EXPLORING THE DEVELOPMENT OF COMPUTATIONAL THINKING AMONG PRE-SERVICE TEACHERS USING VISUAL PROGRAMMING: AN INTERVENTIONIST CASE STUDY

BY

CHIKONDI SEPULA

A thesis submitted in accordance with the requirements for the degree of

MASTER OF EDUCATION

In the subject of

INFORMATION AND COMMUNICATION TECHNOLOGY (ICT) IN EDUCATION

FACULTY OF EDUCATION

AT

RHODES UNIVERSITY

Supervisor: Associate Professor Clement Simuja

December 2024

ABSTRACT

Due to its cognitive effect on learners, computational thinking (CT) has gained significant attention and has been increasingly integrated into primary and secondary education worldwide. The integration of CT into educational curricula offers several benefits, including improved learning outcomes, enhanced problem-solving abilities, and the development of skills necessary for the digital landscape of the 21st century. Recognizing these benefits, South Africa introduced CT in primary schools in 2023 through a dedicated subject, *coding and robotics*. However, teacher upskilling remains a major challenge, as many teachers lack the necessary skills to teach this subject effectively. This problem is particularly pronounced at the foundational phase, where delivering similar content to young learners presents additional pedagogical complexities. Thus, this study explored the development of CT skills with visual programming among foundational phase pre-service teachers. Situated within the interpretive paradigm, a qualitative case study methodology was employed, owing to its effectiveness in exploring contextual factors and complexities that influence human experiences. The study involved 49 first-year pre-service teachers in an Introduction to Technology module at Rhodes University. Purposive sampling was used to select the foundational phase pre-service teachers at Rhodes University. Data was collected using CT reflective tool, semi-structured interviews, focus-group discussions, and reflective journals.

This study was grounded in Vygotsky's sociocultural theory (SCT) to understand and mediate the development of CT through visual programming. It utilized the "Code, Connect, Create" professional development (PD) model as a structured teacher training approach for CT development through visual programming. Additionally, the "Use, Modify, Create" pedagogical model was implemented as a CT framework specific to visual programming to guide and regulate pedagogical decisions during the intervention. Brennan and Resnick's 3D CT framework was employed to identify and analyse the CT concepts and skills incorporated in the study. Thematic analysis, which involved coding was used to generate themes from qualitative data to address the research questions. Results indicated that visual programming approach enhances CT by fostering interest, creativity, and collaboration. Key enablers included contextualised project-based learning, a clear rationale for CT, and prior programming exposure. Conversely, lack of a clear CT rationale, prior programming experience, and the multimedia nature of Scratch were identified as hindrances. The study concluded that visual programming effectively enhances CT among foundational phase pre-service teachers and recommended its early integration into their training programs to enhance CT skills.

Key words: Computational thinking, Visual programming, Sociocultural theory, Problem-solving skills, Scratch, Cognitive development

DECLARATION

I declare that this is my own independent work. It is submitted for the degree of Master of Education in ICT in Education at Rhodes University, Makhanda. It has not been submitted before for any degree or examination at any other university. All the references that I have used have been indicated and acknowledged by means of complete references.

Chikondi Sepula

………………………………..

December 2024

DEDICATION

This thesis is dedicated to my grandmother, Stelia Masamba whose unparalleled dedication, tireless work ethic, and perseverance as a single grandmother have been a constant source of inspiration.

ACKNOWLEDGEMENTS

I am very humbled to express my sincere gratitude to the following:

- My supervisor, Associate Professor Clement Simuja for his insightful guidance, prompt feedback, support, and encouragement throughout this master's degree journey.
- All Rhodes University first-year foundational phase pre-service teachers who willingly participated in this study. Despite knowing that they could refuse to participate and withdraw their participation at any point, they continued to support this study through their authentic contributions and honest participation.
- My wife, Isobell Sepula, for her patience, understanding and unwavering emotional support throughout my study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ACRONYMS

