self-efficacy in teaching CT skills.

Summary

This component of the framework helps to ensure that the PD of teachers is consistent with the school curriculum, and after the intervention, teachers understand the value of adding CT to their lessons, and their skills and beliefs have changed. The findings of the study indicate that the teachers' value beliefs have positively changed, and they believe that they are more knowledgeable about CT and feel confident in teaching CT concepts, as alluded to by (Sherwood, Fancsali, et al., 2020).

7.4 Collective Participation

7.4.1 Effective teachers' professional development

According to the findings, some teachers prefer more hands-on activities and a longer duration of training before they feel comfortable teaching CT to students. This is consistent with the conceptual framework proposed by Desimone (2009), which indicates that for PD to be effective, teachers must be active participants, and the intervention must be sustained over a longer period.

In addition, the results suggested that teachers' knowledge and opinions have evolved and that there was mutual involvement as they worked in groups on the same subjects. Since the topics aligned with their existing curriculum, they included CT in their lesson plans. This relates to the collective participation and coherence components of the conceptual framework by Desimone (2009).

7.4.2 Teachers participate in the design of learning material

Through collaboration and as design partners, teachers incorporated CT concepts into their teachings, as evidenced by the findings. As design partners, I expected them to pick the lesson's topics and apply the concepts they saw to their classroom setting. This is supported by the literature of Kelter et al. (2021), and Kurt (2021), which demonstrates that co-designing by a constructionist approach influences the learner's comprehension and meaning generation and that small groups achieve quality collaboration, and each member can contribute to the group tasks (Agbo et al., 2021).

Most teachers intended to incorporate CT skills through unplugged activities due to the scarcity of computers at their schools, and the fact that some of their students lacked the computer literacy to undertake plugged activities. In a few studies done in developing countries, the context was shown to be crucial in the selection of teaching tactics when implementing CT concepts into lessons (Ausiku & Matthee, 2021; Espinal et al., 2021; Kong & Wong, 2017; Muñoz del Castillo et al., 2019).

Summary

The participatory training achieved collective participation by grouping teachers per subject taught. Through these groups, teachers collaborated and had control of what to design and create for their classrooms. This process enabled them to share their artefacts with other groups and receive feedback. The results showed that teachers were happy with the feedback as it opened their eyes to other concepts they had not thought about infusing into their lessons. Through this constructionist perspective, sharing the artefacts within the groups is as critical as its creation (Kelter et al., 2021; Kurt, 2021).

7.5 Active Learning

In this case study, teachers were active learners involved in the design process of the instructional material and engaged in discussion with the researcher about the lesson plans' objectives, tasks and teaching strategies rather than being passive recipients of information (Darling-Hammond et al., 2017; Desimone, 2009). Results showed that teachers created genuine teaching material suited for their classrooms, helped by the researcher as facilitators in a constructivist approach, as reported by Kurt (2021) and Loi (2004). During the training, teachers used unplugged and programming teaching strategies to integrate CT concepts into their artefacts. Teachers being design partners during active learning were equal decision-makers on what to create for their classroom (Iversen et al., 2017).

7.6 Duration

The duration must be sufficiently long for a PD intervention to be effective. For this study, the duration of the training was 24 hours long, as constructivist intervention takes significantly more time to carry out (Kenny & Wirth, 2009). The results indicated that the support needed by some teachers who felt they did not feel confident in teaching CT concepts was to get a longer duration of training to comprehend all the concepts and practice (Desimone, 2009).

7.7 Conclusion

This chapter discussed the case study's findings and how they relate to the literature and answer the study's research questions. Findings were discussed according to the components of the proposed framework. The next chapter will discuss the validation of the study by explaining the validation methods used and the validation participants and presents the data analysis of the validation process.

8 CHAPTER 8: VALIDATION OF THE FRAMEWORK

8.1 Introduction

This chapter describes the validation procedure and validation findings for the PD4PCT framework provided in Chapter 3. The PD4PCT framework is linked with the study's results and proposed ways to enhance the professional development of CT teachers at the primary level. It is essential that the PD4PCT framework be tested before its dissemination. The validation approach aims to assess the validity and applicability of the PD4PCT framework components and verify the dependability of the recommendations.

The sections that follow analyse validation and the validation technique utilised in this study. Finally, a summary of the feedback, its inclusion, and the ensuing improvement of the PD4PCT framework is provided.

8.2 The concept of validation

Any framework creation process should always include validation to assure the worth and dependability of the model. The integrity of the study findings is confirmed by validation and the accuracy of the research's targeted measurements (Golafshani, 2003).

Because qualitative research lacks established processes connected to validation, it is more difficult to establish validity in qualitative research investigations than in quantitative studies (Creswell & Miller, 2000; Golafshani, 2003). Due to the significance of confirming the calibre of research, the literature recommends several ways qualitative researchers may employ to establish credibility in their investigations (Morse et al., 2002; Golafshani, 2003). In qualitative investigations, the emphasis on validation is on ensuring reliable, valid research (Maxwell, 1992). A research study can be validated by presenting its findings in a form that appropriately represents the phenomena it is trying to understand (Golafshani, 2003).

8.3 Method of validation

The researcher utilised an online questionnaire since it was more convenient for the participants and cost-effective because it was challenging to gather all the reviewers in one place simultaneously.

The participants were cordially asked to help with the validation process through a telephone call followed by an email. A PowerPoint presentation describing the suggested framework and a link to the online survey were emailed to the reviewer once they confirmed their participation.

8.4 Validation Presentation

A PowerPoint presentation (Appendix M) was prepared for the reviewers to give the background of the study, an overview of CT, and explain the framework, so they could answer the open-ended questions in the questionnaire.

Slide 1: Gives the research topic

Slide 2: Provides the background of the study

Slide 3: Provides the aim of the study

Slide 4: Provides what Computational Thinking is

Slide 5: Provides an overview of CT Concepts

Slide 6: Provides an overview of CT Practices

Slide 7: Provides a section heading for the framework's components

Slides 8 - 14: Provides a detailed view of the proposed framework's components

Slide 15: Shows the graphical representation of the proposed framework

8.5 Online Questionnaire

To get the reviewers' views, the following open-ended questions were posted on the online questionnaire using Google Forms (see Appendix N) for the reviewers to answer after watching the presentation.

1. How applicable is the framework to the current Namibian curriculum?
2. How do you see the framework being applied to the teachers' professional development?
3. Do you find any gaps in the framework?
4. Do you have any recommendations to improve the framework?

8.6 Expert Feedback

This section discusses the feedback received from the reviewers whose profiles are described in Table 4.7. The responses from the online questionnaires were thematically analysed. The themes are discussed below:

Applicability of the framework to the Namibian context

The reviewers' responses indicate that they all believe the framework is suitable for the existing curriculum in Namibia. They also believe that the study is pertinent and topical and that if the findings are shared with the country's education ministry, they can affect change in our education system. Below are some of the experts' responses.

> ***R1:*** *"Since this is intended for the primary grades, I believe it can be implemented appropriately. Despite the absence of a computing subject at the primary grades. It is essential to begin with student interests. It depends on where students' interests lie. It involves assisting them in connecting CT to problems in which they are interested. Local relevance is necessary for success of CT in our schools. We utilise technology daily, and even young children are able to use cell phones and computers; thus, I believe it is a topic that must be taught in our classrooms, as the technological landscape is changing so rapidly."*

> ***R5:*** *"I am thrilled to learn that this research is being conducted in Namibia. This should be incorporated into the curriculum so that students are introduced to computer principles prior to high school. Therefore, I believe that allowing students to deal with real-world problems and adopting an interdisciplinary approach are crucial. It is essential that students understand Computer Science and have opportunity to apply their knowledge to intriguing multidisciplinary issues, which frequently include group work and thus develop collaboration*

skills, one of the CT abilities. Certainly, CT may be integrated into our present primary subjects; all that is required is training for teachers on how to teach it."

The role of technology in society

The experts also emphasised the importance of technology today and urged the incorporation of CT into the education of existing primary topics. Learner-centred techniques were also recommended since they help increase the efficiency of learning.

> ***R3:*** *"Considering the incorporation of CT into basic education is the correct approach. CT should serve just as a tool. Instead of everyone doing PowerPoint presentations, everyone should be developing computer models of various things. We live in a technologically dominant society, and if we do not teach our youth problem-solving abilities, we will have a problem on our hands. Our education system is in need of reform, and this study presents an excellent chance to implement CT in our classrooms."*

> ***R4:*** *"As computers permeate every aspect of human endeavour and have become indispensable to all of our lives, computational thinking is essential for everyone to not only comprehend what computers are capable of, but also to utilise them to solve issues. I concur that computational thinking is a fundamental skill that should be taught to all students and should be included in the curriculum for all Namibian kids. The framework allows for the integration of CT into any topic utilising any teaching method. This makes it easily relevant to our existing curriculum, despite the fact that many of our schools lack computers and Computer Science teachers."*

Application of the framework to professional development

One possible challenge to implementing this framework is gaining the support of teachers and school authorities. To promote buy-in, one expert expressed that we need to persuade these stakeholders by helping them comprehend the democratic importance of CT and see the need to incorporate CT into teacher training programmes.

> ***R1:*** *"Integrating technology into the teacher development programme will provide obstacles. You must gain the support of your colleagues in teacher preparation and teacher educator training so that they are conversant with the concepts of CT learning. The importance of considering CT education in teacher preparation is due to its democratisation. Certainly, some dedicated individuals with a solid computer background and a solid set of abilities may achieve the necessary progress on their own. There are likely many valuable resources available that make it feasible. However, without a set of frameworks and some knowledge of how to learn, CT will elude the majority of teachers. Therefore, the framework should be used for the design of professional development programmes in order for them to be effective, particularly given that this is a new field and there are few teachers who are computer specialists. It is a useful resource for trainers. Trainers and teachers should collaborate, and the framework allows for this."*

Another point made by the expert regarding the teachers' PD was the shortage of trained teacher educators with the ability to integrate CT. The experts advocated for collaboration between schools and the industry experts to solve this difficulty. The responses also indicate that access to digital tools and curricular resources, is a barrier for teachers and PD providers.

> ***R5:*** *"Yes, teacher professional development in Computer Science is a problem since there are insufficient Computer Science teachers in many nations, including Namibia. Teachers should be encouraged to retrain by providing them with time and incentives, and they should have access to high-quality learning and teaching tools. We should encourage teachers to network so they may exchange ideas and knowledge because we are all still learning about the most effective ways to teach CT Concepts. From my experience teaching teachers how to code using Scratch, I can confidently say that the framework can be used for professional development programmes for CT. Even teachers who didn't have any computing skills before, they leave training with some CT skills and a positive attitude towards CT."*

R5: *"Teacher preparation programmes should collaborate with Computer Science departments to offer faculty resources for pre-service teachers. It is nearly ideal since it facilitates the combination of persons with extensive understanding of CS material with others who are genuinely concerned with pedagogy. These collaborations combine the experience of both parties, ensuring that teachers acquire both the technical and pedagogical knowledge they require. I believe that partnerships may be highly beneficial and healthy. And for the training of in-service teachers, the ministry can collaborate with industry experts to teach CT ideas."*

R2: *"As trainers of teachers, we should first receive expert-level training in CT in order to develop teachers' knowledge. In addition, it is easily applicable to any subject. To make things simpler; however, it should be required that teachers in schools who lack basic computer skills receive computer literacy training first. As a subject matter expert, I can use this framework to create training programmes for my subject teachers where we can learn how to use CT abilities, possibly with the assistance of an IT expert."*

Funding and lack of computing skills among teachers were discussed in terms of implementing the framework. One expert reviewer eluded that the funding element should be considered even as part of implementation guidelines. Another expert proposed that pre-service teachers should take a course on CT to prepare them while they are still in college. Below are some comments from the reviewers.

R3: *"The majority of schools lack the means to build a computer lab, let alone employ a Computer Science teacher. Therefore, the government must give financing for some training programmes, particularly where an expert is required to provide the training. They may also assist schools in establishing fully functional computer laboratories."*

R1: *"Many in-service teachers are reluctant to embrace technology or are scared by it, but if you can reach pre-service teachers while they are still in university as part of their training, I believe they will be far more receptive to it. The majority of pre-service teachers are obliged to attend a computer literacy course; how much more effective would it be if all teachers were compelled to* *take this course? What if we made it truly relevant to what pre-service teachers are doing in their classrooms and showed them how to incorporate CT into the context of their own studies?"*

Relevance of the framework

One expert who is the head of the PD division in the region mentioned poor professional development programmes conducted in the past for teachers and how this framework can design effective training programmes for teachers.

> ***R4:*** *"Poorly implemented professional development has, in my experience, overwhelmed teachers. To me, a symptom of inadequate professional growth is how the teacher is at the conclusion of the process as opposed to the beginning. I've witnessed professional development that overwhelmed participants and left them more nervous than when they arrived. Typically, these are quite content-heavy and offer less opportunities for conversation. If this framework was applied to a training programme, it would provide scaffolding for teachers who are all new to the CT skills, and the active learning component would allow for discussion and question-asking. As the regional head of professional development for teachers, I endorse this approach and can envision it being used to train all our teachers in the region, beginning with senior education officers who train in-service teachers."*

The experts did not find any gaps in the framework, with one citing that since CT is new to her, she could not tell if there was a gap. Other experts remarked that the framework is useful and applicable to training programmes.

> ***R2:*** *"I find all components relevant to the professional development of the teachers. As I prepare my professional development interventions for my teachers, I would definitely borrow some elements from here even if it will be for training on humanities subjects."*

> ***R4:*** *"The framework addresses all important elements of the professional development process, some of which I haven't thought about before like participatory design. I am glad that using subject matter experts should be used*

during intervention. We have recently started using this for the newly introduced AS Level curriculum, and it helped the teachers a lot."

R1: *"I don't see any gaps at the moment since this is new to me. I think it is a very relevant study that can bring about change in our education system if shared with top people in our ministry."*

R4: *"By anchoring students' learning in issues that need to be solved, CT offers teachers with an additional method for engaging their pupils. Consequently, teaching CT is both a pedagogical activity, and an engagement practise. Ideally, all teachers would be required to attend a course on the teaching of CT."*

These reviewers' comments have validated the framework by emphasising that the framework was a good model, illustrative of what was practical and useful in emphasising components that may be considered to effectively construct CT PD programmes for teachers in Namibia.

8.7 Revised PD4PCT Framework

The framework was revised after considering the implementation experience mentioned in Chapter 5 and the experts' evaluation results. This led to the addition of a new component, "*Integration levels.*" As a result of the informal discussions with teachers that transpired during training, a lack of subject expertise can hamper the integration of CT into some topics (see section 6.5). This research identifies various likely reasons teachers could steer clear of certain topics. Teachers typically lack the confidence to instruct something unfamiliar when they have little topic expertise (Koehler et al., 2013; Shulman, 1986). The teachers in this research are similar to other primary school teachers worldwide because they are generalists and lack specialised training, in particular areas. The validation questionnaire's implementation experience and expert comments also alluded to expert involvement.

During the validation phase, experts suggested that computer literacy training is offered to teachers before a CT intervention. This is because not all teachers are computer literate; eventually, they will need to use computing devices to enhance their CT lessons.

The element of subject experts is part of Jocius et al. (2020) 3Cs Code and Create components, where they had CS and core subject experts as facilitators during the intervention. This element was initially ignored in the research design of this PhD thesis, as it was assumed the teachers have adequate subject knowledge and that since they work in groups per subject, the collective expertise would suffice. Following the informal discussions with the teachers and the feedback from reviewers, it is clear now that there is a need to involve subject experts in the interventions. This resulted in its inclusion in the revised framework.

Apart from subject knowledge and expertise, other implementation recommendations were developed because of the framework's implementation experience. The recommendations are:

1) Get an understanding of teachers' subject knowledge using a subject expert.
2) Get a subject matter expert to identify topics suitable for CT integration beforehand.
3) Explain fundamental CT concepts, linking them to specific subjects with examples.
4) Increase the duration of the intervention to over 24 hours.
5) Provide opportunities for further training after the intervention.
6) Offer initial computer literacy to teachers before or at the start of the intervention.

Each of these is elaborated on below:

Getting an understanding of teachers' subject knowledge

It became evident throughout implementation it was necessary to evaluate the teachers' prior understanding of the chosen topic of their subject. It is only natural that teachers learn to design pedagogies for a subject after they grasp the subject matter, claims Zhang (2020), who researched the integration of CT into Swedish schools using block-based programming. When teaching CT through robot programming, Çakıroğlu and Kiliç (2020) proposed a course model for evaluating teachers' pedagogical content knowledge. They discovered that the expected integration is challenging regardless of how useful the curriculum or content is if teachers are not prepared to teach the content. Similarly, Yadav et al. (2017) pointed out that teacher preparation for CT involves two dimensions in their paper “CT for Teacher Education.” First, teachers-in-training should have better basic CT content knowledge (CK). To integrate their CK into teaching CT, PK should be acquired.

Subject matter expert to identify topics that are suitable for CT integration

To avoid forcing teachers to teach unfamiliar topics, it is best to enlist the assistance of the subject expert when choosing the subject for CT integration. Through informal discussions, teachers stated during the training they must have a solid grasp of the selected topic to feel comfortable teaching it; it is necessary to pick an appropriate subject. Because the subject field might be expansive and teachers may lack the essential subject expertise to comprehend a topic, specific guidance should be sought when selecting a suitable topic (Cizadlo, 2018). Therefore, I have included the implementation recommendation to seek assistance from subject matter experts regarding topic selection for CT integration.

Explanations of fundamental CT concepts with examples

The teachers expressed a desire to grasp the concepts we were studying more thoroughly, even at the price of covering more topics, which led to the addition of the implementation advice to concentrate on deeper comprehension.

The recommendation to explain fundamental concepts with examples from their subject is introduced to the framework implementation because of this experience. Every day’s examples of the subject matter and its linkages to CT should be provided. The quote below shows that a teacher was unaware of the CT concepts before they were explained using a simple example she can relate to.

Teacher D: *“I didn’t know that all along; I have been using these concepts.”*

Duration

The component of duration is revised to say that the intervention should be longer than 24 hours and dedicate more time to hands-on practice. The 24 hours done in this study could be used solely for practice before the teachers can be asked to design their lesson plans and activities. The teachers indicated in the interviews and journal entries they needed more time as the 24 hours for this intervention left some not confident enough to integrate CT into their lessons.

Teacher L: *"I can say maybe we need a little bit more time. I wish we were having at least even for a full month; we are being trained about these things."*

Teacher N: *"I will need more maybe training or hands-on so that I can practice and just grasp all the information that I need in everything that I need on this computational thinking."*

Teacher H: *"With more hands-on training, I am sure I can do these activities on my own and apply them to teaching my learners."*

Teacher B: *"We should engage in more practical activities. It makes us understand the session better."*

Further training

Another implementation advice is to provide opportunities for the training to occur over multiple months. The training should be continuous and not just a once-off.

Computer literacy training

One recommendation from the expert reviewers is that not all teachers are computer literate. Even though unplugged exercises were used, some teachers used programming to teach CT. As they will eventually need to use technology to integrate CT into their classrooms, to train them beforehand on how to use computers and introducing them to coding environments is necessary.

Integration levels

During the implementation of the framework, some teachers mentioned they did not realise that they had been using some of the CT concepts all along in their classes. So, it became evident that it is necessary to consider the existence of CT in the curriculum and incorporate it into the framework. Below are what two teachers said during the interviews.

Teacher B: *"But it's just that we did not know; I did not know what it was called. But I was using this method more in the language by letting learners find out things that are wrong, and then they have to correct it, only to come to learn it's called debugging. And I was like, wow, okay."*

> ***Teacher N:*** *"After the workshop, looking at the CT concepts with the logic, evaluation, algorithm and patterns and decomposition and abstraction, looking at these things, we are already using these concepts. So, if you look at logic in mathematics, we do the estimations, and then after that we have to analyse our answers, whether the answers are correct. And then with evaluation, we do these things in mathematics. And then I think we are doing these concepts in our lessons without us knowing."*

The suggestion is to build on the three CT integration levels of *Exist, Enhance* and *Extend* by Waterman et al. (2019). These levels will provide trainers and teachers with a strategy for identifying possibilities for incorporating CT. It will also assist in identifying and highlighting parallels between concepts and practices in subjects and CT and encourages us to explore and create untapped prospects for a deeper dive into CT.

First level: Identifying the CT practices and concepts that are already covered in the existing curriculum

Some lessons already include CT concepts, skills, and practise; they only need to be pointed out. Teachers can give examples of how these CT possibilities can connect to computers or other technology, even if they don't directly involve technology. As it helps teachers realise that they were already, in part, building CT while teaching their subjects, we thought it was crucial to identify this initial level.

Second Level: Adding lessons or activities to deepen the connections between subjects and CT concepts

To deepen the connections between the subjects and CT concepts, new activities and lessons can be developed to provide a direct relationship to computing concepts mentioned in the lesson but are not its main focus.

Third Level: Integrating activities connected to CT to enhance disciplinary knowledge

As a foundation for CS inquiry, new lessons or sequences of lessons that expand the discipline notion are likely to involve programming exercises. Teachers must be computer literate as they may need to use programming to enhance lessons.

Revised graphic representation

The revised framework is provided in Figure 8.1. Based on the implementation experience and recommendations from the validation phase, it now includes one new component. The component of *Integration Levels* deals with identifying CT concepts in existing curricula and then enhancing and extending them by developing new lessons with the help of subject matter experts.