CT: Computational Thinking

CAPS: Curriculum and Assessment Policy Statement

ICT: Information and Communication Technology

DBE: Department of Basic Education

PSTs: Pre-Service Teachers

FP: Foundational Phase

JRC: Joint Research Centre

ISTE: International Society for Technology in Education

CTt: Computational Thinking Test

STEM: Science, Technology, Engineering and Mathematics

PD: Professional Development

TUC: Teachers Union Collaboration

SCT: Sociocultural Theory

ZPD: Zone of Proximal Development

MKO: More Knowledgeable Other

BEd: Bachelor of Education

CES: Curriculum Education Specialist

PIRLS: Progress in International Reading Literacy Study

CHAPTER ONE

INTRODUCTION TO THE STUDY

1.1 Introduction

The term "computational thinking" was first coined by computer science educator Seymour Papert in the 1980s (De Groot, 2018). However, it was not until 2006 that CT began to gain widespread recognition when Jeannette Wing introduced the first structured definition of the concept (Marcelino et al., 2018). Wing defined CT as "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, which could be a human, a computer, or a combination of both" (Wing, 2011 p. 20). It is important to note that CT is not about thinking like a computer but rather about thinking like a computer scientist when solving problems (Wing, 2009). Essentially, CT encompasses a problem-solving skill set drawn from computer science principles. Wing (2009) projected that CT would become a fundamental skill utilized by everyone in the 21^{st} century. She argued that just as reading, writing, and arithmetic are essential skills, every child should learn CT to function effectively in today's global digital economy. Expanding on the conversation, Resnick (2009) described CT as a multifaceted competency that can be acquired through computer programming yet applicable beyond the computer science setting.

The applicability of CT across different disciplines has attracted interest from various educational stakeholders (Stefania et al., 2022). Educational researchers and practitioners have begun advocating for the integration of CT in primary and secondary school education (European Commission. Joint Research Centre, 2016). Literature indicates that CT improves students' performance in reading, math, and science during their primary school years (Code.org et al., 2022; Arfé et al., 2020; Scherer et al., 2019; Code.org, 2018). Similarly, research has demonstrated that students who learn and possess CT skills are more likely to pursue higher education (Bazhouni, 2018; Brown & Brown, 2020; Janssen et al., 2019; Tiruneh et al., 2014).

Moreover, the 2022 State of Computer Science Education Report in the United States concluded that CT plays a crucial role in addressing students' educational and learning needs (Code.org et al., 2022). Therefore, I am of the view that the integration of CT into South Africa's primary school curriculum may be a worthy complementary initiative to help tackle the prevalent problems of reading and learning mathematics among learners. As argued by

Jacob & Warschauer (2018), CT can improve learners' literacy skills by fostering algorithmic and abstract skills that enable learners to understand complex texts and create structured narratives. Similarly, the analytical and abstract skills that learners acquire through CT are crucial for solving numerical problems (Finke et al., 2022). According to the 2021 Progress in International Reading Literacy Study (PIRLS), 80% of grade 4 students in South Africa cannot read for meaning (Mullis et al., 2023). Thus, it is important to employ various measures to address the country's dire challenges in reading and mathematics, particularly among learners in foundation phase grades.

Many developed countries are actively integrating CT into their primary and secondary school curricula, primarily through computer programming to equip learners with essential digital skills and foster problem-solving skills that are foundational for general education (European Commission. Joint Research Centre., 2016). A systematic review study conducted by the European Commission Joint Research Centre (JRC) in 2022, revealed that out of the 29 European countries analysed, 25 had already incorporated computer science concepts into their curriculum to foster CT, while the remaining four countries were in the planning phase of policy development (Stefania et al., 2022). Among these 29 countries, 17 had made CT concepts mandatory in primary and lower secondary schools. However, it is imperative to note that despite the recorded positive strides made in the cited contexts, by far the most significant challenge in integrating CT into primary and secondary education is the upskilling and support of teachers (Stefania et al., 2022). This challenge is particularly pertinent as many teachers lack a background in computer education. Therefore, it becomes essential to introduce interventions that provide teachers with appropriate content and pedagogical knowledge to effectively teach CT.

1.2 Context of the study

In South Africa, CT has been introduced as a subject called coding and robotics following the guidelines outlined in the Curriculum and Assessment Policy Statement (CAPS) for grades R-3. The specific aims of this subject as stated by the Department of Basic Education (2021), are to equip learners with problem-solving skills, critical thinking abilities, proficiency in the digital economy, and the ability to apply and transfer digital and ICT skills to solve everyday problems. The Department of Basic Education (DBE) has announced the rollout of this subject for grades R to 9 in the 2024 academic year, marking its introduction to South African education (Department of Basic Education, 2023). According to the Annual Performance Plan 2022/2023, the DBE plans to collaborate with higher education institutions to upskill teachers in coding and robotics subject (Department of Basic Education, 2022).

However, during the Coding and Robotics Inter-provincial Meeting held in September 2022, where each province presented the status of the subject's pilot programs, it was revealed that teacher training in coding and robotics was a major challenge across all nine provinces (DBE & PED, 2022). The subject was introduced before a comprehensive teacher training program had been implemented, leaving many teachers insufficiently prepared to effectively teach the subject. Furthermore, at the 2022 GovTech Conference, the Deputy Minister for DBE emphasized the need for higher education institutions to produce pre-service teachers who are well trained in integrating CT and other ICT technologies into primary and secondary school curricula (DBE, 2022). This call clearly highlights the necessity for coordinated efforts and interventions to train pre-service teachers to effectively integrate CT in their future teaching classroom.

1.3 Problem statement

The lack of teachers with appropriate background and capabilities has consistently been identified as a significant inhibitor to the successful integration of CT in compulsory education in various countries (Li, 2021; Zhang et al., 2020; Caeli & Yadav, 2020; Grover & Pea, 2013). Stefania et al. (2022) note that a plethora of literature reveals that teaching coding and robotics to enhance CT without a comprehensive training is difficult particularly for teachers without a background in computer education. Researchers have emphasized the need to redesign teacher training curricula to equip educators with CT and pedagogical knowledge necessary for its successful integration (Passey, 2017; Webb et al., 2017).

In the context of developing countries, including South Africa, there is limited literature on CT and teacher training in CT development, in particular. Furthermore, most university pre-service teacher training programs in the country, including Rhodes University, have not yet responded to the newly introduced subject of coding and robotics. The lack of empirical research and evidence on CT in South African educational contexts suggests that the concept is still emerging. Even though conversations about CT are gradually gaining prominence in the primary educational sector, there is little progress in integrating CT into pre-service teacher training programs. Moreover, given the DBE initiatives of increasing focus on CT in primary school education (DBE, 2022), it is important to explore how teacher education programs in South Africa can better equip pre-service teachers with CT skills.

Professional development for teachers and pre-service teacher training in CT are crucial for the successful integration of CT into primary education (Yuan, 2023). Considering the cross-disciplinary nature of CT and the need to transform educational practices in South Africa, it seems advantageous to train pre-service teachers to integrate CT concepts and practices into primary education classrooms. Despite the majority of current CT integration efforts cantering on in-service professional development, I argue that pre-service teacher education provides an ideal opportunity to equip future teachers with the CT understanding they need to integrate it successfully into their teaching. Due to this knowledge and empirical research gap, the study intends to work with first-year Foundational Phase (FP) pre-service teachers in developing computational thinking using visual programming as an approach.

1.4 Purpose and significance of the Study

This study aims to explore the development of CT with FP pre-service teachers using visual programming. It seeks to investigate the enabling and constraining factors involved in this process. The inspiration for this interventionist form of case study research topic stemmed from my participation in the 2023 Makana Schools ICT Day, organized by the Sarah Baartman Department of Education in partnership with Rhodes University. During the event, learners from 13 local primary schools in Makhanda showcased how their schools integrate various ICT technologies into their teaching and learning practices (Grocott's Mail, 2023). Notably, only two schools presented on coding and robotics, despite the coordinator's call for more schools to showcase it during one of our preparatory meetings. The District Curriculum Education Specialist (CES) also expressed the need for collaboration with the Rhodes University team to provide training on coding and robotics to teachers, as they lack experts in this new subject area.