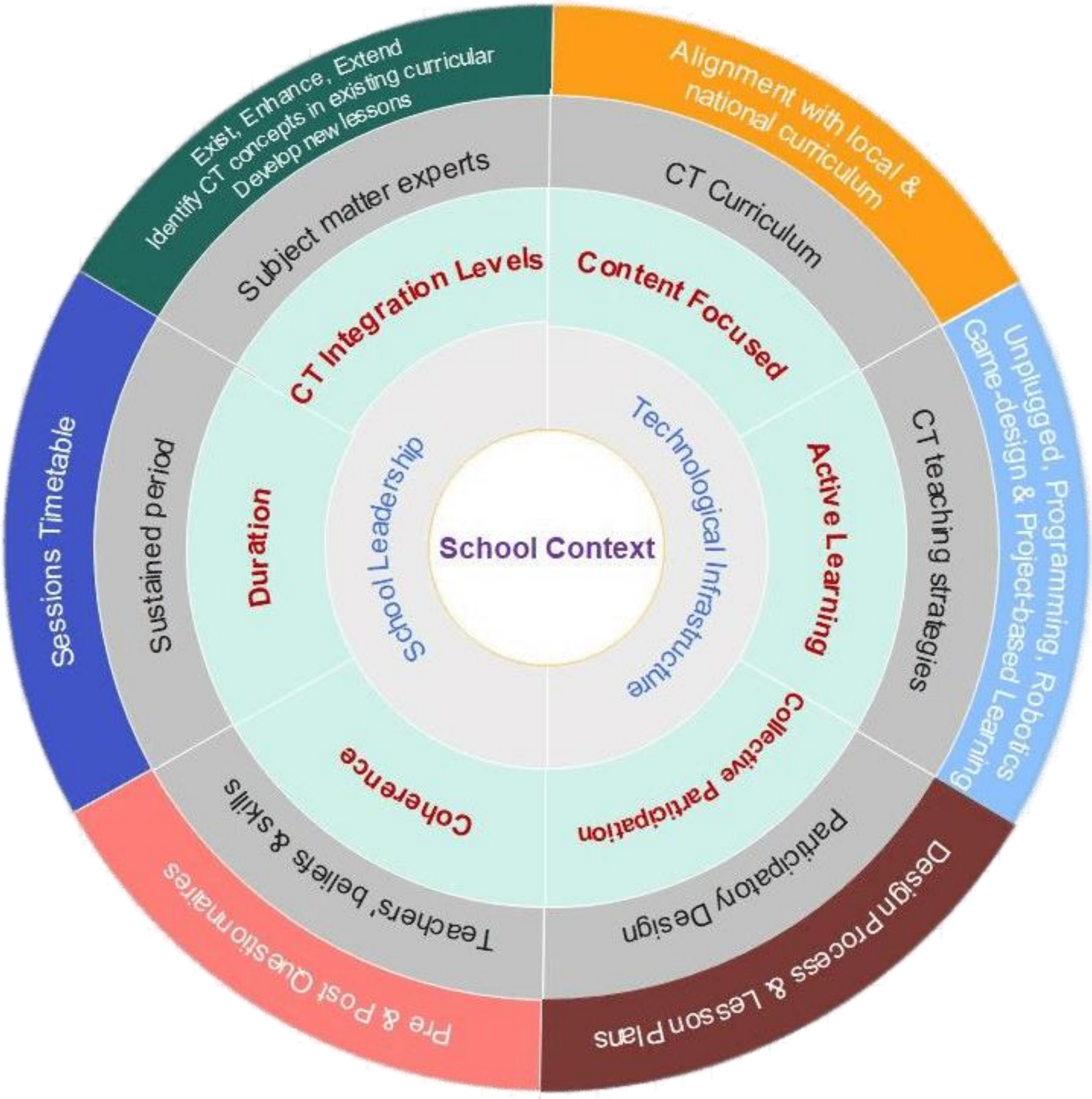


Figure 8.1 Revised PD4PCT Framework

8.8 Guidelines for the implementation of the framework

The PD4PCT framework is a guideline on what components to consider when designing a teacher's professional development programme for teaching CT in primary schools. Implementing this framework may differ based on context, as this study was based on a developing country context.

1. *Step 1* - Understand the school context in terms of technological infrastructure and school leadership. Table 3.4 provides valuable guidelines regarding the school context, infrastructure, and appropriate technology.
2. *Step 2* - Determine what tools & teaching strategies are available and suitable for the identified school context. Table 3.5 provides the tools and techniques developed for implementing the framework, while section 3.2.1 list the strategies that can teach CT.
3. *Step 3* - Plan the participatory intervention – how long will the sessions be, and what content will be covered? According to the findings, 24 contact hours were insufficient for teachers to comprehend all CT concepts and do hands-on activities. Table 5.1 summarises the implementation of the participatory design process where decisions on the groups, topic selection, etc., are made.
4. *Step 4* - Introduce teachers to the CT tools, e.g., the Scratch programming environment. The findings show that some teachers are not confident in using Scratch to teach CT. Hence, they should become familiar with the tools first before attempting to create activities.
5. *Step 5* - Introduce teachers to the CT concepts and approaches. This can be done using visual teaching aids, such as the CT poster shown in Figure 5.3 and through PowerPoint presentation.
6. *Step 6* - Introduce teachers to the CT teaching strategies (unplugged, programming, etc.,) which are listed under section 3.2.1
7. *Step 7* - Use a subject matter expert to help with topic choices for the CT integration and assess the teachers' subject matter understanding.
8. *Step 8* – Use the three integration levels to identify CT concepts in the teachers' existing lessons for enhancing and extending as necessary.
9. *Step 9* - Guide teachers through hands-on activities to practice the application of CT concepts.

10. *Step 10* - Let teachers collaborate through participatory design to infuse CT concepts into their lessons and activities, helped by a subject expert.
11. *Step 11* - Finally, teachers should peer review the produced lesson plans and activities and refine them as needed.

Conclusion

The chapter looked at the framework's validation by using experts to review it through an online questionnaire with open-ended questions. The framework was revised due to the findings and expert views, where two new components of subject matter understanding, and integration levels were added. New recommendations for implementation were also given on the duration component and training as an outcome of the findings. The next chapter concludes the study and makes recommendations for future research.

9 CHAPTER 9: CONCLUSIONS AND RECOMMENDATIONS

9.1 Introduction

This chapter discusses the responses to the research questions explaining the study's conclusions and recommendations. The chapter then describes the researcher's practical and theoretical contributions and concludes with suggestions for further research.

This study's major objective was to investigate how in-service primary school teachers in developing nations may be prepared and encouraged to incorporate CT concepts into the current curriculum. The Professional Development for Primary school teachers for CT framework (PD4PCT) was accomplished by developing a framework that training providers and researchers may use to design effective CT teacher professional development courses.

To accomplish the goals of this study, an interpretative case study was conducted with 14 primary school teachers from a single school in Windhoek, Namibia. A literature review, questionnaires, journals, and semi-structured interviews with teachers of various disciplines were used to collect data. Constructionism served as the theoretical, pedagogical foundation, whereas the existing PD conceptual frameworks of Desimone (2009), Kirwan et al. (2022), Kong and Lai (2022) and Jocius et al. (2020) were adapted to inform the suggested framework. The principles of participatory design were incorporated into the suggested framework. To achieve this, the following primary research question and four sub-research questions were posed:

Main research question:

How should in-service teachers be prepared to teach CT in primary schools in a developing country context?

Sub-research questions:

> **SRQ 1:** What is the change in teachers' beliefs and attitudes towards CT resulting from a professional development program in the Namibian context?

SRQ 2: What are the findings from previous research on professional development of teachers to teach CT?

SRQ 3: How can in-service teachers participate in the design of learning material for CT in primary schools?

SRQ 4: What are the components of a framework for the professional development of primary school in-service teachers for teaching CT?

9.2 Conclusions to the research questions

These sections present the closing remarks for each sub-research question and demonstrate how they all contributed to address the primary research question.

SRQ 1: What is the change in teachers' beliefs and attitudes towards CT resulting from a professional development program in the Namibian context?

The results of this study's questionnaires and the analysed literature indicate that teachers have various attitudes and views about CT. As most studies incorporated CT into science and mathematics, it was discovered in Chapter 2 (section 2.10) that teachers could draw connections between their maths and science lessons and perceive CT as a problem-solving technique, primarily equating it to algorithmic reasoning. It was also determined that the current emphasis on programming as a way of teaching CT might deter teachers from schools without access to computers or computing experience, influencing their attitudes towards CT. The following issues were identified as influencing teachers' attitudes and beliefs about CT (section 2.11):

- Pedagogical and content knowledge **-** Inadequate understanding of the subject matter may also have major repercussions since academic content is the foundation for teaching and learning. For instance, when a teacher cannot comfortably teach a topic, this would prohibit them from facilitating CT learning.

- Technological difficulties – when teachers cannot enhance their skills because of a lack of computing devices and technical help in their classrooms, it leaves them with negative views about CT.

- Lack of shared vision – lack of collaboration and communication among teaching staff, administrative staff and school leaders causes teachers to feel discouraged about CT integration.

- Inadequate time for learning & teaching – packed school timetables and calendars leave teachers with no time to learn and teach CT skills, leading to them not feeling positive about CT.

Results from pre-post-questionnaires showed that teachers' beliefs and attitudes improved because of the training (section 6.3.1). They believed that CT should be taught in primary schools since their students could comprehend the material. The findings also showed that the teachers believed that everyone can learn computing skills, and that CT is a crucial 21st-century skill set.

SRQ 2: What are the findings from previous research on professional development of teachers to teach CT?

The definition of what qualifies as PD is broad in the literature. Chapter 2, section 2.13 describes the many methods are used for teacher development. It presented the conceptual framework Desimone (2009) developed, which consists of the essential elements of context, duration, collective participation, coherence, and content focus. The TPACK, which includes components of content knowledge (CK), PK, and pedagogical content knowledge, is another framework used to guide teachers' professional development (PCK). To build an educational technology course for teachers focusing on CT integration and integrating technological tools into the classroom, TPACK was employed as a framework for teaching CT by Kong and Lai (2022).

Using PD to get teachers ready to teach CT from a constructionism perspective was discussed in section 2.13.8.

The literature described how teachers were trained to teach CT. Chapter 2, section 2.9, presents this. Teachers must get thorough training on how to develop CT learning exercises, how to teach CT and how to use technology to teach CT ideas if CT teaching and learning are to be implemented in schools. The literature claims that professional development interventions were the main way teachers were exposed to CT knowledge. PD programmes have been held to effectively prepare both pre-service and in-service teachers, and teacher educators have included CT instruction in their pre-service curricula. While some academics have created a PD activity for primary teachers to integrate CT into certain subjects like science and mathematics, others have urged that teachers should see CT as a foundational and cross-curricular skill.

The research revealed various attempts to build methods for CT integration into teachers' PD (see section 2.9.2). Existing frameworks to guide the teacher's PD on CT have been discussed in section 2.13.3, to include CT into teachers' PD, these particular strategies were used:

- Unplugged – this strategy does not involve computing devices.
- Programming – this strategy was the most used and used a basic or block-based programming language.
- Robotics – This strategy was the second most used and used robotic kits and coding.
- Project-based learning – this strategy, the learning is organised around tasks.
- Game-based learning – this strategy uses game creation and generates models and representations.

SRQ 3: How can in-service teachers participate in the design of learning material for CT in primary schools?

A participatory design workshop was held to teach teachers how to incorporate CT in their classes as part of implementing the PD4PCT framework in Chapter 5. Throughout the training programme, teachers were regarded as design partners. The results (section 6.4) showed that teachers developed lesson plans and activities using CT. It was discovered that most teachers used algorithms, decomposition, and pattern recognition concepts in their lesson plans. The results indicated that collaboration and perseverance were the CT approaches that were most frequently used during the participatory workshop. The findings also showed that teachers learnt about incorporating CT into their lessons and valued the feedback given during collaboration as design partners. According to the findings, teachers felt uncomfortable utilising programming as a teaching strategy and incorporating concepts like abstraction and debugging.

SRQ 4: What are the components of a framework for the professional development of primary school in-service teachers for teaching CT?

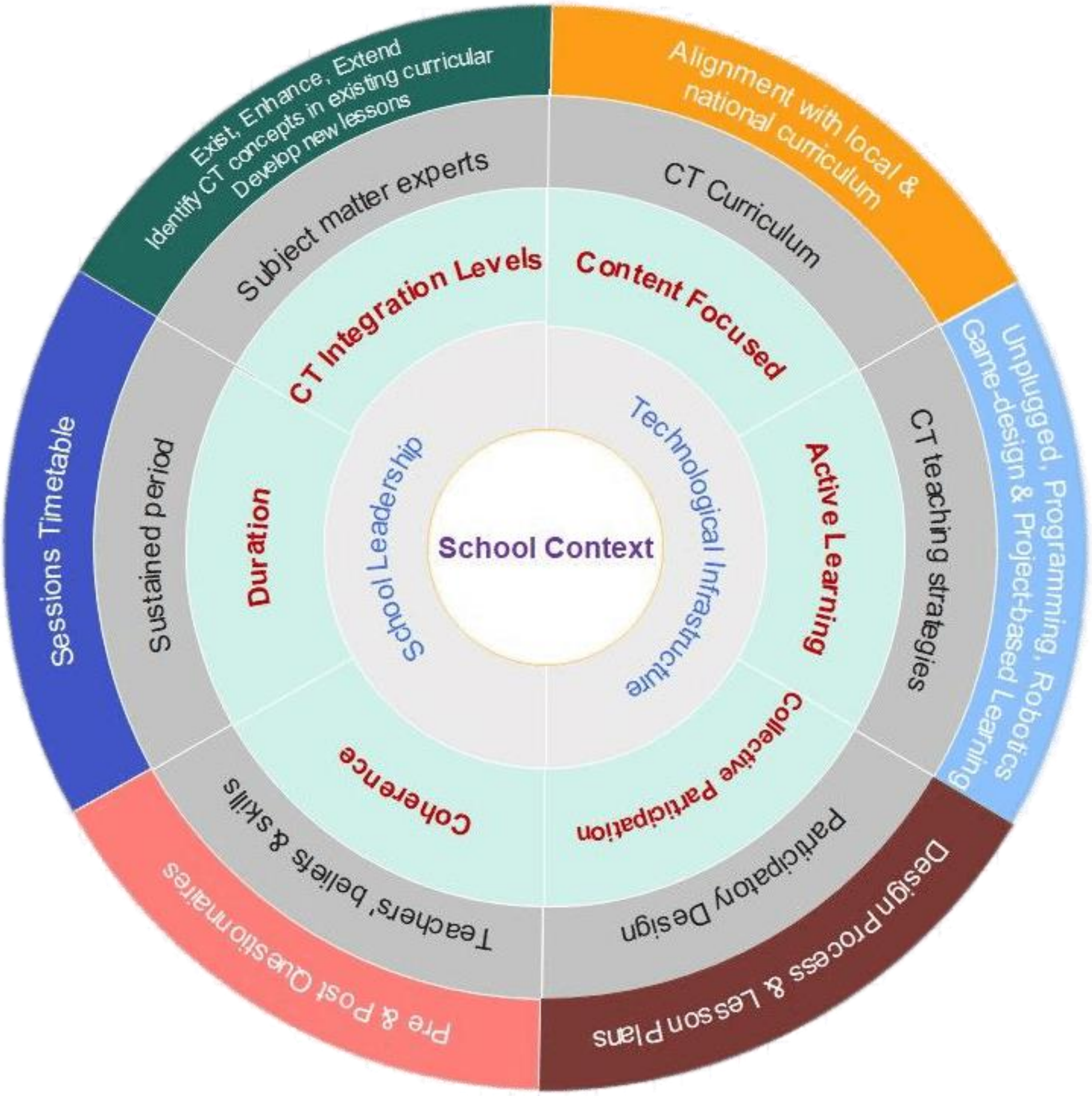


Figure 9.1 The PD4PCT Framework

The PD4PCT framework shown above in Figure 9.1 was developed consisting of six components that each has attributes and implementation advice, as shown in Table 9.1 below.

Table 9.1 Components of the professional development framework

Component	Attribute	Implementation advise and instruments
Content Focus	CT Curriculum	Alignment with local and national curriculum
Active Learning	CT teaching Strategies	Unplugged, programming, robotics, game-based and project-based learning
Collective Participation	Participatory design	Design process and lesson plans
Coherence	Teachers' beliefs & skills	Pre-post-questionnaires to determine beliefs and skills before and after the intervention
Duration	Sustained period	Sessions' timetable
CT Integration Levels	Use subject matter experts, Identify CT concepts in existing curricula, Develop new lessons	Exist, Enhance, Extend

These can be applied by researchers or training providers wanting to conduct effective PD for CT teachers in primary schools. A more comprehensive discussion is provided in Chapters 3 and 5.

Main RQ: How should in-service teachers be prepared to teach computational thinking in primary schools in a developing country context?

The analysis and literature review findings were combined to provide a triangulated response to the main research question. The study's major goal is to create and put into practice a PD framework that may help teachers incorporate CT into already-existing curricula at the primary level.

The main research topic is addressed using the framework proposed in Chapter 3 and revised in Chapter 8 (section 8.8). It applies the components that can contribute to the effective PD of primary school CT teachers. The framework was developed using the literature, refined after being used in a participatory workshop with teachers and validated by expert reviewers.

The results showed that while developing a PD programme or intervention for teachers, the school context of the teachers should be considered first. This element will establish the tone for the intervention's tactics and strategies. It was also revealed that teachers' beliefs and attitudes should be known before implementing training. Teachers can enhance their CT knowledge through various instructional strategies by participating in participatory design workshops. Another result is that to produce meaningful artefacts and work together; teachers should be considered as design partners or co-creators of their instructional materials. The lack of confidence in teaching CT, which is mostly brought on by less time spent on training, is one element that might hinder the integration of CT, according to the findings. Thus, it is crucial to carry out the training over a continuous longer period. It was also discovered that a lack of subject matter knowledge could impede the integration of CT into certain topics, hence the importance of using subject matter experts during an intervention.

9.3 Evaluation of the research

Care in reporting, which is both a matter of argumentation style and correctness of methodologies utilised, is important to build credibility and validity of a research study for an audience (Walsham, 1995). It is now a standard procedure for interpretative field studies to provide evidence of their adherence to pre-established standards for judging their work (Pawlowski & Robey, 2004).

This study is evaluated using the criteria by Klein and Myers (1999) as shown in Table 9.2 below.

Table 9.2 Evaluation Criteria

Klein & Myers Criteria	**How the principle was applied in this study**
The fundamental meta-principle is the hermeneutic circle: suggests that human understanding is achieved by iterating between the interdependent meaning of the parts and the whole they form.	Data analysis of the literature, questionnaires, journal entries and interview transcripts proceeded iteratively between examinations of data and the development of theoretical interpretations. In this hermeneutic process, the analysis of respondents' statements represented the parts, while the evolving conceptual framework represented the whole.
The principle of contextualisation: To make sense, the interpretation requires a historical and social context.	FCR Primary School provided a theoretically relevant organisational context for studying CT skills. A review of the existing literature to get insights into what CT is and how teachers have been prepared to teach CT in the past enabled the development of the initial framework. The semi-structured interviews and weekly journals that followed after and during the initial framework implementation enabled the evaluation of the framework in the current context to identify any shortcomings.
The principle of interaction between the subjects and the researchers: suggests there should be a critical reflection of how research materials were socially constructed through the interaction between researchers and participants.	The participants answered the pre-post questionnaire during the study. The PD intervention was participatory, where the researcher and participants interacted. At the end of the workshop, participants' semi-structured interviews were conducted to get the participants to interpret events.
The principle of abstraction and generalisation: recommends that researchers relate the idiographic details revealed by the data interpretation to theoretical and more general level concepts.	The questionnaires and interviews had specific questions to understand the entire study. The results from these resulted in themes done by abstracting my interpretation of the study. It was argued from the particular to the general.

The principle of dialogical reasoning: cautions there should be sensitivity to possible contradictions between the theoretical framework guiding the research and the actual findings.	Prior research literature informed the analysis, but a variety of other theoretical interpretations were considered as the analysis continued. Dialogical reasoning facilitated the emergence of new insights, which eventually merged into coherent themes linked with prior theoretical frameworks. These themes comprised the elements of the framework that resulted from the analysis. There were some contradictions, such as the assumption that teachers were subject experts, but the findings say otherwise. Another one is that although literature indicates that CT can be applied through unplugged computing, findings from the validation survey imply that technology or coding is an integral part of teaching CT.
The principle of multiple interpretations: underscores the sensitivity to possible differences in interpretations of the events under study among the participants.	This research emphasises the subjective understandings of one primary group of participants (teachers). Although they had similar views, there were a few divergent views because of the different subjects taught and level of experience, which is not surprising, which was considered.
The principle of suspicion: the researcher should also be sensible to possible biases and distortions in the narratives collected from the participants.	As the results show, most respondents provided a favourable portrait of their activities of learning CT skills. The resulting impression is that the teachers found the intervention to have affected their CT skills, beliefs and attitudes. My suspicion was that the teachers would say, for example, they knew of CT before the intervention and said the opposite post-intervention. This was partially mitigated by carefully examining the interview transcripts, journal entries and pre-post-questionnaires for contradictory evidence.

These principles investigate the roles of hermeneutics, anthropology, and phenomenology in eliciting knowledge through interpretative investigations. Following the guidelines does not automatically result in interesting results, but they help direct the conduct and reporting of interpretative research and "can also be used for post hoc evaluation" (Klein & Myers 1999, p. 71).

9.4 Contribution to knowledge

Practical level:

First, by developing the PD4PCT framework to guide researchers and training providers in designing effective professional development programmes for primary school teachers to teach CT skills, this study contributed to professional development on a practical level.

Second, the study has demonstrated that a consistent PD programme may significantly and favourably affect primary teachers' comprehension of computational concepts and their attitudes and beliefs towards teaching CT. Studies on the effects of PD for CT have focused on the effects of brief programmes, like one day workshops, as opposed to sustained over a month, and include secondary school teachers.

Third, I discovered that teachers' plans to incorporate CT, in particular, subject areas were influenced by their subject matter knowledge.

Studies investigating computing education and PD organisations may benefit from these insights to better comprehend teachers' difficulties while incorporating CT into various topics and to create strategies and resources to help them do so.

Theoretical Level

This study is distinctive because it developed the PD4PCT framework and assessed its implementation on primary school teachers in a developing country. Apart from that the framework also consists of the context component at the school level as the main element that guides other components. The school context should be considered first regarding technological infrastructure and school leadership before planning an intervention for teachers. Another unique feature of this framework is the participatory design approach used during the intervention. This approach made teachers design partners, which afforded them equal ownership of the design process.

Three frameworks for teacher development in CT (3C, CTTD, ADAPPTER) by Jocius et al. (2020), Kong and Lai (2021) and Kirwan et al. (2022) exist, but they were all developed and implemented in a developed country context. These frameworks are discussed in more detail in section 2.13.3.

The developed PD4PCT framework is compared against the three existing professional development frameworks in Table 9.3 below using the criteria of components of the framework, teaching approaches that can be used, target population, target educational level and the country context where it was developed and implemented.

Table 9.3 Comparing the PD4PCT framework to the three existing professional development frameworks for CT

	PD4PCT	**3C Jocius et al., (2020)**	**ADAPPTER Kirwan et al., (2022)**	**CTTD Kong & Lai (2021)**
Components	Content Focus, Active Learning, Collective participation, Coherence, Duration, Context Subject Matter Knowledge	**Code** (*programming knowledge*) **Connect** (*content and PRADA knowledge)* **Create** (*Learning segment, Snap! Protoype, Lesson Plans*)	Activities, Demonstrations, Application, Pre-activation, Transparency, Theory, Exemplification, and Reflection	Content knowledge (CK) Pedagogical Content Knowledge (PCK) Technological content knowledge (TCK) Technological pedagogical Content knowledge (TPACK)
Teaching Approaches	Programming, Unplugged, Robotics, Game-based Learning, Project-based Learning	Programming, Unplugged	Unplugged	Programming Unplugged, project-based learning

Target population	In-service teachers	In-service teachers (Math Science, Social studies, English, Arts)	In-service teachers	In-service teachers
Target Level	Primary schools	K-12: middle and high school	Secondary schools	Primary schools
Country context	Developing (Namibia)	Developed (USA)	Developed (Ireland)	Developed (Hong Kong)

From the above, it is clear that the ADAPPTER framework was purely for secondary school teachers and focused more on the teaching process, while the 3C framework combined middle school and high school teachers. None of the three existing frameworks explicitly mentions the elements of context, collective participation, and coherence (aligning the intervention with teachers' beliefs and attitudes towards CT). The PD4PCT framework contributes towards the scientific body of knowledge on the professional development of primary school teachers in a developing context to integrate CT into their teaching.

9.5 Research Limitations

It is crucial to highlight the constraints that this study had. A key limitation is that the study was limited to 14 teachers from one school in Windhoek, focusing only on the senior primary phase. Case studies, unlike qualitative surveys, are not meant to generalise "from samples to universes," as stated by Yin (2012, p. 18). The claims made when generalising from cases cannot be "proof" in a statistical sense. Instead, they develop theoretical foundations that make claims about circumstances similar to the one being researched. Additionally, if other case studies provide similar results, they might be considered to confirm the assumptions and so contribute to develop the theory (Yin, 2012).

Time was another limitation, as getting all teachers to attend all the sessions was difficult. Due to other school commitments, some teachers missed some training sessions, which led to one conclusion from this study that teachers did not have enough time to understand all the CT components.

It was also challenging to collect the qualitative data as I served as the workshop facilitator and the study's sole researcher. When running programme sessions, it was time-consuming to collect all the necessary data while also running the programme sessions.

Another limitation is that this study did not focus that much on the teaching process itself, so one could learn, for example, the ADDAPTER framework and explain that process as part of the framework. Also, the CS Unplugged project offers a great number of ready-to-use materials that could have been explored more and used.

One more limitation was the quantitative instrument to measure the understanding of CT knowledge. Rather than measuring actual CT knowledge, these statements (Table 6.2) are a measure of the teachers' conceptions of their own understanding.

9.6 Recommendations for future research

The study lays the groundwork for future research and adds to the knowledge related to teacher skill development in CT. In light of this, suggestions for more research are provided below.

Understanding the CT curriculum is insufficient; there has to be a mapping to the subject areas where CT can be used, so more research is needed on how these mappings can be done effectively. One recommendation for further research-based on this study is to retest the results in other developing nations in Africa or Asia. This will make it easier to assess if implementing the framework has the same effect in other emerging nations.

Another recommendation is that other techniques may validate a framework in a different setting or get more CT experts onto the team. As new components appear, further study may be done to enhance the framework. It is also recommended to consider adding a basic ICT course to the interventions for teachers who are computer illiterate, so they can use computing devices. Even if they initially use unplugged activities, it will be necessary to advance their CT skills through computing devices.

For the future iteration of the PD4PCT model, additional days should be added for teaching the topics of CT definition, concepts and approaches as findings revealed that allocated time was not adequate for teachers to grasps everything.

The recent trend of CT2.0, which refers to teaching machine learning (ML) in schools, is another suggestion for future study. The computational thinking (CT) consensus in computing education is challenged by ML technology. In order to prepare for and overcome these challenges, CT must evolve in new ways. We need to determine why and how a variety of traditional "CT1.0" concepts from control structures and problem-solving processes to correctness and hypothetical machines, need to be rethought for the "CT2.0" (machine learning) era.

9.7 Conclusion

Since these abilities are often not taught at that level of education, introducing CT into compulsory education by modifying the curriculum presents significant obstacles for educational stakeholders. One difficulty posed by modifying curriculum is preparing primary teachers to teach these skills because many in-service teachers never studied them during their teacher education, let alone during their compulsory schooling. It may be difficult to add the teaching of more skills to an already packed curriculum since primary school teachers are frequently generalists who already teach several disciplines. However, there are opportunities for engaging and effective CT instruction, particularly at the primary level, where teachers instruct students in various disciplines. Researchers in computing education have urged to include CT in K–12 topics for many years. Arguments in favour of incorporating CT in K–12 schools include the possibility that it will assist teachers in making up for their class time shortage and improving subject matter instruction. My research was on creating and putting into practice a framework for training primary school teachers to teach and integrate CT. My research results offer suggestions for overcoming obstacles to CT integration in elementary school subjects and maximising prospects. However, there is still considerable work to be done to enhance the design, implementation, and research of teachers’ PD for CT, as described in the section on suggestions.

REFERENCES

Ackermann, E. (2001). Piaget′s constructivism, Papert′s constructionism: What′s the difference. *Future of Learning Group Publication*, *5*(3), 438.

Adler, R. F., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies*, *23*(4), 1501–1514. https://doi.org/10.1007/s10639-017-9675-1

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Laine, T. H. (2021). Co-design of mini games for learning computational thinking in an online environment. In *Education and Information Technologies* (Vol. 26, Issue 5). Springer US. https://doi.org/10.1007/s10639-021-10515-1

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835.

Al-Eraky, M. M., Donkers, J., Wajid, G., & Van Merrienboer, J. J. G. (2015). Faculty development for learning and teaching of medical professionalism. *Medical Teacher*, *37*(sup1), S40--S46.

Alqahtani, M. M., Hall, J. A., Leventhal, M., & Argila, A. N. (2021). Programming in Mathematics Classrooms: Changes in Pre-service Teachers' Intentions to Integrate Robots in Teaching. *Digital Experiences in Mathematics Education 2021 8:1*, *8*(1), 70–98. https://doi.org/10.1007/S40751-021-00096-6

Anastasi, A., & Urbina, S. (1997). *Psychological testing*. Prentice Hall/Pearson Education.

Angeli, C. (2022). The effects of scaffolded programming scripts on pre-service teachers' computational thinking: Developing algorithmic thinking through programming robots. *International Journal of Child-Computer Interaction*, *31*, 100329. https://doi.org/10.1016/J.IJCCI.2021.100329

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*. https://doi.org/10.1016/j.chb.2019.106185

Angeli, C., & Valanides, N. (2009). Epistemological and methodological issues for the conceptualization, development, and assessment of ICT--TPCK: Advances in technological pedagogical content knowledge (TPCK). *Computers & Education*, *52*(1), 154–168.

Angeli, C., Valanides, N., & Christodoulou, A. (2016). Theoretical considerations of technological pedagogical content knowledge. *Handbook of Technological Pedagogical Content Knowledge (TPACK) for Educators*, *11*.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge E-QUAL View project Handbook of Information Technology in Primary and Secondary Education View project. In *Educational Technology & Society* (Vol. 19, Issue 3). https://doi.org/10.2307/jeductechsoci.19.3.47

Angevine, C., Cator, K., Roschelle, J., Thomas, S. A., Waite, C., & Weisgrau, J. (2017).

Computational Thinking for a Computational World.

Araujo, A. L. S. O., Andrade, W. L., Guerrero, D. D. S., & Melo, M. R. A. (2019). How Many Abilities Can We Measure in Computational Thinking? *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*. https://doi.org/10.1145/3287324.3287405

Ausiku, M., & Matthee, M. (2021). Preparing Primary School Teachers for Teaching Computational Thinking: A Systematic Review. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12511 LNCS* (pp. 202–213). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-66906-5_19

Babbie, E. R. (2013). *The basics of social research*. Cengage learning.

Balanskat, A., & Engelhardt, K. (2015). Computing our future. *Computer Programming and Coding. Priorities, School Curricula and Initiatives across Europe. European Schoolnet, Brussels*.

Bannon, L., Bardzell, J., & Bødker, S. (2018). *Introduction: Reimagining participatory design—Emerging voices*. ACM New York, NY, USA.

Barakabitze, A. A., Kitindi, E. J., Sanga, C., Kibirige, G., Makwinya, N., Alex Barakabitze, A., … Makwinya, N. (2015). Exploring Students' Skills and Attitudes on Effective Use of ICTs: Case Study of Selected Tanzanian Public Secondary Schools. *Universal Journal of Educational Research*, *3*(6), 407–425. Retrieved from http://www.hrpub.org/journals/article_info.php?aid=2724

Computing at School. (2014). *Computing at Schools*. http://barefootcas.org.uk

Barr, B. D., Harrison, J., & Conery, L. (2011). *Computational Thinking : A Digital Age Skill for Everyone*. *5191*(April), 21–22.

Barr, V., & Stephenson, C. (2011a). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Baxter, J., & Eyles, J. (1997). Evaluating qualitative research in social geography: establishing 'rigour'in interview analysis. *Transactions of the Institute of British Geographers*, *22*(4), 505–525.

Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015). Starting from scratch: Developing a pre-service teacher training program in computational thinking. *Proceedings - Frontiers in Education Conference, FIE*, *2014*, 1–8. https://doi.org/10.1109/FIE.2015.7344237

BEBRAS. (n.d.). *BEBRAS Challenge*. Retrieved October 1, 2021, from https://www.bebras.org/

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

Bell, T., & Henderson, T. (2022). Providing a CS unplugged experience at a distance. *ACM Inroads*, *13*(4), 26–31. https://doi.org/10.1145/3571093

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bevir, M., & Rhodes, R. (2003). *Interpreting british governance*. Routledge.

Bitsch, V. (2005). Qualitative research: A grounded theory example and evaluation criteria. *Journal of Agribusiness*, *23*(345-2016–15096), 75–91.

Blumberg, B. F., Cooper, D. R., & Schindler, P. S. (2005). Survey research. *Business Research Methods*, 243–276.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing Computational Thinking : Approaches and Orientations in K-12 Education. In *Proceedings of the EdMedia 2016 Conference* (Issue June). https://doi.org/10.2791/792158

Bødker, S. (2003). A for Alternatives. *Scandinavian Journal of Information Systems*, *15*(1), 1.

Bødker, S., & Kyng, M. (2018). Participatory design that matters—Facing the big issues. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *25*(1), 1–31.

Bogdan, R. C., & Biklen, S. K. (1998). Foundations of qualitative research in education. *Qualitative Research in Education: An Introduction to Theory and Methods*, 1–48.

Bogner, A., & Menz, W. (2009). The theory-generating expert interview: epistemological interest, forms of knowledge, interaction. In *Interviewing experts* (pp. 43–80). Springer.

Bono, R., Blanca, M. J., Arnau, J., & Gómez-Benito, J. (2017). Non-normal distributions commonly used in health, education, and social sciences: A systematic review. *Frontiers in Psychology*, *8*, 1602.

Boone, H. N., & Boone, D. A. (2012). Analyzing Likert Data. *Journal of Extension*, *50*(2). http://www.joe.org/joe/2012april/tt2p.shtml[8/20/20129:07:48AM]

Borko, H. (2004). Professional development and teacher learning: Mapping the terrain. *Educational Researcher*, *33*(8), 3–15.

Boulden, D. C., Rachmatullah, A., Oliver, K. M., & Wiebe, E. (2021). Measuring in-service teacher self-efficacy for teaching computational thinking: development and validation of the T-STEM CT. *Education and Information Technologies 2021 26:4*, *26*(4), 4663–4689. https://doi.org/10.1007/S10639-021-10487-2

Boulden, D. C., Wiebe, E., Akram, B., Aksit, O., Buffum, P. S., Mott, B., Boyer, K. E., & Lester, J. (2018). Computational Thinking Integration into Middle Grades Science Classrooms: Strategies for Meeting the Challenges. In *Middle Grades Review* (Vol. 4, Issue 3).

Boulden, D. M. (2020). *Building the Capacity of In-Service Teachers to Integrate and Teach Computational Thinking*. North Carolina State University.

Bower, M., Wood, L. N., Howe, C., Lister, R., Lai, J. W. M. M., Howe, C., Lister, R., Lai, J. W. M. M., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, *42*(3), 53–72. https://doi.org/10.14221/ajte.2017v42n3.4

Brackmann, C., Barone, D., Casali, A., Boucinha, R., & Muñoz-Hernandez, S. (2016). Computational thinking: Panorama of the Americas. *2016 International Symposium on Computers in Education (SIIE)*, 1–6.

Brackmann, C. P., Román-gonzález, M., Rey, U., Carlos, J., Moreno-león, J., Rey, U., Carlos, J., Casali, A., Barone, D., Federal, I., Iffar, F., Brackmann, C. P., Román-gonzález, M., Rey, U., Carlos, J., Rey, U., Carlos, J., Moreno-león, J., Casali, A., & Barone, D. (2017). *Development of Computational Thinking Skills through Unplugged Activities in Primary School Gregorio Robles*. *November*, 65–72. https://doi.org/10.1145/3137065.3137069

Brandt, E., Binder, T., & Sanders, E. (2012). Tools and techniques: ways to engage telling, making and enacting. In *Routledge International Handbook of Participatory Design* (pp. 165–201). Routledge. https://doi.org/10.4324/9780203108543-14

Brant, J. M., Haas-Haseman, M. L., Wei, S. H., Wickham, R., & Ponto, J. (2015). Understanding and Evaluating Survey Research. *Journal of the Advanced Practitioner in Oncology*, *6*(2), 168. https://doi.org/10.6004/jadpro.2015.6.2.9

Bråting, K., & Kilhamn, C. (2020). Exploring the intersection of algebraic and computational thinking. *Https://Doi.Org/10.1080/10986065.2020.1779012*, *23*(2), 170–185. https://doi.org/10.1080/10986065.2020.1779012

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Braun, V., & Clarke, V. (2012). Thematic analysis. In *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological.* (pp. 57–71). American Psychological Association. https://doi.org/10.1037/13620-004

Brennan, K. (2015). Beyond technocentrism. *Constructivist Foundations*, *10*(3), 289–296.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association Meeting, Vancouver, BC, Canada*, 1–25. https://doi.org/10.1.1.296.6602

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, *18*(1), 32–42.

Brown, M. M. (1992). Caribbean first-year teachers' reasons for choosing teaching as a career. *Journal of Education for Teaching*, *18*(2), 185–195.

Brush, T., & Saye, J. (2009). Strategies for preparing preservice social studies teachers to effectively integrate technology: Models and practices. *Contemporary Issues in Technology and Teacher Education*, *9*(1), 46–59.

Bryman, A., & Bell, E. (2003). Breaking down the quantitative/qualitative divide. *Business Research Methods*, 465–478.

Brysman, H., & Burgess, R. (1999). *Qualitative research methodology: A review*. London: Routledge.

Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 423–432.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, *1*(2), 67–69.

Burns, M., Santally, M. I., Roopesh, K., Sungkur, K. R., Juggurnath, B., & Rajabalee, Y. (2019). ICTs in secondary education in Sub-Saharan Africa: Policies, practices, trends and recommendations. *Secondary Education in Africa*, 21–25.

Burrell, G., & Morgan, G. (1979). Sociological Paradigms and Organizational. *Analysis. London: Heinemann*.

Butler, D. (2007). A constructionist view of what it means to be digitally literate--Learning in 21st century classrooms. *Nordic Journal of Digital Literacy*, *2*(02), 61–77.

Butler, D., & Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking_ A constructionist approach.pdf. *British Journal of Educational Technology*, *2021;00*, 1–18.

Çakır, R., Şahin, H., Balci, H., & Vergili, M. (2021). The effect of basic robotic coding in-service training on teachers' acceptance of technology, self-development, and computational thinking skills in technology use. *Journal of Computers in Education*, *8*(2), 237–265. https://doi.org/10.1007/S40692-020-00178-1/TABLES/11

Çakıroğlu, Ü., & Kiliç, S. (2020). Assessing teachers' PCK to teach computational thinking via robotic programming. *Interactive Learning Environments*, *0*(0), 1–18. https://doi.org/10.1080/10494820.2020.1811734

Carr, W., & Kemmis, S. (1986). Becoming critical: education. *Knowledge and Action Research. London: Falmer*.

Castillo, J. J. (2009). Research population. *Retrieved August*, *3*, 2011.

Cateté, V., Lytle, N., Dong, Y., Boulden, D., Akram, B., Houchins, J., Barnes, T., Wiebe, E., Lester, J., Mott, B., & Boyer, K. (2018). Infusing computational thinking into Middle Grade Science classrooms. *Proceedings of the 13th Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/3265757.3265778

Cetin, I. (2016). Preservice Teachers' Introduction to Computing: Exploring Utilization of Scratch. *Journal of Educational Computing Research*, *54*(7), 997–1021. https://doi.org/10.1177/0735633116642774

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93–100. https://doi.org/10.1016/j.ijcci.2018.06.005

Chalmers, D., Manley, D., Wasserman, R., & others. (2009). *Metametaphysics: New essays on the foundations of ontology*. Oxford University Press on Demand.

Chin, G. J. (2004). *A case study in the participatory design of a collaborative science-based learning environment* [Virginia Tech]. https://search-proquest-com.proxy3.library.mcgill.ca/dissertations/docview/305105766/abstract/4E1B6275350C4CDCPQ/10?accountid=12339

Cizadlo, T. (2018). *A framework for professional learning and development in science for New Zealand upper primary teachers*.

Clement, A., & den Besselaar, P. (1993). A retrospective look at PD projects. *Communications of the ACM*, *36*(6), 29–37.

Code.org. (n.d.). *Hour of Code |* . Retrieved October 1, 2021, from https://code.org/

Cohen, L., Manion, L., & Morrison, K. (2000). Research Methods in Education [5 th edn] London: Routledge Falmer. *Teaching in Higher Education*, *41*, 21.

Cohen, L., Manion, L., & Morrison, K. (2011). Planning educational research. *Research Methods in Education. New York: Routledge Editors*.

Collis, J., & Hussey, R. (2009). *A Practical Guide for Undergraduate and Postgraduate Students*.

Cooksey, R. W., & McDonald, G. (2011). *Writing up your research: Surviving and thriving in postgraduate research*. Tilde University Press.

Corradini, I., Lodi, M., & Nardelli, E. (2017). Computational Thinking in Italian Schools: Quantitative Data and Teachers' Sentiment Analysis after Two Years of "Programma Il Futuro." *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, 224–229. https://doi.org/10.1145/3059009.3059040

Creswell, J. (2013). Qualitative Inquiry & Research Design: Choosing Among Five Approaches. In *SAGE Publications* (Vol. 11).

Creswell, J. W. (2003). A framework for design. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 9–11.

Creswell, J. W., & Clark, V. L. P. (2017). *Designing and conducting mixed methods research*. Sage publications.

Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory into Practice*, *39*(3), 124–130.

Creswell, J. W., Hesse-Biber, S. N., Leavy, P., Creswell, J. W., & Eidlin, F. (2013). A Framework for Design. Research design: Qualitative, quantitative, and mixed methods approaches. *Political Science & Politics*, *44*(04), 183–233. https://doi.org/10.1017/S1049096511001260

Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*. Sage.

CS Unplugged. (n.d.). Retrieved October 1, 2021, from https://csunplugged.org/en/

Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the Constructionist Learning Theory with Computational Thinking Classroom Activities. *Informatics in Education*, *18*(1), 41–67. https://doi.org/10.15388/infedu.2019.03

CSTA, & ISTE. (2011). *Computational Thinking in K–12 Education leadership toolkit*. http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf.%0A

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: a framework*. https://eprints.soton.ac.uk/369594/

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014). Introducing Teachers to Computational Thinking Using Unplugged Storytelling. *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, 89–92. https://doi.org/10.1145/2670757.2670767

Damodaran, L. (1996). User involvement in the systems design process-a practical guide for users. *Behaviour & Information Technology*, *15*(6), 363–377.

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). Effective teacher professional development. *Learning Policy Institute*.

Dawidowicz, P. (2010). *Literature reviews made easy: A quick guide to success*. IAP.

Denning, P. J. (2009). The profession of ITBeyond computational thinking. *Communications of the ACM*, *52*(6), 28. https://doi.org/10.1145/1516046.1516054

Denscombe, M. (2010). *The Good Research Guide: for small social research projects*. Glasgow, McGraw-Hill House.