Therefore, this study has the potential to prepare the FP pre-service teachers to effectively teach coding and robotics and infuse CT into their future classrooms. Consequently, these teachers may also serve as mentors to their colleagues, helping them in the teaching of coding and robotics. This peer mentorship could contribute to broader systemic improvement in CT integration at the primary school level. Furthermore, by developing CT skills, pre-service teachers may not only enhance their instructional practices but also apply these problem-solving strategies in their own academic and personal lives, as CT has been proven to be a transferrable skill (Resnick et al., 2009). This broader applicability underscores the long-term value of CT training beyond the immediate classroom context.

Additionally, the results of this study may inform the redesign of the *Introduction to Technology* module within the FP teacher education program at Rhodes University, ensuring that it incorporates CT and aligns with the new primary school curriculum. The redesigned course structure might serve as an exemplar of CT integration in teacher education, providing a model for other institutions seeking to integrate CT into their training programs. Beyond curriculum development, the study's findings could also inform education policy by offering evidence-based recommendations for CT development and integration in pre-service teacher training. As a researcher and computer science educator, the findings of this study may enhance my own skills, knowledge, and practice regarding teacher training in CT. Lastly, the study's outcomes may fill a contextual literature gap and serve as a basis for future research.

1.5 Research Goals and Questions

The main goal of this study is to explore the development of computational thinking with Foundation Phase pre-service teachers using a visual programming approach. The research aims to answer the following exploratory research questions:

1. How do Foundation Phase pre-service teachers develop computational thinking with visual programming?
2. What are the enabling and constraining factors that affect the development of computational thinking using visual programming approach among foundational phase pre-service teachers?

1.6 Overview of the thesis

This section presents the structural overview of the thesis as follows:

Chapter One defines and introduces the main research concepts by presenting background, aim, and significance of the study. It discusses the research problem within both international and local contexts. Additionally, Chapter One outlines research questions and presents an overview of the entire thesis.

Chapter Two reviews the latest and the most relevant literature related to this study. The aim of the study and the research questions guide the literature review process. All literature reviewed in this study is drawn from credible, authentic and authoritative electronic databases such as Google Scholar, Scopus, ERIC, Web of Science and Science Direct. Due to the lack of literature on CT teacher education in South African context, most literature is drawn from international context. The literature reviewed on CT teacher education covers both pre-service and in-service teacher training. In addition, this chapter discusses the theoretical frameworks that underpinned the study. The Vygotsky's (1978) social cultural theory is the main theory that underpins this study. Informed by this theory, two models guide the intervention. The "code-connect-create" professional development model is discussed where only the first phase of the model is implemented in this study. The "use, modify, create" pedagogical model is also presented in this chapter. The Brennan and Resnick's (2012) CT framework is discussed as a theory that determines CT concepts and skills covered in this study. The basis for the adoption of these theories and models is sufficiently explained and justified.

Chapter Three presents the research design and methodology adopted in this study, located within the interpretivist paradigm using a case study method. The rationale behind the adopted research design and methodology is clearly given and justified. The criteria for the sampling and the sample size are also given. Furthermore, the strengths of the data collection tools used in this study are highlighted while acknowledging their inherent shortcomings. In addition, the chapter presents the thematic data analysis process, triangulation, research evaluation and concludes by addressing the ethical considerations of the study.

Chapter Four provides the results of the study. The emerging themes from the collected data are accurately identified and presented based on the research questions. The chapter presents the results as given or quoted by the participants in order to enhance dependability.

Chapter Five outlines a discussion of the findings of the study. The findings are compared and contrasted to prior literature reviewed in Chapter Two. Existing findings that are consistent with the findings of this study are cited. The research questions are comprehensively addressed in this chapter.

Chapter Six This chapter presents the overview of the whole study. Furthermore, it provides a summary of the major research findings from which conclusions are drawn and recommendations made and highlights gaps which would serve as potential future research areas.