Denzin, N. K. (2000). Aesthetics and the practices of qualitative inquiry. *Qualitative Inquiry*, *6*(2), 256–265.

Department of Basic Education of South Africa. (2021). *Draft CAPS for Coding and Robotics*. https://www.education.gov.za/DraftCapsCodingRobotics.aspx

Desimone, L. M. (2009). Improving Impact Studies of Teachers' Professional Development: Toward Better Conceptualizations and Measures. *Educational Researcher*, *38*(3), 181–199. https://doi.org/10.3102/0013189X08331140

Dwivedi, A. K., Mallawaarachchi, I., & Alvarado, L. A. (2017). Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method. *Statistics in Medicine*, *36*(14), 2187–2205. https://doi.org/10.1002/SIM.7263

Doering, A., Veletsianos, G., Scharber, C., & Miller, C. (2009). Using the technological, pedagogical, and content knowledge framework to design online learning environments and professional development. *Journal of Educational Computing Research*, *41*(3), 319–346.

Dolgopolovas, V., Dagienė, V., Jasutė, E., & Jevsikova, T. (2019). Design Science Research for Computational Thinking in Constructionist Education: A Pragmatist Perspective. *Problemos*, *95*, 144–159. https://doi.org/https://doi.org/10.15388/Problemos.95.12

Dolma, S. (2010). The central role of the unit of analysis concept in research design. *İstanbul Üniversitesi İşletme Fakültesi Dergisi Istanbul University Journal of the School of Business Administration Cilt*, *39*(1), 169–174. www.ifdergisi.org

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*. https://doi.org/10.1145/3287324.3287431

Druin, A. (2002). The role of children in the design of new technology. *Behaviour and Information Technology*, *21*(1), 1–25.

Du, X., & Igwe, K. C. (2018). Computational Thinking in Teacher Professional Development Programs. In V. Dagienė & E. Jasute (Eds.), *Constructionism 2018* (pp. 193–203).

Duarte, A. M. B., Brendel, N., Degbelo, A., & Kray, C. (2018). Participatory design and participatory research: an HCI case study with young forced migrants. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *25*(1), 1–39.

Duncan, C. (2019). Computer Science and Computational Thinking in Primary Schools. *University of Canterbury*, 297. http://www.gonzaga.edu/academics/colleges-and-schools/College-of-Arts-and-Sciences/Majors-Programs/Computer-Science-and-Computational-Thinking/default.asp

Duncan, C., Bell, T., & Atlas, J. (2017). What do the teachers think? Introducing computational thinking in the primary school curriculum. *ACM International Conference Proceeding Series*, 65–74. https://doi.org/10.1145/3013499.3013506

Easterby-Smith, M. P. V., Thorpe, R., & Jackson, P. (2008). *Management research: theory and research*. Sage.

Ehn, P. (1989). The art and science of designing computer artifacts. *Scandinavian Journal of Information Systems*. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1210&context=sjis

Elo, S., Kääriäinen, M., Kanste, O., Pölkki, T., Utriainen, K., & Kyngäs, H. (2014). Qualitative Content Analysis. *SAGE Open*, *4*(1), 215824401452263. https://doi.org/10.1177/2158244014522633

Emembolu, I. C., Emembolu, C., Umechukwu, K., Sulaiman, A., & Aderinwale, O. (2020). Building a Community of STEM Educators in Nigeria Using the TeachAKid2Code Program. *Advances in Intelligent Systems and Computing*, *1135 AISC*, 536–545. https://doi.org/10.1007/978-3-030-40271-6_53

Enochs, L. G., & Riggs, I. M. (2002). *Science Teaching Efficacy Belief Instrument*

(STEBI) – Form B (Pre-Service). *October*, 1990.

Enochs, L. G., Smith, P. L., & Huinker, D. (2000). Establishing factorial validity of the mathematics teaching efficacy beliefs instrument. *School Science and Mathematics*, *100*(4), 194–202.

Ertmer, P. A., Ottenbreit-Leftwich, A. T., Sadik, O., Sendurur, E., & Sendurur, P. (2012). Teacher beliefs and technology integration practices: A critical relationship. *Computers & Education*, *59*(2), 423–435. https://doi.org/10.1016/J.COMPEDU.2012.02.001

Espinal, A., Vieira, C., & Magana, A. J. (2021). Professional Development in Computational Thinking for teachers in Colombia. *Proceedings - Frontiers in Education Conference, FIE*, *2021-October*. https://doi.org/10.1109/FIE49875.2021.9637310

Esterberg, K. (2002). *Qualitative methods in social research*. McGraw-Hill.

Esteve-Mon, F. M., Adell-Segura, J., Ángeles, M., Nebot, L., Novella, G. V., Aparicio, J. P., Llopis Nebot, M. Á., Valdeolivas Novella, G., & Pacheco Aparicio, J. (2019). The Development of Computational Thinking in Student Teachers through an Intervention with Educational Robotics. *Journal of Information Technology Education: Innovations in Practice*, *18*, 139–152. https://doi.org/10.28945/4442

Faber, H. H., Wierdsma, M. D. M., Doornbos, R. P., Van Der Ven, J. S., & De Vette, K. (2017). Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network*, *12*, 13–24. https://jeten-online.org/index.php/jeten/article/viewFile/131/159

Fadhel, K. (2002). Positivist and hermeneutic paradigm: A critical evaluation under their structure of scientific practice. *The Sosland Journal*, *6*(5), 21–28.

Falcão, T. P., e Peres, F. M., de A., de Morais, D. C. S., & da Silva Oliveira, G. (2018). Participatory methodologies to promote student engagement in the development of educational digital games. *Computers & Education*, *116*, 161–175.

Falkner, K., Vivian, R., & Falkner, N. (2015). Teaching Computational Thinking in K-6 : The CSER Digital Technologies MOOC. *Ace '15*.

Falkner, K., Vivian, R., & Falkner, N. (2018). Supporting Computational Thinking Development in K-6. *Proceedings - 2018 6th International Conference on Learning and Teaching in Computing and Engineering, LaTiCE 2018*, 126–133. https://doi.org/10.1109/LaTICE.2018.00031

Fares, K., Fowler, B., & Vegas, E. (2021). *How South Korea implemented its computer science education program*. https://www.brookings.edu/wp-content/uploads/2021/10/How-South-Africa-implemented-its-CS-education-program_FINAL.pdf

Fessakis, G., & Prantsoudi, S. (2019). Computer Science Teachers' Perceptions, Beliefs and Attitudes on Computational Thinking in Greece. In *Informatics in Education* (Vol. 18, Issue 2, pp. 227–258).

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R*. Sage publications.

Finnis, J. (2011). *Natural law and natural rights*. Oxford University Press.

Frauenberger, C., Good, J., Fitzpatrick, G., & Iversen, O. S. (2015). In pursuit of rigour and accountability in participatory design. *International Journal of Human-Computer Studies*, *74*, 93–106.

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational Thinking in MATHS Teacher Education.pdf. *Contemporary Issues in Technology and Teacher Education*, *17*(4). https://citejournal.org/volume-17/issue-4-17/mathematics/computational-thinking-in-mathematics-teacher-education/

Galliers, R. D. (1991). Choosing appropriate information systems research methods. In *Information systems research: contemporary approaches and emergent traditions*. North Holland, Amsterdam.

Garet, M. S., Porter, A. C., Desimone, L., Birman, B. F., & Yoon, K. S. (2001). What makes professional development effective? Results from a national sample of teachers. *American Educational Research Journal*, *38*(4), 915–945.

Geldreich, K., Talbot, M., & Hubwieser, P. (2018). Off to new shores: Preparing primary school teachers for teaching algorithmics and programming. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3265757.3265783

Gleasman, C., & Kim, C. (2020). Pre-Service Teacher's Use of Block-Based Programming and Computational Thinking to Teach Elementary Mathematics. *Digital Experiences in Mathematics Education*. https://doi.org/10.1007/s40751-019-00056-1

Glesne, C. (2011). Becoming qualitative researchers: An introduction 4th ed. *Cranbury, NJ*.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The Qualitative Report*, *8*(4), 597–607.

Goldkuhl, G. (2004). Meanings of pragmatism: Ways to conduct information systems research. *Action in Language, Organisations and Information Systems*, 13–26.

Goldkuhl, G. (2012). Pragmatism *vs* interpretivism in qualitative information systems research. *European Journal of Information Systems*, *21*(2), 135–146. https://doi.org/10.1057/ejis.2011.54

Google for Education. (2016). *Exploring computational thinking*. https://edu.google.com/resources/programs/exploring-computational-thinking/

Gramm, A., Hornung, M., & Witten, H. (2012). Email for you (only?) design and implementation of a context-based learning process on internetworking and cryptography. *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, 116–124.

Grgurina, N., Mannila, L., Dagiene, V., Settle, A., Demo, B., Rolandsson, L., Mirolo, C., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, 1–29. https://doi.org/10.1145/2713609.2713610

Grönvall, E., & Kyng, M. (2013). On participatory design of home-based healthcare. *Cognition, Technology & Work*, *15*(4), 389–401.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. *Ectj*, *29*(2), 75–91.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 105–117). Sage Publications, Inc.

Haduong, P., & Brennan, K. (2019). Helping K--12 Teachers Get Unstuck with Scratch: The Design of an Online Professional Learning Experience. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1095–1101. https://doi.org/10.1145/3287324.3287479

Hall, N. A., & Hall, R. (2008). *Applied social research*. Macmillan Education AU.

Hancock, D. R., & Algozzine, B. (2006). *A practical guide for beginning researchers doing case study research*. New York: Teachers College Press.

Harris, J., & Hofer, M. (2009). Grounded Tech Integration: An Effective Approach Based on Content, Pedagogy, and Teacher Planning. *Learning & Leading with Technology*, *37*(2), 22–25.

Hartas, D. (2010). *Educational Research and Inquiry: Qualitative and Quantitative Approaches*.

Hay, C. (2011). Interpreting interpretivism interpreting interpretations: The new hermeneutics of public administration. *Public Administration*, *89*(1), 167–182.

Heale, R., & Twycross, A. (2015). Validity and reliability in quantitative studies. *Evidence-Based Nursing*, *18*(3), 66–67.

Hennink, M., Hutter, I., & Bailey, A. (2011). Data preparation and developing codes. *Qualitative Research Methods. London: SAGE Publications Ltd*, 201–229.

Hickmott, D. P. (2020). *A Study of a Professional Learning Program for Primary School Teachers Implementing the New South Wales Science and Technology Syllabus Statement of Originality*. *February*.

Hickmott, D., & Prieto-Rodriquez, E. (2018). To Assess or Not to Assess: Tensions Negotiated in Six Years of Teaching Teachers about Computational Thinking. *Informatics in Education*, *17*(2), 229–244. https://doi.org/10.15388/infedu.2018.12

Hoic-Bozic, N., Loncaric, D., & Dlab, M. H. (2019). Preparing Primary Junior Grade Teachers To Teach Computational Teaching: Experiences From the Glat Project. *Mathematics and Informatics*, *62*(5), 487–499.

Hsieh, M. C., Pan, H. C., Hsieh, S. W., Hsu, M. J., & Chou, S. W. (2022). Teaching the Concept of Computational Thinking: A STEM-Based Program With Tangible

Robots on Project-Based Learning Courses. *Frontiers in Psychology*, *12*(January), 1–7. https://doi.org/10.3389/fpsyg.2021.828568

Hsu, T.-C. C., Chang, S.-C. C., & Hung, Y.-T. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. https://doi.org/https://doi.org/10.1016/j.compedu.2018.07.004

Huang, W., & Looi, C. K. (2020). A critical review of literature on “unplugged” pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, *31*(1), 1–29. https://doi.org/10.1080/08993408.2020.1789411

Humphreys, S., Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking A guide for teachers*. Hachette UK.

Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: Mathematics+ computational thinking. *Interactive Learning Environments*, *28*(3), 362–382.

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 and Education*, *82*, 263–279. https://doi.org/10.1016/j.compedu.2014.11.022

ISTE. (2021). *Computational thinking competencies*. International Society for Technology in Education. https://www.iste.org/standards/iste-standards-for-computational-thinking

Iversen, O. S., Halskov, K., & Leong, T. W. (2012). Values-led participatory design. *CoDesign*, *8*(2–3), 87–103.

Iversen, O. S., Smith, R. C., & Dindler, C. (2017). Child as protagonist: Expanding the role of children in participatory design. *Proceedings of the 2017 Conference on Interaction Design and Children*, 27–37.

Jacob, S. R., & Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, *1*(1), 1. https://doi.org/10.26716/jcsi.2018.01.1.1

Jagušt, T., Krzic, A. S., Gledec, G., Grgić, M., & Bojic, I. (2018). Exploring Different Unplugged Game-like Activities for Teaching Computational Thinking. *2018 IEEE Frontiers in Education Conference (FIE)*, 1–5. https://doi.org/10.1109/FIE.2018.8659077

Jaipal-jamani, K., & Angeli, C. (2017). Effect of Robotics on Elementary Preservice Teachers ’ Self-Efficacy , Science Learning , and Computational Thinking. *Journal of Science Education & Technology*, *26*(2), 175–192. https://doi.org/10.1007/s10956-016-9663-z

Jevsikova, T., Dagienė, V., Stupurienė, G., & Juškevičienė, A. (2019). Teaching Computational Thinking in Primary Schools : Worldwide Trends and Teachers ’ Attitudes *. *Informatics in Schools. New Ideas in School Informatics. ISSEP*, *11913*(0), 83–94. https://doi.org/https://doi.org/10.1007/978-3-030-33759-9_7

Jocius, R., Joshi, D., Dong, Y., Robinson, R., Catete, V., Barnes, T., Albert, J., Andrews, A., & Lytl, N. (2020). Code, connect, create: The 3c professional development model to support computational thinking infusion. *SIGCSE 2020 - Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 971–977. https://doi.org/10.1145/3328778.3366797

Johnson, P., & Duberley, J. (2000). *Understanding management research: An introduction to epistemology*. Sage.

Jones, B. F., Rasmussen, C. M., & Moffitt, M. C. (1997). *Real-life problem solving: A collaborative approach to interdisciplinary learning.* American Psychological Association.

Kadirhan, Z., Gül, A., & Battal, A. (2018). Self-efficacy to teach coding in K-12 education. In *Self-Efficacy in Instructional Technology Contexts* (pp. 205–226). Springer International Publishing. https://doi.org/10.1007/978-3-319-99858-9_12

Kafai, Y. B. (1996). Learning design by making games. *Constructionism in Practice: Designing, Thinking and Learning in a Digital World*, 71–96.

Kafai, Y. B., & Burke, Q. (2013). The social turn in K-12 programming: moving from computational thinking to computational participation. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 603–608.

Kale, U., Akcaoglu, M., Cullen, T., & Goh, D. (2018). Contextual Factors Influencing Access to Teaching Computational Thinking. *Computers in the Schools*, *35*(2), 69–87. https://doi.org/10.1080/07380569.2018.1462630

Kalelİoğlu, F., Gülbahar, Y., & Kukul, V. (2016). *A Framework for Computational Thinking Based on a Systematic Research Review*. *4*(3), 583–596.

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: a literature-informed Delphi study. *Https://Doi.Org/10.1080/14794802.2020.1852104*, *23*(2), 159–187. https://doi.org/10.1080/14794802.2020.1852104

Kaya, E., Newley, A., Yesilyurt, E., & Deniz, H. (2019). Improving preservice elementary teachers' engineering teaching efficacy beliefs with 3D design and printing. *Journal of College Science Teaching*, *48*(5), 76–83.

Kaya, E., Yesilyurt, E., Newley, A., & Deniz, H. (2019). Examining the Impact of a Computational Thinking Intervention on Pre-service Elementary Science Examining the Impact of a Computational Thinking Intervention on Pre-service Elementary Science Teachers ' Computational Thinking Teaching Efficacy Beliefs , I. *Jl. of Computers in Mathematics and Science Teaching*, *38*(4), 385–392.

Kelly, A. E. (2013). When is design research appropriate. *Educational Design Research*, 135–150.

Kelter, J., Peel, A., Bain, C., Anton, G., Dabholkar, S., Horn, M. S., & Wilensky, U. (2021). Constructionist co-design: A dual approach to curriculum and professional development. *British Journal of Educational Technology*, *52*(3), 1043–1059. https://doi.org/10.1111/bjet.13084

Kenny, R. F., & Wirth, J. (2009). Implementing Participatory, Constructivist Learning Experiences through Best Practices in Live Interactive Performance. *The Journal of Effective Teaching*, *9*(1), 34–47. https://eric.ed.gov/?id=EJ1092145

Kensing, F., & Blomberg, J. (1998). Participatory design: Issues and concerns. *Computer Supported Cooperative Work (CSCW)*, *7*(3), 167–185.

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2019). Teacher Change Following a Professional Development Experience in Integrating Computational Thinking into Elementary Science. *Journal of Science Education and Technology*, 173–187. https://doi.org/10.1007/s10956-019-09798-4

Keune, A., Toikkanen, T., & Leinonen, T. (2014). *FACILITATOR GUIDE FOR DESIGNING LEARNING ACTIVITIES Participatory Design Model*.

Kidd, T., & Morris Jr, L. R. (Eds.). (2017). *Handbook of research on instructional systems and educational technology*. IGI Global.

Killen, H., Coenraad, M., Byrne, V. L., Cabrera, L., & Ketelhut, D. J. (2020). Reimagining computational thinking professional development: Benefits of a community of practice model. *Computer-Supported Collaborative Learning Conference, CSCL*, *4*(October), 2125–2132.

Kinley. (2015). *PROFESSIONAL DEVELOPMENT THROUGH PARTICIPATORY DESIGN: An Attempt to Enhance ICT Use inTeaching At the Royal University of Bhutan.*

Kirwan, C., Costello, E., & Donlon, E. (2022). ADAPTTER: Developing a Framework for Teaching Computational Thinking in Second-Level Schools by Design Research. *TechTrends*, *66*(3), 495–509. https://doi.org/10.1007/s11528-022-00735-8

Kivunja, C., & Kuyini, A. B. (2017). Understanding and Applying Research Paradigms in Educational Contexts. *International Journal of Higher Education*, *6*(5), 26. https://doi.org/10.5430/ijhe.v6n5p26

Kjällander, S., Mannila, L., Åkerfeldt, A., & Heintz, F. (2021). *Elementary Students' First Approach to Computational Thinking and Programming*. https://doi.org/10.3390/educsci11020080

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly: Management Information Systems*, *23*(1), 67–94. https://doi.org/10.2307/249410

Koehler, M. J., & Mishra, P. (2009). What Is Technological Pedagogical Content Knowledge? – CITE Journal. *Contemporary Issues in Technology and Teacher Education*, *9*(1). https://www.citejournal.org/volume-9/issue-1-09/general/what-is-technological-pedagogicalcontent-knowledge

Koehler, M. J., Mishra, P., & Cain, M. W. (2013). What is Technological Pedagogical Content Knowledge (TPACK)? *Journal of Education*, *193*(3), 13–19. https://doi.org/10.1177/002205741319300303

Knowles, M. S. (1984). *Andragogy in action*.

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.

Komperda, R., Pentecost, T. C., & Barbera, J. (2018). Moving beyond alpha: A primer on alternative sources of single-administration reliability evidence for quantitative chemistry education research. *Journal of Chemical Education*, *95*(9), 1477–1491.

Kong, R., & Wong, G. K. W. W. (2017). Teachers' perception of professional development in coding education. *Proceedings of 2017 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2017*, *2018-Janua*(December), 377–380. https://doi.org/10.1109/TALE.2017.8252365

Kong, S.-C., & Lao, A. C.-C. (2019). Assessing In-Service Teachers' Development of Computational Thinking Practices in Teacher Development Courses. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 976–982. https://doi.org/10.1145/3287324.3287470

Kong, S. C., & Lai, M. (2021). A proposed computational thinking teacher development framework for K-12 guided by the TPACK model. *Journal of Computers in Education*, *9*(3), 379–402. https://doi.org/10.1007/s40692-021-00207-7

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers and Education*, *151*(March), 103872. https://doi.org/10.1016/j.compedu.2020.103872

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. https://doi.org/10.1016/j.chb.2017.01.005

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154–171. https://doi.org/10.1007/s40751-017-0031-2

Kukali, A. N. (2013). Opportunities and challenges for use of Information Communication and Technology in management of public secondary schools in Bungoma South District Kenya. *International Journal of Science and Research (IJSR) India Online ISSN*, 231–97064.

Kurt, S. (2021). *Constructivist Learning Theory - Educational Technology*. Educational Technology. https://educationaltechnology.net/constructivist-learning-theory/

Kyza, E. A., & Georgiou, Y. (2014). Developing In-Service Science Teachers' Ownership of the Profiles Pedagogical Framework through a Technology-Supported Participatory Design Approach to Professional Development. *Science Education International*, *25*(2), 57–77.

Kyza, E., & Nicolaidou, I. (2017). Co-designing reform-based online inquiry learning environments as a situated approach to teachers' professional development. *CoDesign*, *13*(4), 261–286.

Lamprou, A., & Repenning, A. (2018). Teaching how to teach computational thinking. *Annual Conference on Innovation and Technology in Computer Science*

Education, ITiCSE, 69–74. https://doi.org/10.1145/3197091.3197120

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Lavigne, H., Presser, A. L., Rosenfeld, D., Wolsky, M., & Andrews, J. (2020). *Creating a Preschool Computational-Thinking Learning Blueprint to Guide the Development of Learning Resources for Young Children | NSTA*. Connected Science Learning https://www.nsta.org/connected-science-learning/connected-science-learning-april-june-2020/creating-preschool

Laws, S., Harper, C., & Marcus, R. (2003). *Research for Development: A practical guide*. Sage Publications. London.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, *29*(1), 1–8. https://doi.org/10.1007/s10956-019-09803-w

Leinonen, T., Toikkanen, T., & Silfvast, K. (2008). *Software as hypothesis: research-based design methodology.* 61–70. https://doi.org/10.1145/1795234.1795244

Leonard, J., Barnes-Johnson, J., Mitchell, M., Unertl, A., Stubbe, C. R., & Ingraham, L. (2017). Developing Teachers' Computational Thinking Beliefs and Engineering Practices through Game Design and Robotics. In *North American Chapter of the International Group for the Psychology of Mathematics Education*.

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. *Journal of Science Education and Technology*, *25*(6), 860–876. https://doi.org/10.1007/s10956-016-9628-2

Leonard, J., Mitchell, M., Barnes-Johnson, J., Unertl, A., Outka-Hill, J., Robinson, R., & Hester-Croff, C. (2018). Preparing Teachers to Engage Rural Students in Computational Thinking Through Robotics, Game Design, and Culturally Responsive Teaching. *Journal of Teacher Education*, *69*(4), 386–407. https://doi.org/10.1177/0022487117732317

Lewin, C., & McNicol, S. (2014). *Creating the future classroom: Evidence from the iTEC project: Summary report*. Manchester Metropolitan University.

Lin, V., & Shaer, O. (2016). Beyond the lab: Using technology toys to engage South African youth in computational thinking. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 655–661.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. sage.

Linde-koomen, D. (2019). Computational Thinking on Primary Education Teacher Education. *EdMedia + Innovate Learning 2019 - Amsterdam, Netherlands, June 24-28, 2019 Computational*, 684–691.

Lindtner, S., & Lin, C. (2017). Making and its promises. *CoDesign*, *13*(2), 70–82.

Little, J. W., & Lieberman, A. (1987). Teachers as colleagues. *Report No Pub Date*, *170*.

Littlejohn, A. H., & Stefani, L. A. J. (1999). Effective use of communication and information technology: bridging the skills gap. *ALT-J*, *7*(2), 66–76.

Lloyd, M., & Chandra, V. (2020). Teaching coding and computational thinking in primary classrooms: perceptions of Australian preservice teachers. *Curriculum Perspectives*, *40*(2), 189–201. https://doi.org/10.1007/s41297-020-00117-1

Loi, D. (2004). Teaching Design Theory and Practice : a Participatory Journey. *International Design Conference - Design 2004*, 1–6. https://www.designsociety.org/publication/19818/TEACHING+DESIGN+THEORY+AND+PRACTICE%3A+A+PARTICIPATORY+JOURNEY

Looi, C. K., How, M. L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education*, *28*(3), 255–279. https://doi.org/10.1080/08993408.2018.1533297

Louckes-Horsley, S., Love, N., Stiles, K. E., Mundrey, S., & Hewson, P. (1998). *Designing professional development for teachers of science and mathematics*. Thousand Oaks, CA: Corwin Press.

Lu, J. J., & Fletcher, G. H. L. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, *41*(1), 260–264.

Lye, S. Y., Hwee, J., Koh, L., & 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

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

Mardi, F. (2020). Using think alouds and digital powerups to embed computational thinking concepts while in-service teachers reflect on a math solution design project. *Https://Doi.Org/10.1080/21532974.2020.1781001*, *36*(4), 237–249. https://doi.org/10.1080/21532974.2020.1781001

Margolis, J. (2010). *Stuck in the shallow end: Education, race, and computing*. MIT Press.

Marksbury, N. (2017). Monitoring the Pipeline: STEM Education in Rural U.S. In *Forum on Public Policy Online* (Vol. 2017, Issue 2).

Marshall, P. H., Kelder, J. C., & Perry, A. (2005). Social constructionism with a twist of pragmatism: A suitable cocktail for information systems research. *ACIS*, EJ.

Martinez, S. L., & Stager, G. (2019). Invent to learn. In *Making, Tinkering, and Engineering in the Classroom. Torrance, Canada: Construting Modern Knowledge* (2nd ed.). Constructing Modern Knowledge Press.

Mason, S. L. (2018). *Teacher Professional Development for computational thinking*.

Mason, S. L., & Rich, P. J. (2019). Preparing Elementary School Teachers to Teach Computing, Coding, and Computational Thinking. In *Contemporary Issues in Technology and Teacher Education (CITE Journal)* (Vol. 19, Issue 4).

Massachusetts Department of Elementary and Secondary Education. (2016). *2016 Massachusetts Digital Literacy and Computer Science (DLCS) curriculum framework*. *June*. http://www.doe.mass.edu/frameworks/dlcs.pdf

Mcbride, K., Misnikov, Y., & Draheim, D. (2021). *Discussing The Foundations for Interpretivist Digital Government Research*.

McGinnis, J. R., Ketelhut, D. J., Mills, K., Hestness, E., Jeong, H., & Cabrera, L. (2019). Preservice Science Teachers' Intentions and Avoidances to Integrate Computational Thinking into Their Science Lesson Plans for Young Learners. *Grantee Submission*.

McMillan, J. H. (2012). *Educational Research: Fundamentals for the Consumer*. Pearson College Division. https://books.google.com.na/books?id=AaeWMQEACAAJ

Mensan, T., Osman, K., & Majid, N. A. A. (2020). Development and Validation of Unplugged Activity of Computational Thinking in Science Module to Integrate Computational Thinking in Primary Science Education. *Science Education International*, *31*(2), 142–149. https://doi.org/10.33828/sei.v31.i2.2

Merriam, S. B. (1998). *Qualitative Research and Case Study Applications in Education. Revised and Expanded from" Case Study Research in Education.".* ERIC.

Mertens, D. M., & McLaughlin, J. A. (2004). Research and evaluation methods in special education. In *Research and evaluation methods in special education.* https://doi.org/10.4135/9781412985666

Meuser, M., & Nagel, U. (2009). The expert interview and changes in knowledge production. In *Interviewing experts* (pp. 17–42). Springer.

Mingers, J., & Brocklesby, J. (1997). Multimethodology: Towards a framework for mixing methodologies. *Omega*, *25*(5), 489–509.

Ministry of Education. (2005). *Namibian National ICT Policy for Education: Ministry of Basic Education, Sports and Culture: Ministry of Higher Education, Training and Employment Creation.* NIED. http://www.tech.na/download/ICTPolicy2005_15March2005.pdf

Ministry of Education. (2019). *Namibian National ICT Policy for Education: Ministry of Education, Arts and Culture.* NIED.

Mirel, B. (1998). "Applied Constructivism" for User Documentation Alternatives to Conventional Task Orientation. *Article in Journal of Business and Technical Communication*, *12*(1), 7–49. https://doi.org/10.1177/1050651998012001002

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, *108*(6), 1017–1054.

Montealegre, R. (1999). A case for more case study research in the implementation of Information Technology in less-developed countries. *Information Technology for Development*, *8*(4), 199–207. https://doi.org/10.1080/02681102.1999.9525310

Montiel, H., & Gomez-Zermeño, M. (2021). *Educational Challenges for Computational Thinking in K-12 Education: A Systematic Literature Review of "Scratch" as an Innovative Programming Tool*. https://doi.org/10.3390/computers10060069

Mor, Y., Warburton, S., & Winters, N. (2012). Participatory pattern workshops: a methodology for open learning design inquiry. *Research in Learning Technology*, *20*.

Morse, J. M., Barrett, M., Mayan, M., Olson, K., & Spiers, J. (2002). Verification Strategies for Establishing Reliability and Validity in Qualitative Research. *International Journal of Qualitative Methods*, *1*(2), 13–22. https://doi.org/10.1177/160940690200100202

Mouza, C., Yang, H., Pan, Y. C., Yilmaz Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, *33*(3). https://doi.org/10.14742/ajet.3521

Muller, M. J., & Kuhn, S. (1993). Participatory design. *Communications of the ACM*, *36*(6), 24–28. https://doi.org/10.1145/153571.255960

Mumford, E. (1981). Participative Systems Design: Structure and Method. *SYS. OBJECTIVES, SOLUTIONS.*, *1*(1), 5–19.

Muñoz del Castillo, A., Muñoz, M. A., Acosta Huertas, L. C., Herrera, E., Jimenez Toledo, J., & Ramos, D. X. (2019). Developing a Teacher Training Curriculum Including Computational Thinking Skills: Early Advances on a Study Focused on Colombia. *2019 XIV Latin American Conference on Learning Technologies (LACLO)*, 8–11. https://doi.org/10.1109/LACLO49268.2019.00012

Myers, M. D. (2009). Qualitative research in business & management. In *Qualitative research in business & management.* Sage Publications Ltd.

Myers, M. D. (2013). *Qualitative research in business and management*. Sage.

National Institute for Educational Development. (2015). *Revised Basic Education Curriculum: Syllabuses & Learning Material*. Ministry of Education, Arts & Culture. http://www.nied.edu.na/documents/syllabuses/

National Research Council, N. (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. National Academies Press. https://doi.org/10.17226/12840

Ng, W. S. (2017). Coding education for kids: What to learn? How to prepare teachers? *Proceedings of ICICTE*, 195–205. http://www.icicte.org/ICICTE_2017_Proceedings/6.2_Ng 2017.pdf

Nickerson, H., Brand, C., & Repenning, A. (2015). Grounding computational thinking skill acquisition through contextualized instruction. *ICER 2015 - Proceedings of the 2015 ACM Conference on International Computing Education Research*, 207–

216. https://doi.org/10.1145/2787622.2787720

Nieuwenhuis, J. (2016). Qualitative research designs and data collecting techniques. *First Steps in Research*, 72–100.

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Primary Mathematics Teachers' Understanding of Computational Thinking. *KI - Kunstliche Intelligenz*, *36*(1), 35–46. https://doi.org/10.1007/s13218-021-00750-6

Nyakito, C., Amimo, C., & Allida, V. (2021). Challenges of Integrating Information and Communication Technology in Teaching among National Teachers' Colleges in Uganda. *East African Journal of Education and Social Sciences*, *2*(3), 157–171. https://doi.org/https://doi.org/10.46606/eajess2021v02i03.0114

Oates, B. J. (2005). *Researching information systems and computing*. Sage.

Ogegbo, A. A., & Ramnarain, U. (2022). Teachers' Perceptions of and Concerns About Integrating Computational Thinking into Science Teaching After a Professional Development Activity. *African Journal of Research in Mathematics, Science and Technology Education*, 1–11.

Olivier, M. S. (2009). *Information technology research: A practical guide for computer science and informatics*. Van Schaik.

Olmo, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, *2*(1), 1–28. https://doi.org/10.1287/isre.2.1.1

Ostwald, J. (1996). Knowledge construction in software development. *Unpublished Doctoral Dissertation, University of Colorado, Boulder, CO*.

Ouyang, Y., Hayden, K. L., & Remold, J. (2018a). Introducing Computational Thinking through Non-Programming Science Activities. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 308–313. https://doi.org/10.1145/3159450.3159520

Ouyang, Y., Hayden, K. L., & Remold, J. (2018b). Preparing Upper Elementary School Teachers for Integrating Computational Thinking into Regular Classroom Activities. *J. Comput. Sci. Coll.*, *33*(4), 165–171.

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*, *50*(2), 149–163. https://doi.org/10.1080/15391523.2018.1431573

Pajares, M. F. (1992). Teachers' beliefs and educational research: Cleaning up a messy construct. *Review of Educational Research*, *62*(3), 307–332.

Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas*. Basic Books. https://dl.acm.org/citation.cfm?id=1095592

Papert, S. (1997). Looking at technology through school-colored spectacles. *Logo Exchange*.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1–11.

Pawlowski, S. D., & Robey, D. (2004). Bridging user organizations: Knowledge brokering and the work of information technology professionals. *MIS Quarterly: Management Information Systems*, *28*(4), 645–672. https://doi.org/10.2307/25148658

Piaget, J. (1955). The construction of reality in the child. *Journal of Consulting Psychology*, *19*(1), 77.

Piaget, J. (2001). In RL Campbell. *Studies in Reflecting Abstraction*.

Plomp, T. (2013). Educational design research: An introduction. *Educational Design Research*, 11–50.

Põldoja, H., Väljataga, T., Laanpere, M., & Tammets, K. (2014). Web-based self-and peer-assessment of teachers' digital competencies. *World Wide Web*, *17*(2), 255–269.

Punch, K. (2005). *Introduction to Social Research Quantitative and Qualitative Approaches* (2nd ed.). SAGE Publications Ltd.

Putnam, R. T., & Borko, H. (1997). Teacher learning: Implications of new views of cognition. In *International handbook of teachers and teaching* (pp. 1223–1296). Springer.

Putnam, R. T., & Borko, H. (2000). What do new views of knowledge and thinking have to say about research on teacher learning? *Educational Researcher*, *29*(1), 4–15.

Reding, T. E., & Dorn, B. (2017). Understanding the" teacher experience" in primary and secondary CS professional development. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 155–163.

Repenning, A., Basawapatna, A., Yadav, A., Gretter, S., Good, J., Mclean, T., Repenning, A., & Basawapatna, A. (2017). Emerging Research, Practice, and Policy on Computational Thinking. *Emerging Research, Practice, and Policy on Computational Thinking*, *July 2020*. https://doi.org/10.1007/978-3-319-52691-1

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch. *Communications of the ACM*, *52*(11), 60. https://doi.org/10.1145/1592761.1592779

Rich, K. M., Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational Thinking, Mathematics, and Science: Elementary Teachers'... *Journal of Technology and Teacher Education*, *27*(2), 165–205.

Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2019). Synergies and differences in mathematical and computational thinking: implications for integrated instruction. *Interactive Learning Environments*, *0*(0), 1–12. https://doi.org/10.1080/10494820.2019.1612445

Rich, K. M., & Yadav, A. (2019). Infusing computational thinking instruction into elementary mathematics and science: Patterns of teacher implementation. *Proceedings of the Society for Information Technology & Teacher Education International Conference 2019*, 330–334.

Rich, K. M., Yadav, A., & Larimore, R. A. (2020). Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. *Education and Information Technologies*. https://doi.org/10.1007/s10639-020-10115-5

Rich, K., Yadav, A., & Schwarz, C. (2019). Computational Thinking, Mathematics, and Science: Elementary Teachers' Perspectives on Integration. *Journal of Technology and Teacher Education*, *27*(2), 165–205.

Rich, P. J., Browning, S. F., Perkins, M. K., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2019). Coding in K-8: International Trends in Teaching Elementary/Primary Computing. *TechTrends*, *63*(3), 311–329. https://doi.org/10.1007/s11528-018-0295-4

Rich, P. J., Jones, B. L., Belikov, O., Yoshikawa, E., & Perkins, M. (2017). Computing and Engineering in Elementary School: The Effect of Yearlong Training on Elementary Teacher Self-Efficacy and Beliefs about Teaching Computing and Engineering. *International Journal of Computer Science Education in Schools*, *1*(1), n1.

Rich, P. J., Larsen, R. A., & Mason, S. L. (2020). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, *0*(0), 1–21. https://doi.org/10.1080/15391523.2020.1771232

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education*, *17*(1), 77–92. https://doi.org/10.15388/infedu.2018.05

Rocco, T. S., Hatcher, T. G., & Creswell, J. W. (2011). *The handbook of scholarly writing and publishing*. Retrieved from https://www.wiley.com/en-us/The+Handbook+of+Scholarly+Writing+and+Publishing-p-9780470393352

Rodrigo, R., & Ramirez, C. D. (2017). Balancing institutional demands with effective practice: A lesson in curricular and professional development. *Technical Communication Quarterly*, *26*(3), 314–328.

Rosenthal, R. (1994). Science and ethics in conducting, analyzing, and reporting psychological research. *Psychological Science*, *5*(3), 127–134.

Rosenthal, R., & Rosnow, R. L. (1984). Applying Hamlet's question to the ethical conduct of research: A conceptual addendum. *American Psychologist*, *39*, 561–563. https://doi.org/10.1037/0003-066X.39.5.561

Rowley, J. (2002). Using case studies in research. *Management Research News*, *25*(1), 16–27.

Rubin, H. J., & Rubin, I. S. (2011). *Qualitative interviewing: The art of hearing data*. sage.

Saidin, N. D., Khalid, F., Martin, R., Kuppusamy, Y., & Munusamy, N. A. P. (2021). Benefits and challenges of applying computational thinking in education. *International Journal of Information and Education Technology*, *11*(5), 248–254. https://doi.org/10.18178/ijiet.2021.11.5.1519

Sands, P., Yadav, A., & Good, J. (2018). Computational Thinking in K-12: In-service Teacher Perceptions of Computational Thinking. In *Computational Thinking in the STEM Disciplines* (pp. 151–164). Springer International Publishing. https://doi.org/10.1007/978-3-319-93566-9_8

SAP. (2021). *2.3 Million Young Africans Learnt Digital Skills during Africa Code Week 2018 - SAP Africa News Center*. https://news.sap.com/africa/2021/03/new-innovations-help-sap-africa-code-week-inspire-next-generation-of-african-coders/

Saunders, M., Lewis, P., & Thornhill, A. (2007). Research methods. *Business Students 4th Edition Pearson Education Limited, England*.

Saunders, M., Lewis, P., & Thornhill, A. (2009). Research methods for business students. Essex. *Financial Times/Prentice Hall*.

Schrum, M. L., Johnson, M., Ghuy, M., & Gombolay, M. C. (2020). Four years in review: Statistical practices of likert scales in human-robot interaction studies. *ACM/IEEE International Conference on Human-Robot Interaction*, 43–52. https://doi.org/10.1145/3371382.3380739

Schuler, D., & Namioka, A. (1993). *Participatory design: Principles and practices*. CRC Press.

Schwandt, T. A. (1993). *Constructivist, interpretivist approaches to human inquiry. Chapter 7, In, Denzin, NK, and Lincoln, YS (eds). Handbook of Qualitative Research*. Thousand Oaks, Sage.

Schwandt, T. A. (1997). *Qualitative inquiry: A dictionary of terms.* Sage Publications, Inc.

Seiter, L., & Foreman, B. (2013). *Modeling the learning progressions of computational thinking of primary grade students*. https://doi.org/10.1145/2493394.2493403

Selby, C. C. (2014). *How can the teaching of programming be used to enhance computational thinking skills?* [University of Southampton]. https://eprints.soton.ac.uk/366256/

Selby, C., & Woollard, J. (2014). *Refining an understanding of computational thinking*.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. *Computational Thinking in the STEM Disciplines*, 49–72.

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. https://doi.org/10.1007/s10639-012-9240-x

Sentance, S., & Csizmadia, A. (2015). Teachers ' perspectives on successful strategies for teaching Computing in school. *Ifip Tc3*, *June*, 1–11. http://community.computingatschool.org.uk/files/6303/original.pdf

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. https://doi.org/10.1007/s10639-016-9482-0

Sentance, S., & Humphreys, S. (2015). Online vs face-to-face engagement of computing teachers for their professional development needs. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, 69–81.

Settle, A., Goldberg, D. S., & Barr, V. (2013). *Beyond computer science*. https://doi.org/10.1145/2462476.2462511

Sherwood, H., Fancsali, C., Israel, M., Diamond, J., & Moeller, B. (2020). *Framework for Effective Computational Thinking Integration in Elementary (K–5) Schools*. *2*.

Sherwood, H., Yan, W., Liu, R., Israel, M., Fancsali, C., & Pierce, M. (2020). School-Wide Integration of Computational Thinking into Elementary Schools: A Cross-Case Study. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 1325. https://doi.org/10.1145/3328778.3372623

Shulman, L. S. (1986). Those Who Understand: Knowledge Growth in Teaching. *Educational Researcher*, *15*(2), 4–14. https://doi.org/10.3102/0013189X015002004

Simonsen, J., & Robertson, T. (2012). Routledge international handbook of participatory design. In *Routledge International Handbook of Participatory Design* (Issue 63070). Routledge. https://doi.org/10.4324/9780203108543

Sneider, C., Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational Thinking in High School Science Classrooms: Exploring the... *Science Teacher*, *81*(5), 53–59.

So, H.-J. J., Lossman, H., Lim, W.-Y. Y., & Jacobson, M. J. (2009). Designing an online video-based platform for teacher learning in Singapore. *Australasian Journal of Educational Technology*, *25*(3), 440–457. https://doi.org/10.14742/ajet.1144

Society, T. R. (2012). *Shut down or restart? The way forward for computing in UK schools.* https://royalsociety.org/~/media/Royal_Society_Content/education/policy/computing- in-schools/2012-01-12-Computing-in-Schools.pdf%0AVan

Spinuzzi, C. (2005). The methodology of participatory design. *Technical Communication*, *52*(2), 163–174.

Stanton, J., Goldsmith, L., Adrion, R., Dunton, S., Hendrickson, K. A., Peterfreund, A., Yongpradit, P., Zarch, R., & Dounay Zinth, J. (2017). *State of the States Landscape Report: State-Level Policies Supporting Equitable K–12 Computer Science Education | EDC*. https://www.edc.org/state-states-landscape-report-state-level-policies-supporting-equitable-k-12-computer-science

Statter, D., & Armoni, M. (2016). Teaching abstract thinking in introduction to computer science for 7th graders. *ACM International Conference Proceeding Series*, 80–

83. https://doi.org/10.1145/2978249.2978261

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research*. Sage publications.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20. https://doi.org/10.1007/s10798-015-9304-5

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. *Technology, Knowledge and Learning*, *22*(3). https://doi.org/10.1007/s10758-017-9328-x

Sürücü, L., & Maslakçi, A. (2020). Validity and reliability in quantitative research. *Business \& Management Studies: An International Journal*, *8*(3), 2694–2726.

Sysło, M. M., & Kwiatkowska, A. B. (2013). *Informatics for All High School Students*. 43–56. https://doi.org/10.1007/978-3-642-36617-8_4

Takahashi, S. (2011). Co-constructing efficacy: A “communities of practice” perspective on teachers’ efficacy beliefs. *Teaching and Teacher Education*, *27*(4), 732–741.