1.7 Conclusion

This chapter presented a compelling argument for the need to explore the development of computational thinking among FP pre-service teachers using visual programming. Furthermore, the chapter presented the context of the research, problem statement, aim of the study, research questions and purpose and significance of the study. The focus of the next chapter is a literature review related to the phenomenon under investigation, particularly the trends, debates and contextual gaps.

CHAPTER TWO

LITERATURE REVIEW AND THEORETICAL AND ANALYTICAL FRAMEWORKS

2.1 Introduction

This chapter reviews literature related to the development of computational thinking among pre-service teachers using visual programming. Literature review is defined as a process of actively and carefully analyzing published information related to a particular research topic (Snyder, 2019). The search for relevant and credible literature was conducted using authentic, authoritative and credible electronic databases such as Google Scholar, Scopus, ERIC, Web of Science and Science Direct and other secondary sources such as reports, government gazettes and newspapers. The purpose of literature review in this study was to; provide fundamental knowledge on the research topic (Frank & Hatak, 2014); to identify areas of previous research to prevent duplication (Hammer et al., 2023); position this study within the context of existing literature to ensure use of conventional, effective and acceptable research mechanisms (Webster & Watson, 2024); and to help identify contextual knowledge gap and thus building an argument for why this research is necessary (Nakano & Muniz Jr., 2018). This chapter therefore consists of two main sections, the first section covers literature, and the second section covers the theoretical and analytical frameworks. The literature section has six sub-sections and begins with (a) definition of CT and debates around its definition, (b) visual programming framework, (c) CT integration approaches, (d) assessment of CT, (e) benefits of CT, and (f) integration of CT in teacher education. Section two of this chapter consists of three main sub-sections; the first sub-section discusses sociocultural theory; the second sub-section discusses the professional and pedagogical models, and the last sub-section presents the Brennan and Resnick's CT framework.

2.2 Literature Review

2.2.1 Computational Thinking

The literature on CT reveals a lack of consensus among researchers regarding its definition. However, Wing's (2011) definition has gained prominence and become a reference point for most studies and discussions on CT. Wing defines CT as a cognitive or thought process, thus, focusing on the development of thinking skills rather than the production of computer artifacts (Csizmadia et al., 2015). Although there is a lack of consensus on the precise definition of CT, the literature highlights a set of CT skills that are recursively emerging (European Commission. Joint Research Centre., 2016). Depending on the CT framework used, these skills might bear a different label. For instance, the analytical framework adopted for this study identifies them as CT practices. In the European Commission Joint Research Centre (JRC)'s report of 2016, they are called CT skills and encompass abstraction, pattern recognition, algorithmic thinking, problem decomposition, automation, and debugging (Stefania et al., 2022). Overall, while the definition of CT may vary, there is agreement on its core skills and concepts, all of which contribute to improved learning and problem-solving abilities (Hsu et al., 2018). Through their CT course called *Computational Thinking for Educators*, Google and the International Society for Technology in Education (ISTE) identify Abstraction, Decomposition, Algorithmic thinking, and Pattern Recognition as the four fundamental CT skills that should be integrated in every CT education curriculum (Google for Education, 2022). These foundational skills serve as essential cognitive tools for analysing and solving problems, fostering logical reasoning and developing digital fluency among learners (Lodi, 2020). Each of these skills is discussed in detail below, though they are not presented in a hierarchical order, as they are interrelated and collectively contribute to enhance CT development.

Algorithmic Thinking

According to Dong et al. (2019), algorithmic thinking is a skill that involves developing a precise sequence of step-by-step instructions for solving a problem. To illustrate algorithmic thinking, Riley & Hunt (2014) use an example of a calculator. A calculator operates based on a predefined set of rules or algorithms that it follows consistently for performing arithmetic operations like multiplication or division. Regardless of the specific arithmetic operation entered, the calculator applies the same sequence of steps to compute the solution. The scholars note that this adherence to a set of instructions enables the calculator to handle a wide range of similar arithmetic problems. When learners start writing their own computer programs, they develop algorithmic thinking skills (Romero et al., 2017). They learn to breakdown complex tasks or problems into smaller, manageable steps and create a precise sequence of instructions to guide the computer's behaviour. Consequently, by thinking algorithmically, individuals learn to solve problems systematically and precisely, organizing their thoughts logically and structured. Lodi (2020) argues that algorithmic thinking fosters a mindset of logical reasoning and problem-solving that extends beyond programming and can be applied in various domains to approach and solve a wide range of problems effectively.