Talib, C. A., Aliyu, H., Zawadzki, R., & Ali, M. (2019). *Developing student’s computational thinking through graphic calculator in STEAM education*. 030003. https://doi.org/10.1063/1.5136371

Tankiz, E., & Atman Uslu, N. (2022). Preparing Pre-Service Teachers for Computational Thinking Skills and its Teaching: A Convergent Mixed-Method Study. *Technology, Knowledge and Learning 2022*, 1–23. https://doi.org/10.1007/S10758-022-09593-Y

Tashakkori, A., & Teddlie, C. (2003). *Handbook on mixed methods in the behavioral and social sciences*. Thousand Oaks, CA: Sage Publications.

Teddlie, C., & Tashakkori, A. (2003). Major issues and controveries inthe use of mixed methods in the social and behvioral sciences. *Handbook of Mixed Methods in Social & Behavioral Research*, 3–50.

Teddlie, C., & Tashakkori, A. (2009). *Foundations of mixed methods research: Integrating quantitative and qualitative approaches in the social and behavioral sciences*. Sage.

Theodoropoulos, A. (2022). Participatory design and participatory debugging: Listening to students to improve computational thinking by creating games. *International Journal of Child-Computer Interaction*, 100525. https://doi.org/10.1016/J.IJCCI.2022.100525

Tilya, F. (2018). *Information and Communication Technology and Educational Policies in the Sub-Saharan African Region*. 1301–1320. https://doi.org/10.1007/978-3-319-71054-9_90

Tobin, G. A., & Begley, C. M. (2004). Methodological rigour within a qualitative framework. *Journal of Advanced Nursing*, *48*(4), 388–396.

Toikkanen, T., Keune, A., & Leinonen, T. (2015). Designing Edukata, a participatory design model for creating learning activities. In *Re-engineering the Uptake of ICT in Schools* (pp. 41–58). Springer, Cham.

Tran, Y. (2018). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research*, *57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V., & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, *179*, 104425. https://doi.org/10.1016/J.COMPEDU.2021.104425

Tucker-Raymond, E., Cassidy, M., & Puttick, G. (2021). Science teachers can teach computational thinking through distributed expertise. *Computers & Education*, *173*, 104284. https://doi.org/10.1016/J.COMPEDU.2021.104284

Tuhkala, A. (2019). Participatory Design: an Approach for Involving Teachers as Design Partners. In *JYU Dissertations*. http://urn.fi/URN:ISBN:978-951-39-7782-5

Tulinius, C., Nielsen, A., Hansen, L., Hermann, C., Vlasova, L., & Dalsted, R. (2012). Increasing the general level of academic capacity in general practice: Introducing mandatory research training for general practitioner trainees through a participatory research process. *Quality in Primary Care*, *20*, 57–67.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of Women in Culture and Society*, *16*(1), 128–157.

Ung, L. L., Labadin, J., & Mohamad, F. S. (2022). Computational thinking for teachers: Development of a localised E-learning system. *Computers & Education*, *177*, 104379. https://doi.org/10.1016/J.COMPEDU.2021.104379

Uzumcu, O., & Bay, E. (2020). The effect of computational thinking skill program design developed according to interest driven creator theory on prospective teachers. *Education and Information Technologies*, *26*(1), 565–583. https://doi.org/10.1007/S10639-020-10268-3/TABLES/4

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly*, 21–54.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. https://doi.org/10.1007/s10639-015-9412-6

Wallet, P. (2015). *Measuring ICT in education in sub-Saharan Africa: A call for action.* http://uis.unesco.org/sites/default/files/documents/information-and-communication-technology-ict-in-education-in-sub-saharan-africa-2015-en.pdf

Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems*, *4*(2), 74–81.

https://doi.org/10.1057/ejis.1995.9

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2019). Integrating Computational Thinking into Elementary Science Curriculum: an Examination of Activities that Support Students' Computational Thinking in the Service of Disciplinary Learning. *Journal of Science Education and Technology*, *29*(1), 53–64. https://doi.org/10.1007/s10956-019-09801-y

Webb, M., Davis, N., Bell, T., Katz, Y., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, *22*(2). https://doi.org/10.1007/s10639-016-9493-x

Weese, J. L., & Feldhausen, R. (2017). STEM outreach: Assessing computational thinking and problem solving. *ASEE Annual Conference and Exposition, Conference Proceedings*, *2017-June*. https://doi.org/10.18260/1-2--28845

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., & Wilensky, U. (2013). Robobuilder: a computational thinking game. *SIGCSE*, 736.

Wicks, A. C., & Freeman, R. E. (1998). Organization studies and the new pragmatism: Positivism, anti-positivism, and the search for ethics. *Organization Science*, *9*(2), 123–140.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725.

Wing, J. M. (2011). No Title. *The Link - The Magazine of Carnegie Mellon University's School of Computer Science*, 1. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yadav, A., Krist, C., Good, J., Caeli, E. N., & Krist, C. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, *28*(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019). Integrating computational thinking in elementary classrooms: Introducing a... *Society for Information Technology & Teacher Education International Conference*, *2019*(1), 347–350. https://www.learntechlib.org/primary/p/208366/

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, *14*(1), 1–16. https://doi.org/10.1145/2576872

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55–62. https://doi.org/10.1145/2994591

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11*, 456. https://doi.org/10.1145/1953163.1953297

Yin, R. K. (1994). Discovering the future of the case study. Method in evaluation research. *Evaluation Practice*, *15*(3), 283–290.

Yin, R. K. (2009). *Case study methods: Design and methods*. Thousand oaks: SAGE.

Yonazi, E., Kelly, T., Halewood, N., & Blackman, C. (2012). *The transformational use of information and communication technologies in Africa*. World Bank.

Zha, S., Jin, Y., Moore, P., & Gaston, J. (2020). A cross-institutional investigation of a flipped module on preservice teachers' interest in teaching computational thinking. *Journal of Digital Learning in Teacher Education*, *36*(1), 32–45. https://doi.org/10.1080/21532974.2019.1693941

Zhang, L. (2020). *Integrating Computational Thinking into Swedish Compulsory Education with Block-Based Programming A case study from the perspective of teachers* (Issue 20). Stockholm University.

Zhao, Y., Byers, J., Mishra, P., Topper, A., Chen, H., Enfield, M., Ferdig, R., Frank, K., Pugh, K., & Tan, S. H. (2001). What do they know? A comprehensive portrait of exemplary technology-using teachers. *Journal of Computing in Teacher Education*, *17*(2), 24–36.

APPENDICES

Appendix A: Ethics Clearance

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimošo

Reference number: EBIT/159/2020

Ms MM Ausiku
Department: EBIT Dean's Office
University of Pretoria
Pretoria
0083

Dear Ms MM Ausiku

FACULTY COMMITTEE FOR RESEARCH ETHICS AND INTEGRITY

Your recent application to the EBIT Research Ethics Committee refers.

Approval is granted for the application with reference number that appears above.

1. This means that the research project entitled "Preparing primary school teachers in developing countries to teach computational thinking: A Namibian Case Study" has been approved as submitted. It is important to note what approval implies. This is expanded on in the points that follow.
2. This approval does not imply that the researcher, student or lecturer is relieved of any accountability in terms of the Code of Ethics for Scholarly Activities of the University of Pretoria, or the Policy and Procedures for Responsible Research of the University of Pretoria. These documents are available on the website of the EBIT Research Ethics Committee.
3. If action is taken beyond the approved application, approval is withdrawn automatically.
4. According to the regulations, any relevant problem arising from the study or research methodology as well as any amendments or changes, must be brought to the attention of the EBIT Research Ethics Office.
5. The Committee must be notified on completion of the project.

The Committee wishes you every success with the research project.

Prof K.-Y. Chan
Chair: Faculty Committee for Research Ethics and Integrity
FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

Appendix B: Permission Letter from Khomas Directorate of Education

REPUBLIC OF NAMIBIA

KHOMAS REGIONAL COUNCIL
DIRECTORATE OF EDUCATION, ARTS AND CULTURE

Tel: [09 264 61] 293 4356
Fax: [09 264 61] 231 367
Enquiries: Gerard N. Vries Tel: +264 811446544) Email: gerardv120@gmail.com

Private Bag 13236
WINDHOEK

File No: 17/6/9/2

Ms. Maria Ausiku
P.O Box 2921
Windhoek
Cell: +264 814458924
Email: kamweshi@gmail.com

REQUEST FOR PERMISSION TO CONDUCT RESEARCH IN THE KHOMAS DIRECTORATE OF EDUCATION, ARTS AND CULTURE

Your letter dated 15 June 2020 on the above topic refers.

Permission is hereby granted to you to do research for your PhD (Information Technology) under the title: "Preparing primary school teachers in developing countries to teach Computational Thinking: A Namibian Case Study" which will focus on professional development of teachers in preparation for computational thinking integration within existing subjects at primary level. The study will target senior primary teachers in the Khomas Region.

The following conditions must be adhered to:

- Teaching by the respective teachers should not be disrupted;
- Teachers who will take part in the research should do so voluntary;
- A copy of your thesis with your findings/recommendations must be provided to the Directorate of Education, Arts and Culture, Khomas Regional Council.

I trust this confirmation will suffice.

Yours sincerely,

19/06/2020

MINISTRY OF EDUCATION
ARTS AND CULTURE
PRIVATE BAG 13236 WINDHOEK
19 -06- 2020
DIRECTOR
KHOMAS REGION

....
Gerard N Vries
Director of Education, Arts and Culture

Appendix C: Consent Form for Participants

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

Informed Consent Form

1. Project information

1.1 Title of research project

Preparing primary school teachers in developing countries for CT teaching: A Namibian Case Study

1.2 Researcher details

Researcher Name: Maria Ausiku

Department: Informatics

Contact Details: Tel: +264 81445892; Email: kamweshi@gmail.com

1.3 Research study description

I am a PhD student at the University of Pretoria, currently conducting research on preparation of teachers for teaching CT skills in primary schools in Namibia. As a participating teacher, you will be requested to complete a pre- and post-questionnaire and attend a professional development programme that will introduce you to CT. You will also be requested to keep a journal where you write down your thoughts on CT and be interviewed at the end of the programme. The interviews will be voice-recorded and a pseudonym will be used instead of real names. The study will be conducted in a safe school environment and no physical or emotional harm will be caused to the participants.

2. Informed consent

2.1 I, ____ hereby voluntarily grant my permission for participation in the project as explained to me by Maria Ausiku.

2.2 The nature, objective, possible safety and health implications have been explained to me and I understand them.

2.3 I understand my right to choose whether to participate in the project and that the information furnished will be handled confidentially. I am aware that the results of the investigation may be used for the purposes of publication.

2.4 Upon signature of this form, the participant will be provided with a copy.

Signed: ____________________________ Date: ________________

Witness: ____________________________ Date: ________________

Researcher: ____________________________ Date: ________________

Appendix D: Pre-Questionnaire

Pre-Workshop Questionnaire

Preparing primary school teachers in developing countries for computational thinking teaching.

Thank you for participating in the Computational Thinking Training Workshop. Before starting the workshop, you are requested to complete the following survey. The survey includes:

- Demographic information about your teaching background
- Questions about your understanding of computational thinking

The purpose of this research, the nature of the data that will be collected and how it will be protected is detailed in the consent form that was provided to you.

SECTION A

1. Demographics Information

Grades Currently Taught	
Subjects Currently Taught	
Years of Teaching Experience	

SECTION B

Note: *All items use a five-point Likert scale with options of: Strongly Agree (**SA**), Agree (**A**), Neutral (**N**), Disagree (**D**), Strongly Disagree (**SD**).*

2. CT Knowledge Comprehension

Item #	Statement	SD	D	N	A	SA
1.	I can define what computation thinking (CT) is.					
2.	I can describe fundamental computational thinking concepts (e.g., algorithms, abstraction, decomposition, pattern recognition & evaluation).					
3.	I can describe fundamental coding/programming concepts (e.g., loops, variables, conditional logic).					
4.	I can look at a process and figure out how to make it more efficient.					
5.	I can suggest different solutions in order to solve problems.					
6.	I can generalise solutions that can be applied to many different problems.					
7.	I am good at finding patterns in data.					

8.	I am good at solving puzzles.					
9.	I can read a formula (e.g., algorithm, equation, input/output process) and explain what it should do.					
10.	When I'm presented with a problem, I can easily break it down into smaller steps.					
11.	When solving a problem, I work with others to solve different parts of the problem at the same time.					
12.	When solving a problem, I look how information can be collected, stored, and analysed to help solve the problem.					
13.	When solving a problem, I create a solution where steps can be repeated.					
14.	When solving a problem, I create a solution where some steps are done only in certain situations.					
15.	When solving a problem, I try to simplify the problem by ignoring details that are not needed.					

3. Value Beliefs towards Computational Thinking

Item #	Statement	SD	D	N	A	SA
1.	Computing should be taught in primary schools.					
2.	Learning about computing can help primary school learners become more engaged in school.					
3.	Computing is like art—you are either born with the ability to think that way or you are not.					
4.	Computing content and principles can be understood by primary school children.					
5.	My current teaching situation does lend itself to teaching computing concepts to my learners.					
6.	Knowledge of computer programming is needed in most careers.					
7.	Providing more computational thinking activities is necessary to enrich my learners' overall learning.					
8.	Computational thinking is an important 21st-century skill.					
9.	My current primary school learners are going to need to know how to apply computing concepts to remain competitive for jobs by the time they are adults.					

4. Self-Efficacy for Computational Thinking

Item #	Statement	SD	D	N	A	SA
1.	I feel confident using computer technology.					
2.	I feel confident writing simple instructions for another person on paper.					
3.	I know how to teach computing concepts effectively without the use of a computer.					

4.	I know how to teach programming concepts effectively without the use of a computer.					
5.	I can promote a positive attitude towards computing education to my learners.					
6.	I can guide learners in using programming as a tool while we explore other topics.					
7.	I feel confident using programming as an instructional tool within my classroom.					
8.	I can adapt lesson plans incorporating unplugged activities as an instructional tool.					
9.	I can adapt lesson plans incorporating programming as an instructional tool.					
10.	I can identify how computational thinking concepts relate to the syllabus.					

Appendix E: Post Questionnaire

Post-Workshop Questionnaire

Preparing primary school teachers in developing countries for computational thinking teaching: A Namibian Case Study

Thank you for participating in the Computational Thinking Training Workshop. After attending the workshop, you are requested to complete the following survey. The survey includes:

- Demographic information about your teaching background.
- Questions to measure how your perceptions about the meaning of CT, beliefs and attitudes toward CT and potential integration into classrooms have changed after the intervention.

The purpose of this research, the nature of the data that will be collected and how it will be protected is detailed in the consent form that was provided to you.

SECTION A

5. Demographics Information

Grades Currently Taught	
Subjects Currently Taught	
Years of Teaching Experience	

SECTION B

Note: *All items use a five-point Likert scale with options of: Strongly Agree (SA), Agree (A), Neither Agree nor Disagree (N), Disagree (D), Strongly Disagree (SD).*

6. CT Knowledge Comprehension

Item #	Statement	SD	D	N	A	SA
16.	I can define what computation thinking (CT) is.					
17.	I can describe fundamental computational thinking concepts (e.g., algorithms, abstraction, decomposition, pattern recognition & evaluation).					
18.	I can describe fundamental coding/programming concepts (e.g., loops, variables, conditional logic).					
19.	I can look at a process and figure out how to make it more efficient.					
20.	I can suggest different solutions in order to solve problems.					
21.	I can generalise solutions that can be applied to many different problems.					
22.	I am good at finding patterns in data.					

23.	I am good at solving puzzles.					
24.	I can read a formula (e.g., algorithm, equation, input/output process) and explain what it should do.					
25.	When I'm presented with a problem, I can easily break it down into smaller steps.					
26.	When solving a problem, I work with others to solve different parts of the problem at the same time.					
27.	When solving a problem, I look how information can be collected, stored, and analysed to help solve the problem.					
28.	When solving a problem, I create a solution where steps can be repeated.					
29.	When solving a problem, I create a solution where some steps are done only in certain situations.					
30.	When solving a problem, I try to simplify the problem by ignoring details that are not needed.					

7. Value Beliefs

Item #	**Statement**	**SD**	**D**	**N**	**A**	**SA**
10.	Computing should be taught in primary schools.					
11.	Learning about computing can help primary school learners become more engaged in school.					
12.	Computing is like art—you are either born with the ability to think that way or you are not.					
13.	Computing content and principles can be understood by primary school children.					
14.	My current teaching situation does lend itself to teaching computing concepts to my learners.					
15.	Knowledge of computer programming is needed in most careers.					
16.	Providing more computational thinking activities is necessary to enrich my learners' overall learning.					
17.	Computational thinking is an important 21st-century skill.					
18.	My current primary school learners are going to need to know how to apply computing concepts to remain competitive for jobs by the time they are adults.					

3. Self-Efficacy for Computational Thinking

Item #	Statement	SD	D	N	A	SA
11.	I feel confident using computer technology.					
12.	I feel confident writing simple instructions for another person on paper.					
13.	I know how to teach computing concepts effectively without the use of a computer.					
14.	I know how to teach programming concepts effectively without the use of a computer.					
15.	I can promote a positive attitude towards computational thinking to my learners.					
16.	I can guide learners in using programming as a tool while we explore other topics.					
17.	I feel confident using programming as an instructional tool within my classroom.					
18.	I can adapt/create lesson plans incorporating unplugged activities as an instructional tool.					
19.	I can adapt/create lesson plans incorporating programming as an instructional tool.					
20.	I can identify how computational thinking concepts relate to the syllabus.					

a. Items Aligned with Programming Concepts

Item #	Statement	SD	D	N	A	SA
I can create a computer program which …						
1.	executes a step-by-step sequence of commands					
2.	uses loops to repeat commands					
3.	responds to events like pressing a key on the keyboard					
4.	does more than one thing at the same time					
5.	only executes some commands when a specific condition is met					
6.	perform arithmetic operations like addition and subtraction					
7.	can store, update, and retrieve values					
8.	can ask the user a question					

b. Items Aligned with CT Practices

Item #	Statement	SD	D	N	A	SA
When creating a computer program, I …						
1.	make improvements one step at a time and incorporate new ideas as I have them.					
2.	run my programme frequently to make sure it does what I want, and fix any problems I find.					
3.	share my programmes with others and look at others' programmes for ideas.					
4.	break my programme into multiple parts to carry out different actions.					

5. Teaching Self-Efficacy for Computational Thinking (TSECT)

Item #	Statement	SD	D	N	A	SA
With my future classes …						
1.	I can explain basic computing concepts to children (e.g., algorithms, loops, conditionals, functions, variables, debugging, pattern-finding).					
2.	I can help learners debug their computer programs.					
3.	I can integrate unplugged activities into my current curriculum.					
4.	I can integrate computer programming into my current curriculum.					
5.	I know where to find the resources to help learners learn CT skills.					
6.	I believe that I have the necessary computational thinking skills to integrate computing content into my class lessons.					
7.	I can recognise and appreciate computational thinking concepts in all subject areas.					
8.	I can create computational thinking activities at the appropriate level for my learners.					
9.	I can explain how computing concepts are connected to daily life.					
10.	I can develop and plan effective computational thinking lessons.					

Appendix F: Interview Guide

Interview Guiding Questions

The purpose of this research, the nature of the data that will be collected and how it will be protected is detailed in the consent form that was provided to you. Your real names will not be used for anonymity.

The interview will get an in-depth evaluation of whether the training workshop contributed to the teacher's understanding and self-efficacy of computational thinking (CT) and their future plans for integration as well as the anticipated challenges.

A. CT Knowledge

1. In your view, what is computational thinking and what does it involve?
2. What computational thinking skills are developed by the lesson plans and activities that you designed?
3. From what you have learned now, can you identify how your existing lessons may already have alignment with the CT concepts?

B. CT Integration

4. How would you integrate CT into your curriculum based on your experience of this professional development programme?
5. What types of activities and teaching strategies to integrate CT do you find relevant for your school context and why?

C. Self-efficacy

6. How confident are you about developing the students' computational thinking capabilities?
7. What prevents you from feeling confident about developing your students' computational thinking capabilities?
8. What could help you to feel more confident about developing your students' computational thinking capabilities?

D. Challenges

9. What challenges do you expect to face in teaching computational thinking skills at your school after this professional development programme?
10. How do you think the said challenges can be overcome?

Appendix G: Self-reporting Journal

Daily Journal

Participant Name (pseudonym) ……………………….   Date: …………………

Overview of the session

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

What did you learn?

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

What did you find challenging?

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………………………………

Any comments or suggestions?

…………………………………………………………………………………………………
…………………………………………………………………………………………………
…………………………………………………………………………

Appendix H: Existing Lesson Plan & Activity

Pattern-finding and building (Mason, 2018)

Subject	Pattern-finding
Grade level	K-6
Suggested time	15-30 minutes
Objectives	Identify repeating patterns.
Learning targets	• I can find repeating patterns. • I can make repeating patterns.
Background, Overview	Programmers need to be able to identify and use patterns to simplify their code. In this lesson, students will practice identifying and making patterns.
Materials needed	• Building blocks in assorted shapes and colours • Manipulatives for students to use to make patterns
Curriculum connections	Patterns (math, art, science, music, language)

Activity

Anticipatory Set	• Make a simple pattern using building blocks. • Ask the students: 　○ What comes next? 　○ How do you know? 　○ What's the pattern?
Guided Practice	• Make a few more patterns, of increasing complexity. • Ask the students: 　○ What comes next? 　○ How do you know? 　○ What's the pattern?
Independent Practice	• Put students in pairs and have them work together to make patterns, • Or let them take turns making patterns and having the partner figure out what comes next.
Assessment	• Check students' patterns. • Have students self-assess their progress toward the learning targets.