Abstraction

Abstraction is a CT skill which refers to the process of simplifying an artifact, a problem or process by focusing on relevant information while disregarding irrelevant details (Dong et al., 2019). Csizmadia et al. (2015) highlight the importance of selecting the appropriate information to hide or ignore in order to simplify a problem without losing essential details. A student timetable is often used as an example to illustrate the construct of abstraction (Çakiroğlu & Çevik, 2022). For example, a timetable condenses an entire week's worth of activities into a concise representation of days and periods. It serves as a tool to help students prepare for and attend the correct classes at the appropriate times. However, as observed by Tolboom et al.(2018), not every single activity or event that occurs throughout the week is included in the timetable. Irrelevant details, such as personal errands or social events, are omitted to focus solely on the relevant information related to class schedules and locations. By employing abstraction in this context, the student's timetable simplifies the representation of their weekly activities, allowing them to navigate their academic schedule quickly and efficiently without being overwhelmed by unnecessary information. This example demonstrates how abstraction streamlines complex information, highlighting the crucial elements while disregarding non-essential details.

Decomposition

Decomposition is the skill of breaking down data, processes, or complex problems into smaller, more manageable components (Dong et al., 2019). This allows for a clearer understanding of the individual parts and facilitates the development, evaluation, or solving of these components separately (Csizmadia et al., 2015). Decomposition simplifies complex problems, enhances understanding of novel phenomena, and makes the design of extensive systems more manageable (Stefania et al., 2022). The task of making breakfast can be used to illustrate decomposition. The process of making breakfast can be decomposed into separate small activities, such as boiling an egg, making pancakes, and preparing tea. As breaking down the process into these individual activities, each task can be understood, planned, and executed separately. This CT skill enables a focused approach to each activity, ensuring that it is carried out effectively and efficiently (Shute et al., 2017).

Pattern Recognition

Pattern recognition is defined as the process of observing and identifying patterns, similarities, and trends in data, processes, or problems as described in the ISTE's CT Education Curriculum

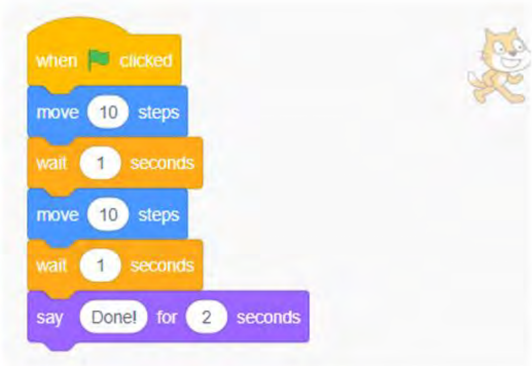
(Google for Education, 2022). It involves use of prior experience and knowledge to solve new problems (Shute et al., 2017). Weintrop & Wilensky (2015) add that pattern recognition entails exploiting identified patterns and similarities to efficiently solve new problems based on previous solutions. Csizmadia et al. (2015) comment that pattern recognition allows computational thinkers to identify similarities between different problems or data sets, enabling them to apply algorithms or procedures previously used to solve specific problems to a broader class of similar problems. When confronted with a new problem, computational thinkers are trained to ask questions such as 'Is this similar to a problem I have already solved?' and 'How is it different?' Through drawing upon pattern recognition skills, individuals can apply their prior problem-solving experiences to identify similarities and adapt previous solutions to the new problem at hand. This skill facilitates the transfer of problem-solving skills across different contexts (Csizmadia et al., 2015).