Appendix I: Blank Lesson Plan Template

Lesson Plan Template

Grade: **Subject:**	**Which key learning topic is this lesson plan for?**
Syllabus Outcome(s): *What do students learn and are able to do as a result of this lesson?*	
Introduction: *How will you get the students motivated, curious and ready to learn?*	
Metalanguage: *What are the key concepts or procedures that you want students to understand as a result of this lesson?*	
Computational Thinking: *Which of the computational concepts, practices and perspectives will students have the opportunity to learn about in the lesson?*	
Teaching Activities: *What strategies will you use to teach the content and skills? How long will you spend on each of those strategies and with the content? How would you address different levels or prior knowledge?*	
Lesson Conclusion: *How will you bring the lesson to a conclusion?*	
Assessment: *How will you know whether the students achieved what you wanted them to achieve?*	
Resources: *What materials do you need for this lesson? Have you used ideas from elsewhere?*	

Appendix J: Workshop learning material & tools guide

Participatory CT Workshop Material Guide

Aim of the PD Workshop	
Workshop stages	
Workshop Components	
Activities	
Time	
Venue	
Participants	
Tools Needed	
Timetable of the PD Workshop	
Workshop sessions in detail	

Participatory Design (PD) Workshop

Aim of the PD Workshop

The aim of the workshop is for teachers to:

- Learn what Computational *Thinking (CT)* is.
- Discover ways of teaching *Computational Thinking Concepts* without using a computer (unplugged).
- Solve a variety of *problems* using *unplugged and plugged activities.*
- Work together with other teachers to develop lessons and activities that involves the CT concepts that they have learnt about during the workshop.
- Reflect on the activities & lesson plans and give feedback.
- Refine the activities & lesson plans.

Activities used in the workshop are adapted from (Computing at School, 2014), (BEBRAS, n.d.), (*CS Unplugged*, n.d.) and (Code.org, n.d.).

Workshop stages

Stage 1 - Pre-Workshop Survey: A pre-workshop survey will be conducted to measure the teachers' perceptions about the meaning of CT, their beliefs and attitudes toward CT and potential integration into classrooms.

Stage 2 - PD Workshop: A participatory design workshop designed with Professional Development Framework for Computational Thinking (**PD4PCT)** as a guiding framework will involve teachers as design partners. Table 1 below shows the important guiding concepts taken from the framework.

Table 1: Framework concepts mapped to the study

Content	**Active Learning**	**Coherence**	**Collective Participation**	**Duration**
✓ CT Definition ✓ CT Concepts	✓ CT teaching Strategies	✓ Skills & Beliefs	✓ Establish groups ✓ Establish roles ✓ Design lesson plans ✓ Provide feedback	✓ 5 Weeks ✓ 30 Hours
Context ✓ School leadership ✓ Technological infrastructure				

Stage 3 - Post-Workshop Survey: A post-workshop survey will be conducted to measure how the teachers' perceptions about the meaning of CT, their beliefs and attitudes toward CT and potential integration into classrooms have changed after the intervention.

Stage 4 - Interviews: In-depth group interviews with participants of the PD workshop will be conducted at the end of the workshop to get more details on the challenges, successes and the future of CT integration.

Workshop Components

Activities

- Getting familiar with CT concepts
- Designing lesson plans via participatory design

Time

- The PD workshop will involve three after-school sessions per week, which will run from 2:00pm – 4:00pm for 5 weeks.

Venue

- Training will take place in a classroom or computer lab at one of the participating schools.

Participants

- Primary school teachers from participating schools will attend the training workshop.

Tools Needed

- Lesson plans examples
- Lesson plans templates
- Diaries / daily logs
- Blank papers
- Pens/ Colour pencils
- Flip Charts
- Marker pens
- Coins
- Scissors
- Computers with Scratch software

Timetable of the PD Workshop

Time	Session
Day 1: 11 October 2021	
2:00pm - 2:15pm	Introductions: Researcher, Study & Aims of Workshop
2:15pm - 2:30pm	Registration & Signing of Consent Forms
2:30pm - 3:00pm	Introduction to Computational thinking (CT)
3:00pm - 3:30pm	CT Concepts/Skills
3:30pm - 4:00pm	CT Approaches
Day 2: 12 October 2021	
2:00pm - 3:00pm	CT Teaching strategies
3:00pm - 4:00pm	Key programming concepts
Day 3: 13 October 2021	
2:00pm - 2:15pm	Unplugged strategy
2:15pm - 4:00pm	Hands-on Unplugged Activities
Day 4 : 18 October 2021	
2:00pm - 3:00pm	Programming strategy - Introduction to Scratch
3:00pm - 4:00pm	Scratch Concepts & other programming concepts
Day 5: 19 October 2021	
2:00pm - 4:00pm	Hands-on Scratch activity
Day 6: 21 October 2021	
2:00pm - 4:00pm	Hands-on Scratch activity
Day 7 – 9: 25, 26, 28 October 2021	
2:00pm - 4:00pm	Design Lesson Plans & Activities

Day 10: 01 November 2021	
2:00pm - 4:00pm	Lesson plan presentation for groups
Day 11: 02 November 2021	
2:00pm - 4:00pm	Reflect & Refine Lesson Plans & Activities
Day 12: 03 November 2021	
2:00pm - 4:00pm	Interviews

Workshop sessions in detail

DAY 1

In this session, first the introduction of the facilitator, the study and the main aims of the workshop will be done. The facilitator will then explain what Computational *Thinking* (CT) is, its concepts and approaches to how they can be integrated into the existing curriculum.

Presentation:

a. **Introduction to Computational thinking (CT)**

CT is a thought process for solving problems through the application of fundamental computing concepts (CT skills), regardless of the domain with or without a computer.

b. **CT Concepts/Skills – Definition & Examples**

- Abstraction
- Decomposition
- Algorithm
- Pattern Recognition
- Evaluation

CT Skills	Descriptions and examples
Algorithms	**Definition:** Algorithmic thinking is the skill involved with creating an algorithm. An algorithm is a series of ordered, logical and unambiguous rules or instructions necessary to solve a problem or achieve an objective. By identifying steps that can be communicated as instructions (verbal or written), codes or programmes to other people or to computing devices, algorithmic thinking skills are employed. **Examples:** • What steps are needed to make a sandwich? Does it matter if the steps are done in a different order? Why or why not? • Create an algorithm that would help you decide what to wear each day. • Create an algorithm to teach a young child how to tie shoelaces in a bow. **Associated with:** • Planning, Organising, Sequencing, Classifying, Sorting
Abstraction	**Definition:** Abstraction is about reducing the complexity of a problem or task by focusing on what is important, capturing relevant information and removing unnecessary details. Abstraction can also be used to have one object stand for many, or to have a word stand for an action. Models and simulations can also be considered abstractions. **Examples:** • Identify the most important skills for a soccer player. • Represent the actions run, stop and hop using images. • Code a simulation of a volcano erupting. • Identify the appropriate formula to use to solve a physics problem. **Associated with:** • Deconstructing, Dividing, Sorting, Classifying

Decomposition	**Definition:** Decomposition involves breaking down a problem into smaller parts or sub problems. Working with smaller subsets of a problem can reduce the overall complexity of a problem. Decomposition may also involve thinking about computational products in terms of their component parts (e.g., graphics, data, and user interface). Not only does CT involve decomposing a problem, it also involves applying knowledge of how previous problems were solved and composing a solution to the problem after all of the sub problems have been tackled. **Examples:** • Identify all of the tasks you will need to do in order to bake a cake. • Explain how the parts of a dragonfly make it an effective predator. • Create a video game with your friends. How could you divide up the tasks? • Create a code that will convert a value in Fahrenheit to one in Celsius. What are the major parts and processes? **Associated with:** • Deconstructing, Dividing, Sorting, Classifying
Pattern Recognition	**Definition:** Pattern recognition involves being able to recognise and use patterns to describe and represent sequences in data or processes. By identifying patterns, predictions can be made as to how things might work or what might happen in a given circumstance. Identifying patterns enables the creation of rules, such as when actions can be repeated automatically. Pattern recognition also enables previously successful methods to be applied to new problems, which can improve the efficiency of the problem-solving process. **Examples:** • What pattern do you notice when you draw a square? How could you apply that knowledge to drawing a pentagon? • Which notes are repeated in the music? When are those parts repeated? • In soccer, what happens after you receive one yellow card? What about two yellow cards? **Associated with:** • Observing, Predicting, Comparing, Generalising

Evaluation	**Definition:** Evaluation is about using critical thinking and judgement to determine if a set of criteria are met. If not, then correction or improvement may be required. Testing involves trying something and observing what happens. Computer programming tends to be an interactive process in which testing and evaluating is constantly occurring. **Examples:** • Did the animated character move in the way you expected it to? • What happened when you tested the robot? Was it able to shoot the ball into the net every time? • What happened when you downloaded the code for scrolling your name onto the micro: bit? Did it scroll the letters of your name across the LED display in the correct order? **Associated with:** • Observing, Comparing, Analyzing

c. **CT Approaches**

CT Approach	**Definition**
Tinkering	Changing things to see what happens
Creating	Designing and making
Debugging	Finding and fixing errors
Persevering	Keeping going
Collaborating	Working together

Activity/Material:

- Computing at School CT Poster
- BEBRAS CT Cards

DAY 2

This session will introduce how CT concepts can be taught using different strategies. It will also introduce three key computational/programming concepts, which will also apply to other sessions.

Presentation:

a. Teaching Strategies for CT

Different types of teaching strategies:

- Unplugged
- Programming
- Robotics
- Game design
- Project-based

Presentation

b. Three key programming concepts

- Sequences
- Loops
- Conditions

Sequences

Definition:

The steps in an algorithm always follow a sequence. It should be noted that sequences are rarely linear and may involve tasks which are repeated or only occur under certain conditions. A flowchart is a good tool for understanding and representing sequences. Simple, linear sequencing is one of the first concepts learned when coding.

Examples:

- What are the steps you need to do to……? (e.g., change a tyre, brush your teeth, make toast). Are there any steps that you repeat?
- Writing out the rules for a game (e.g., what happens first, next, last)
- Describing how you get to school using a map (e.g., where do you start? When do you turn? Where do you end?)
- Explain to someone else how to draw a square of a specific perimeter.

Activity: Origami without Instructions

- Individual | Materials: Square paper (origami paper preferred)

Repetition (Loops)

Definition:

Repetition involves repeating a step or steps in an algorithm a certain number of times until a certain pre-determined end point is reached. Repetitive tasks are very common in computer programming and setting up tasks so that they automatically repeat (loop) can save a lot of time. Loops help programmes be more organized and shorter because they cut down on the amount of code or instructions that needs to be written.

Examples:

- Explain to someone else the most efficient way of making 100 sandwiches.
- Set up a light to turn on and off at certain times every day.
- Code a car game so that the car will automatically go around the track 10 times.

- Create a piece of art that involves a repeating pattern, like a mosaic tile or fractal.

Activity: A Loopy Routine

- Individual, Pairs, or Groups | Materials: Paper and pencil (optional)

Conditionals

Definition:

At times in algorithms, there may be a need to select one action out of a set of actions, such as which way to go at an intersection in the road. Conditional statements give rules to direct the flow of what happens, such as if something is true, then something will happen or else something else will happen. Conditionals allow a programme to make decisions and direct the flow of activities without human intervention.

Examples:

- If it is raining outside then you will need to put on your rain boots, else wear your running shoes.
- If the sensor reads a temperature greater than 25ºC, then show a happy face display, else if the sensor reads a temperature less than 25ºC then show a sad face display.
- If you get to a fork in the maze then turn left else go straight on.
- If (hour < 18:00) {greeting = “Good day”;} else {greeting = “Good evening”:}

Activity: If This, Then…Art!

- Individual or Pairs | Materials: Paper, drawing supplies

DAY 3

This session will introduce the *unplugged* strategy. As you may know, Computer Science (CS) is not just about Coding! Computational Thinking and Algorithmic Thinking are at the core of CS and many of its principles can be taught without the aid of a computer. The activities in this session could be useful for introducing **computational concepts** away from a computer. They could also be used when computers are not working or when computer labs are not available at all.

Presentation:

Unplugged strategy

Definition:

Computational Thinking does not need a computer! Many activities that develop computational thinking can be done "unplugged," meaning away from computers. Playing games with rules, doing logic puzzles and creating and following recipes are all ways that Computational Thinking can be done "unplugged."

Examples:

• Paper programming

• Drawing maps or completing mazes

• Playing games where one person tells someone else what to do or where to go

Activity:

DAY 4-6

This session teach about how you can teach CT concepts through the programming/coding strategy using Scratch which is a drag and drop visual programming language.

Presentation:

- Programming strategy
- Introduction to Scratch

Scratch Concepts

- **Block**

A command which tells the sprite what to do. It can be run by clicking on it.

- **Costumes**

Are alternative ways that a sprite can look on the stage.

- **Hat Block**

Hat blocks *are always placed at the top of a script. They are referred to as* ***Hat blocks*** *because they are shaped like hats. An example of a* ***hat block*** *is the when green flag clicked block.*

- **Move**

A command which makes the sprite change its position.

- **Next costume**

A command which switches to the ***next costume*** *in the list of the sprite's costumes. The* ***next costume*** *after the last one is the first one in the list again.*

- **Repeat block**

A ***repeat block*** *is a block which runs the blocks inside a specified number of times.*

- **Script**

A ***script*** *is a sequence of blocks snapped together, a programme. It can be run by clicking on any part of the script.*

- **Sprite**

A ***sprite*** *is an object we control using our blocks and scripts. For example, the Cat in the Dancing Cat programme is a* ***sprite****.*

- **Stage**

The ***Stage*** *is the area where you can see the sprites.*

- **Stamp**

A ***stamp*** *block is a block which tells the sprite to print its image on the stage.*

- **Turn**

The ***turn*** *block is a command which makes the sprite change its direction.*

- **Wait**

*The **wait** block is a command which waits a specified number of seconds, e.g. 1, 2 or 0.2, then continues with the next blocks.*

Other coding concepts

1. **Variables**

Definition:

Variables are places where you can store and retrieve data. Variables have names that stand in for the data they hold, which makes them an abstract concept since one thing (variable name) represents something else (data). Variables also have a data type (the kind of data that can be stored) and a value (refers to what is stored in the variable). In coding languages, variables are written as case-sensitive single words with no spaces.

Examples:

- var a = 10; var b = 5; a and b are the names of the variables and the = tells you what data is stored in each variable.
- The variable "age" could represent people's ages (e.g., var age;).
- The variable "rateGrowth" could be used to keep track of a plant's growth over a certain period of time (e.g., var rateGrowth;).
- The variable "temperature" could be used to store and retrieve the temperature readings from a sensor.
- The variable "StudentName" could be used to store a student's name and insert it automatically in a personalised email.

2. **Events**

Definition:

An event involves having one action cause another action, such as how you get a computer to respond to the input of a user. User inputs include actions such as clicking a mouse, tapping a key or touching a screen.

Examples:

- Zooming in on a map when someone clicks on it
- Scrolling through images on a phone when someone swipes the screen
- Adjusting the volume on a video when someone presses the + or - buttons E

3. Functions

Definition:

Functions group all the steps of a complex action into one command (e.g., brush teeth). Functions are especially useful for defining a sequence of commands that can be reused, for example how to turn a robot sideways by 90 degrees. The creation of functions is based on pattern recognition, abstraction and logical thinking.

Examples:

- Function: Brush Teeth (involves turning on the water, putting toothpaste on toothbrush, etc.)
- Function: Draw a Square (draw a straight line of 3 cm, turn 90º, draw a straight line of 3 cm, turn 90º, draw a straight line of 3 cm, turn 90º, draw a straight line or 3 cm, turn 90º)
- Function: Header style

4. Inputs & Output

Definition:

Input and output (often abbreviated as I/O) is the communication that occurs between humans and computers, computers and other computers, among processes within a computer, or between a computer/robot and its environment. Humans interact with computers using devices called peripherals. When computers interact with other computers it is often over networks such as the Internet. I/O occurs at all levels of the computer’s operation.

Examples:

- Humans provide input into computers using peripherals such as a keyboard, mouse, webcam, microphone, etc.
- Computers provide output using speakers, a screen, printers, etc.
- Robots use input devices such as sensors (light sensor, temperature sensor, etc.), buttons, etc.
- Robots' output devices include motors, speakers, actuators, lights, etc.

Activity:

DAY 7 - 9

In this session the group will collaborate in grade or subject teams to create lesson plans and activities that infuse CT skills into the existing curriculum.

Steps

1. Group teachers in small teams per subject or grade level
2. Start off with showing the teachers an existing example of a CT-infused lesson plan.
3. Provide teams with a blank template of the lesson plan
4. Each team selects a unit topic, defines learning objectives, brainstorms methods to 'CT-infuse' the topic.
5. The teams develop an overview and unit plan, and then begin developing the unit's contents.

DAY 10

In this session the groups will reflect on the artefacts created and provide feedback.

1. The different teams will present their artefacts to all the participants.
2. Teachers and researcher discuss and critically examine the designed lesson plans and provide feedback.

DAY 11

In this session the teams will iteratively improve the lesson plans based on the feedback given.

DAY 12

In this session the researcher will:

- Conduct the semi-structured interviews with the participants in 2/3 groups depending on the number.
- Collect the daily self-documenting journals from the participants

Unplugged Activities

Activity	Skills	Objective	Overview	Materials Needed
Draw a crab	Abstraction, algorithms, rules, cause-and-effect	Identify (abstract) a model. Use models to write an algorithm.	In this activity, the group will look at a simple picture of a crab and write instructions for how to draw it. Then they will write their own instructions of how to draw something.	Drawing materials (paper, pencils)
Program me a partner	Decomposition	Understand how to break down a task into individual steps.	In this activity, teachers will verbally direct other teachers to special destinations in the classroom (e.g., to a bookcase or a closet).	Paper and pen to write their instructions
Patterns in nature	Pattern-finding	Identify and describe patterns in nature	Programmers need to be able to identify and use patterns to simplify their code. In this lesson, teachers will practice identifying patterns in nature and create paper snowflakes that contain patterns.	• Natural examples of patterns photos): ○ Leaves ○ Shells ○ Flowers ○ Butterflies ○ Snowflakes • Paper for making snowflakes • Scissors
The Swap Puzzle	• Algorithms • Evaluation • Efficiency	*Solve a puzzle, coming up with an algorithm that your team can follow faster than anyone else.*	This introduces the idea of the solution to a problem being a set of instructions that allow others to 'solve' it with no understanding. Also explores how different algorithms can solve the same problem but may not be equally good – some may be faster.	• Papers • Coins

Scratch Activities

The fundamental *Coding* concepts to be introduced through these activities are:

- *Algorithms*
- *Sequencing*
- *Repetition*
- *Variables*
- *User Input*
- *Conditionals*

References for the activities

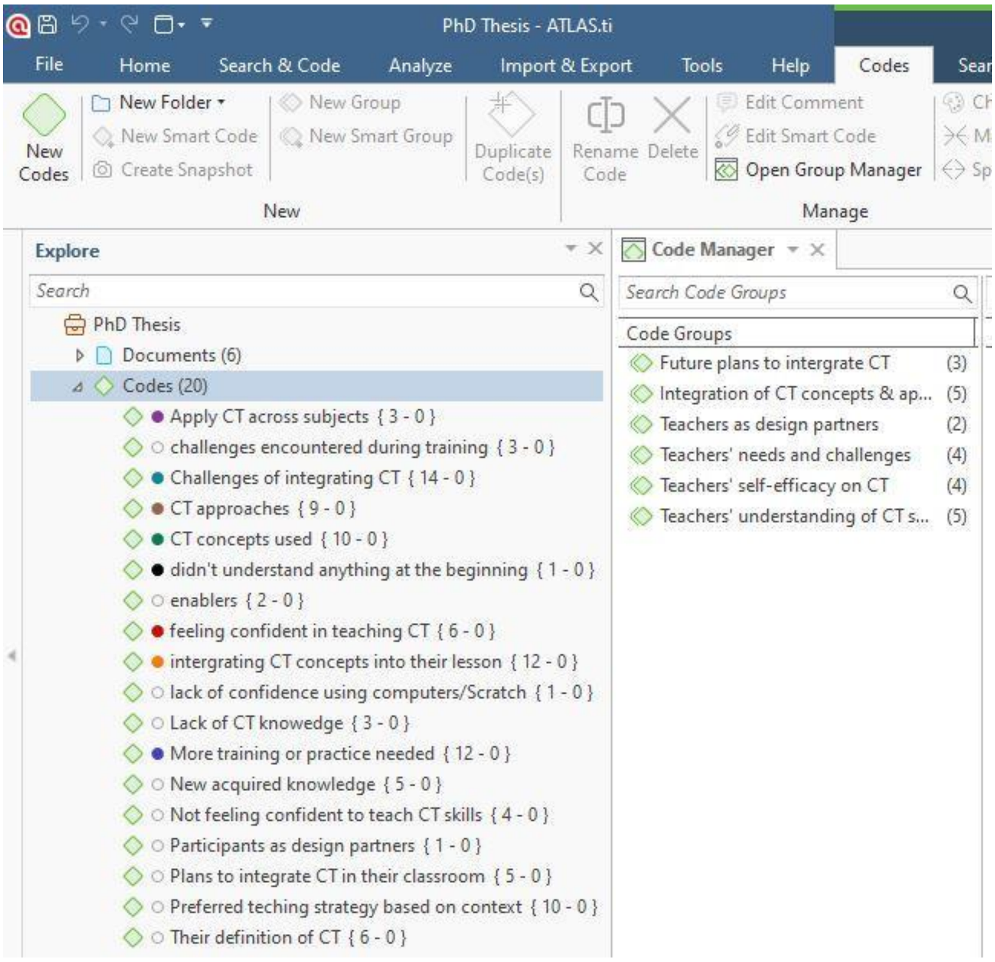
Computing at School (2014). Computing at Schools. Retrieved from http://barefootcas.org.uk

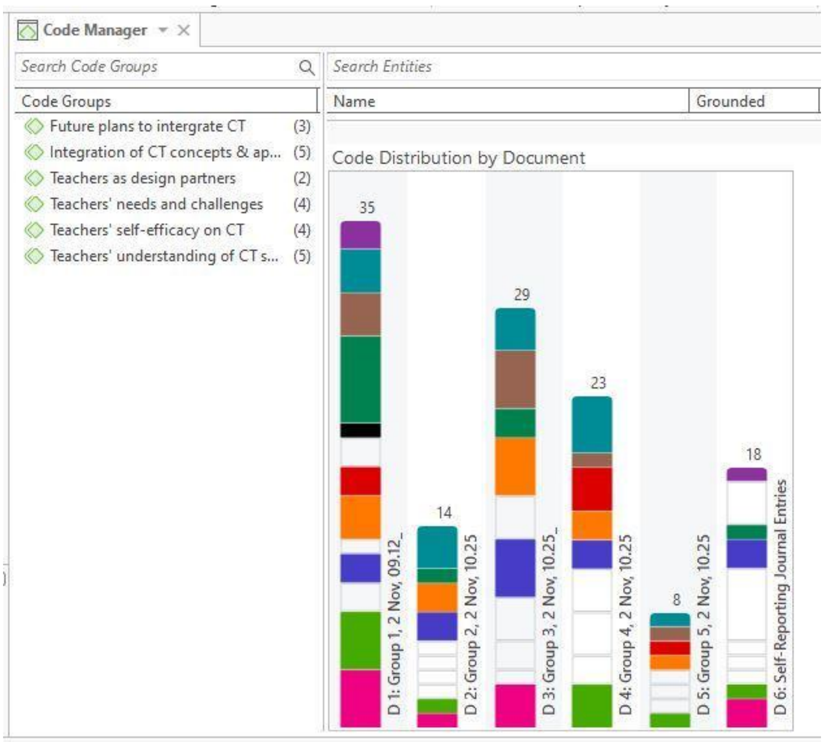
Bebras.org. (n.d.). BEBRAS Challenge. Retrieved October 1, 2021, from https://www.bebras.org/

Code.org. (n.d.). Hour of Code | . Retrieved October 1, 2021, from https://code.org/

CSUnplugged.org. (n.d.). CS Unplugged. Retrieved October 1, 2021, from https://csunplugged.org/en/

Appendix K: Inductive Thematic Analysis coding in Atlas.ti





Appendix L: Groups 'Lesson Plans

Lesson Plan

Grade: 7C-D	**Which key learning topic is this lesson plan for?** Grammatika en Taalgebruik .
Subject: Afrikaans	

Syllabus Outcome(s): *What do students learn and are able to do as a result of this lesson?*

Eenvoudige sinne korrek te skryf. en uitgelate leestekens inkevul

Introduction: *How will you get the students motivated, curious and ready to learn?*

* Onderwyser vra leerders om sinne te skryf.
* Leerders skryf eie sinne, Onderwyser kyk of die leerders die sinne met die benodige leestekens geskryf het.

Metalanguage: *What are the key concepts or procedures that you want students to understand as a result of this lesson?*

* Die gebruik van die leestekens by die regte plek.
* Leerders moet eenvoudige sinne korrek skryf.

Computational Thinking: *Which of the computational concepts, practices and perspectives will students have the opportunity to learn about in the lesson?*

* Debugging – Leerders moet die sinne sonder leestekens uitken en dan die voute regmaak
* Tinkering – ~~To see~~ Leerders moet kan die regte vorm Leesteken
* Creating –

Teaching Activities: *What strategies will you use to teach the content and skills? How long will you spend on each of those strategies and with the content? How would you address different levels or prior knowledge?*

* Onderwyser sit Flitskaarte op die bord, die Leerders moet die leestekens op die flitskaarte kan uitken.
* Onderwyser skryf sinne op die bord dan moet die leerders die leestekens in vul.

Lesson Conclusion: *How will you bring the lesson to a conclusion?*

* Leerders moet gegewende sinne voltooi deur die uitgelate leestekens in te vul.

Assessment: *How will you know whether the students achieved what you wanted them to achieve?*

* Leerders gaan 'n kort toets skryf, waar hulle die uitgelate leestekens moet invul

Resources: *What materials do you need for this lesson? Have you used ideas from elsewhere?*

Flitskaart, skryfbord, skryfboeke, Afrikaans sonder grense woordeboek.

Lesson Plan

Grade: 7	Which key learning topic is this lesson plan for?
Subject: Patterns in numeric	Patterns in numeric

Syllabus Outcome(s): *What do students learn and are able to do as a result of this lesson?*

Know how to find relationships in numeric

Introduction: *How will you get the students motivated, curious and ready to learn?*

- The teacher will ask the learners to say the multiple of two (2) and teacher write them on the Chalkboard.
- Ask learners "how you can discribe the pattern?
- write the lesson's topic on the chalkboard

Metalanguage: *What are the key concepts or procedures that you want students to understand as a result of this lesson?*

- Learners should be able to identify numbers relationship

Computational Thinking: *Which of the computational concepts, practices and perspectives will students have the opportunity to learn about in the lesson?*

- Looping - Learners will identify a number and keeping repeating until the last given number.
- Pattern - Learners will describe the pattern rule/s
- debugging - Perseviaring and collaborating

Teaching Activities: *What strategies will you use to teach the content and skills? How long will you spend on each of those strategies and with the content? How would you address different levels or prior knowledge?*

- Learners centered: whereby the teacher will ask a learner to write two/one examples of numbers pattern on the chalkboard and explains for the learners how to - identify a number pattern
- how to describe the identified number pattern
- Learners working together, Some on the chalkboard.

Lesson Conclusion: *How will you bring the lesson to a conclusion?*

- Ask learners questions to see if they understand
- write 3,6,9.... and ask them to fill in the next three numbers

Assessment: *How will you know whether the students achieved what you wanted them to achieve?*

- learners will be given a table to complete and describe the pattern

Resources: *What materials do you need for this lesson? Have you used ideas from elsewhere?*

- Classwork/home work books
- Poster
- Text book
- Chalkboard

Lesson Plan

Grade: 6	**Which key learning topic is this lesson plan for?**
Subject: NSHE	Ecosystem

Syllabus Outcome(s): *What do students learn and are able to do, as a result of this lesson?*
- learners should be able to explain the terms: herbivores, Carnivores, Omnivores, food chain and food web

Introduction: *How will you get the students motivated, curious and ready to learn?*
- Teacher will ask learners to list different animals (wild, domestic) and what they feed on.

Metalanguage: *What are the key concepts or procedures that you want students to understand as a result of this lesson?*
> Herbivores > Carnivores > Omnivores
> food chain > food web

Computational Thinking: *Which of the computational concepts, practices and perspectives will students have the opportunity to learn about in the lesson?*
> Pattern - Do the similarities and differences
> Algorithms - (food chain and food web)
> Collaboration - learners in the class in groups will do a quick orally activity

Teaching Activities: *What strategies will you use to teach the content and skills? How long will you spend on each of those strategies and with the content? How would you address different levels or prior knowledge?*
> Teacher will explain the meaning of terms using pictures of animals
> Practice using pictures of different animals to match and identify the herbivores, Omnivores, Carnivores
> Teacher will then construct a food chain using the example of. . .

Lesson Conclusion: *How will you bring the lesson to a conclusion?*
> Teacher will instruct the learners, to orally and practically ask the learners to match definitions of the word s written on the flash cards to the terms learned in the class

Assessment: *How will you know whether the students achieved what you wanted them to achieve?* Class work (individual)
- Write the meaning of the terms and give example
e.g Herbivores: cow
Carnivores: Lion
Omnivores: dog | human beings
< Homework
> construct the foodchain

Resources: *What materials do you need for this lesson? Have you used ideas from elsewhere?*
- Posters > textbook
> Flash Cards > Computer

Lesson Plan

Grade: 5

Subject: Social Studies

Which key learning topic is this lesson plan for?
Trade: Export and import

Syllabus Outcome(s): *What do students learn and are able to do as a result of this lesson?*
realise that trade is influence by transport and communication links.

Introduction: *How will you get the students motivated, curious and ready to learn?*
Teacher pose a direct question to learners e.g to imagine their country lacking certain products. What will they do to obtain those certain products?

Metalanguage: *What are the key concepts or procedures that you want students to understand as a result of this lesson?*
How trade is done through import and export

Computational Thinking: *Which of the computational concepts, practices and perspectives will students have the opportunity to learn about in the lesson?*
Decomposition and Algorithm

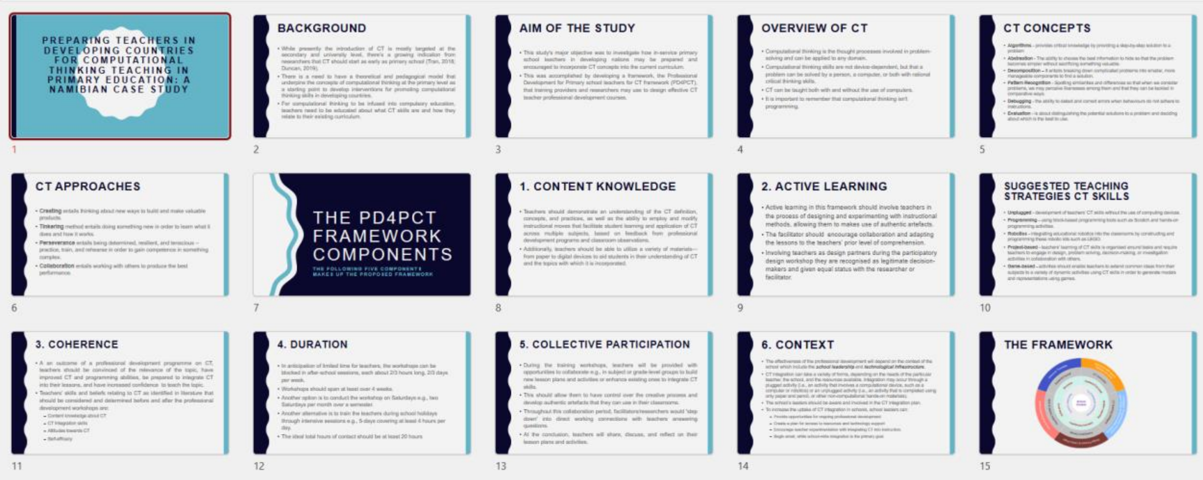
Teaching Activities: *What strategies will you use to teach the content and skills? How long will you spend on each of those strategies and with the content? How would you address different levels or prior knowledge?*
Teacher distribute different pictures of import and export to learners. Teacher ask each learner with a picture to came and place it under the correct heading on the chalkboard, either under import or export. Then the rest of the class say wether its under the correct heading or not.

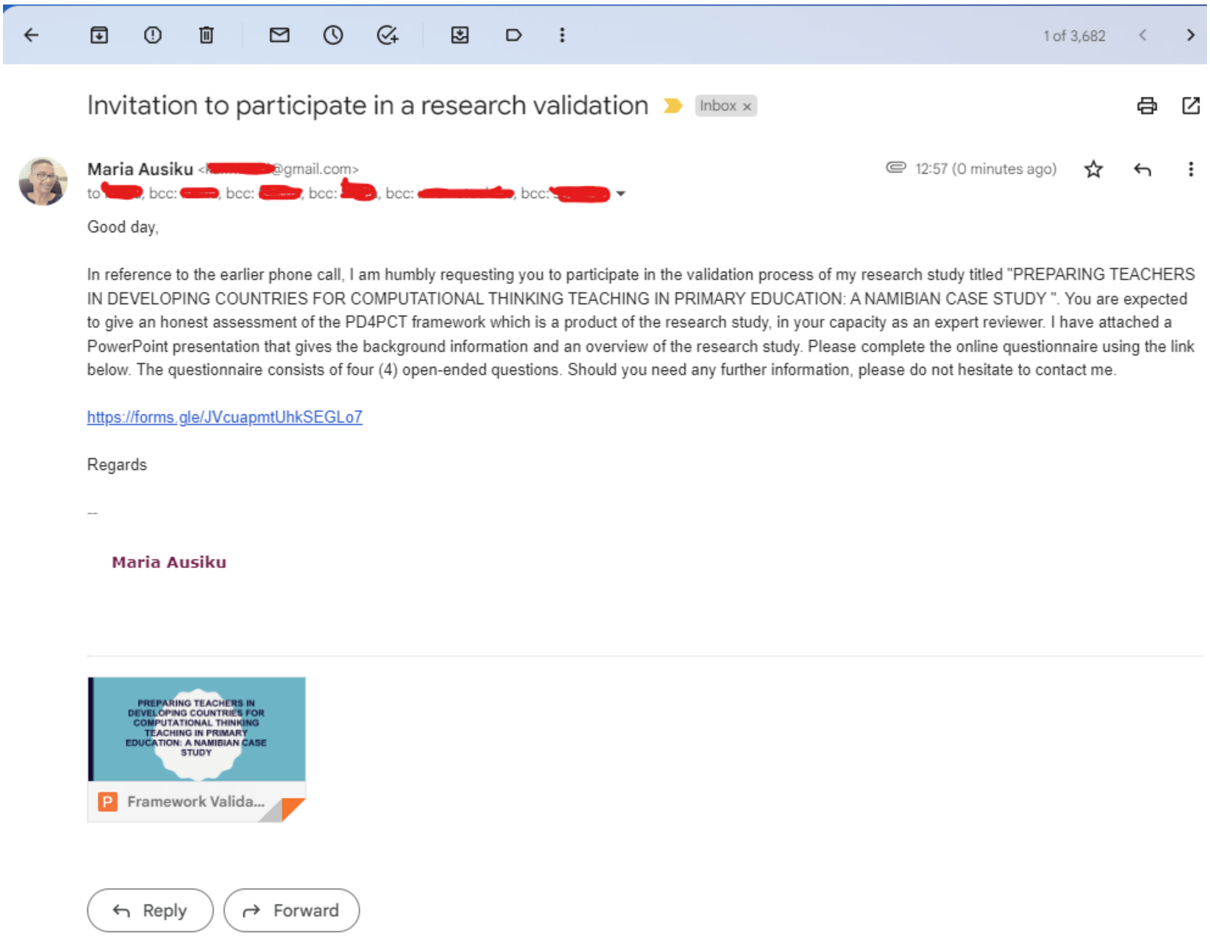
Lesson Conclusion: *How will you bring the lesson to a conclusion?*
Learners are instructed to mention the names of each picture aloud.

Assessment: *How will you know whether the students achieved what you wanted them to achieve?*
Activity: Teacher writes questions based on the topic e.g Explain the terms import and export
Give examples of imports and exports in Namibia

Resources: *What materials do you need for this lesson? Have you used ideas from elsewhere?*
Chalkboard, textbooks, LB, pictures

Appendix M: Validation Presentation & Email





1 of 3,682

Invitation to participate in a research validation Inbox ×

Maria Ausiku <[redacted]@gmail.com>
to [redacted], bcc: [redacted], bcc: [redacted], bcc: [redacted], bcc: [redacted], bcc: [redacted]

12:57 (0 minutes ago)

Good day,

In reference to the earlier phone call, I am humbly requesting you to participate in the validation process of my research study titled "PREPARING TEACHERS IN DEVELOPING COUNTRIES FOR COMPUTATIONAL THINKING TEACHING IN PRIMARY EDUCATION: A NAMIBIAN CASE STUDY ". You are expected to give an honest assessment of the PD4PCT framework which is a product of the research study, in your capacity as an expert reviewer. I have attached a PowerPoint presentation that gives the background information and an overview of the research study. Please complete the online questionnaire using the link below. The questionnaire consists of four (4) open-ended questions. Should you need any further information, please do not hesitate to contact me.

https://forms.gle/JVcuapmtUhkSEGLo7

Regards

--

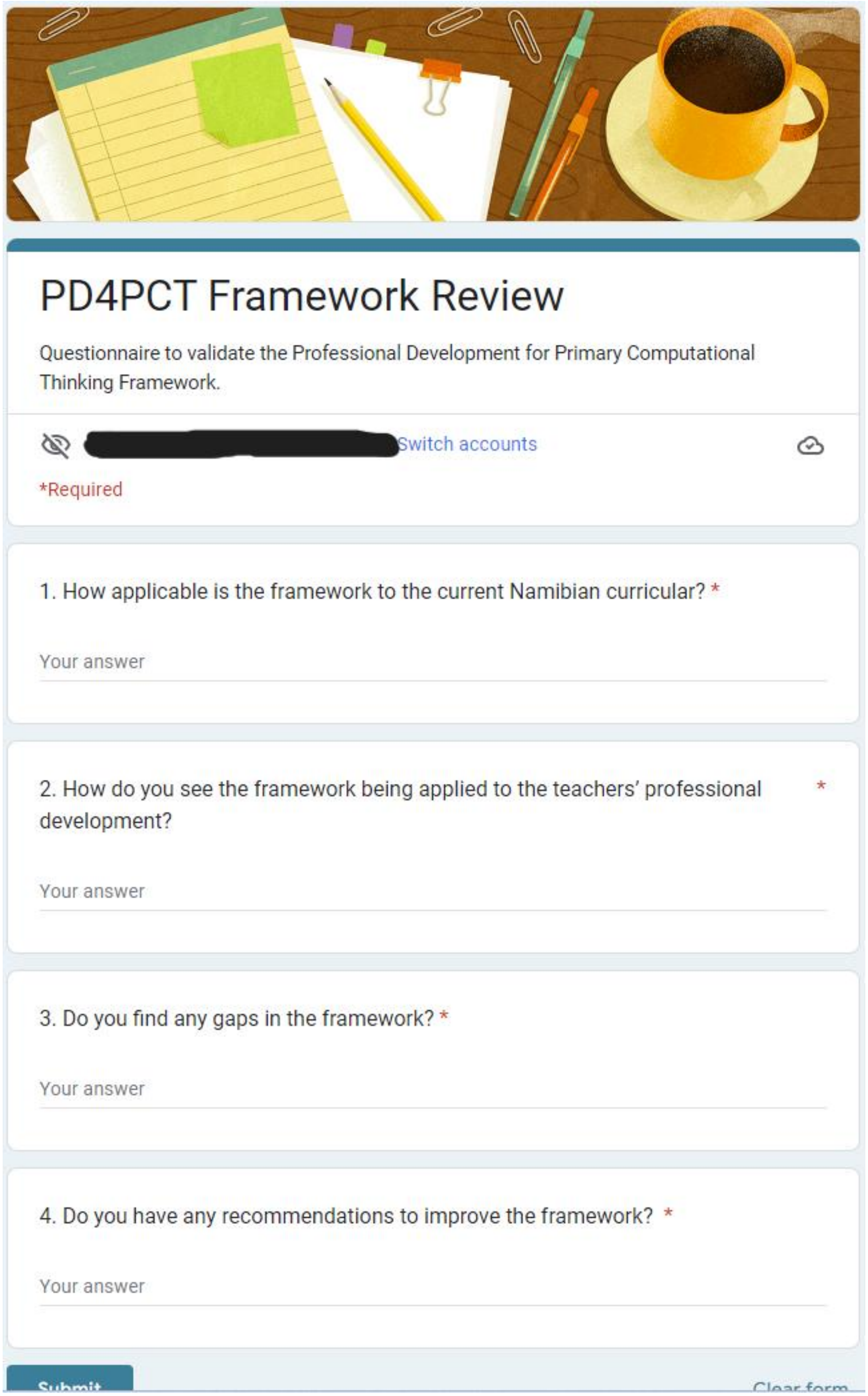
Maria Ausiku

PREPARING TEACHERS IN DEVELOPING COUNTRIES FOR COMPUTATIONAL THINKING TEACHING IN PRIMARY EDUCATION: A NAMIBIAN CASE STUDY

Framework Valida...

Reply Forward

Appendix N: Validation Online questionnaire



PD4PCT Framework Review

Questionnaire to validate the Professional Development for Primary Computational Thinking Framework.

[redacted] Switch accounts

*Required

1. How applicable is the framework to the current Namibian curricular? *

Your answer

2. How do you see the framework being applied to the teachers' professional development? *

Your answer

3. Do you find any gaps in the framework? *

Your answer

4. Do you have any recommendations to improve the framework? *

Your answer

Submit

Clear form

Appendix O: Language Editing Certificate

University Editor (CC)

Certificate of comprehensive English editing

This document certifies that we have edited the manuscript indicated below for English language, grammar, spelling, clarity, and scholarly writing style.

Manuscript Title:

PREPARING TEACHERS IN DEVELOPING COUNTRIES FOR COMPUTATIONAL THINKING TEACHING IN PRIMARY EDUCATION: A NAMIBIAN CASE STUDY

Author:
Maria Ausiku

Date Issued:

13 November 2022

The author's subject matter contents and intentions were unaltered during the editing **process**. Manuscripts with this certification should be grammatically ready for publication; however, the author/s have the final choice to accept or reject our suggestions and changes. If you have any questions regarding the edited document, kindly contact info@uedit.org or visit us at https://uedit.org

Singed by the editor Apostolides in Pretoria

Postal Address: C/O I Lazarus & Co Registered Accountants and Auditors, P O Box 274, Pretoria 0001, South Africa
Office Address: C/O I Lazarus & Co Registered Accountants and Auditors, Upper Level, 4 Greenpark Estates, 27 George Storrar Drive, Groenkloof, Pretoria 0181, South Africa, Email: info@uedit.org https://uedit.org

Chief Executive Officer: A Apostolides BA (UNISA), MPhil, MA (*cum laude*), PhD (University of Pretoria)