2.2.2 Visual Programming and Coding in computing

According to Duncan et al. (2014), *programming* is the process of analysing a problem, designing a solution, and implementing it. *Coding* on the other hand, refers to the final stage of implementing the solution by writing computer instructions in a specific programming language (Bell et al., 2017). However, some scholars note that despite the distinction, many stakeholders and this study use the terms programming and coding interchangeably to mean the same thing. Programming serves as a tool for learners to assimilate and concretize CT concepts (Fragapane & Standl, 2021). As learners analyse problems and create software program solutions, they develop essential CT skills such as algorithmic thinking, decomposition, abstraction, and pattern recognition. Wing (2009) highlighted the unique opportunity for educators to use computer science concepts such as programming as a means to foster the learning of CT. Stefania et al. (2022) described programming as a laboratory for teaching and learning basic computer science concepts underpinning CT skills.

While non-programming approaches, also known as unplugged approaches in CT education have gained attention, programming, also called plugged approach remains a practical and established approach for teaching and acquiring CT skills at various educational levels, as acknowledged by several authors (Voogt et al., 2015; Grover & Pea, 2013; Lu & Fletcher, 2009). The literature presents two categories of programming environments or languages: text-based and graphical/block-based/visual programming languages. Text-based languages are used by professional programmers and may not be user-friendly for novices without a computing background. They involve writing instructions using text, numbers, and symbols, which can be too abstract. In contrast, visual programming languages use visual code blocks stacked together to issue specific instructions to a computer. These languages operate at a higher level of abstraction, focusing more on higher-order concepts rather than technical details. Bull et al. (2020) add that visual programming languages eliminate syntactic errors that are common and frustrating for novice programmers. Examples of widely used visual programming languages include Scratch, Logo, Visual Basic, Bubble, and mBlock, all of which are designed to facilitate an accessible and engaging introduction to programming concepts.

The Scratch visual programming language is one of the most widely used tool. Developed by the MIT Media Lab in 2007 under the leadership of Mitchel Resnick, Scratch is a free, open-source, web-based programming platform (Resnick et al., 2009). It allows learners of all ages, even those with limited or no computing background to create games, interactive stories, simulations, and animations while gaining CT skills. Scratch is widely used globally, with a presence in over 200 countries and translated into more than 70 languages (Scratch Foundation, 2023). Hsu et al. (2018) note that Scratch is by far the most used visual programming platform in scientific research. Although Scratch is not explicitly specified in the South African CAPS for coding and robotics, it aligns with the stated requirement of using free, open-source software for block-based coding. This makes Scratch a viable tool for teaching coding in primary schools, as outlined in the CAPS curriculum (DBE, 2021). Figure 1 shows two similar computer programs written in visual and text-based programming environments respectively.




```
input.onButtonPressed (Button.A, function ())
    music.playTone(262,
    music.beat (BeatFraction.Whole))
    basic.pause (1000)
    music.playTone (330,
    music.beat (BeatFraction.whole))
    basic.pause (1000)
    basic.showString ("Done!")
}
```

Figure 1: Similar programs in two programming environments. (left: Scratch(visual), right: Javascript in Micro:bit(text-based))

In the Scratch program shown in Figure 1, the programmer simply drugs and drops six code blocks, which make up what is called a script, to make a digital character move and say the word “Done!”. Meanwhile, a programmer in Javascript in Micro:bit environment (text-based) is required to type nine lines of code while strictly observing the language syntax to implement the same program function. As illustrated in the example from Figure 1, novice programmers using Scratch do not need to know how to write traditional text-based code. Instead, they construct programs by assembling pre-defined drag-and-drop blocks that visually represent programming concepts.

2.2.3 Plugged and Unplugged CT Development Approaches

Literature presents two primary approaches that can be adopted in developing CT: plugged and unplugged. The plugged approach, first to be introduced, involves the use of computers to develop CT skills (Brackmann et al., 2017). A common example of this approach is the use of programming platforms like Scratch. In this study, the plugged approach was adopted, engaging pre-service teachers in visual programming activities to develop CT. As argued by Fagerlund et al. (2020), the plugged approach is the most practical and effectively concretizes CT concepts and skills while equipping learners with essential digital skills.

Despite the practicality and mainstream use of the plugged approach, research has shown that conceptual learning in CT and programming can also take place in non-programming contexts where computers are not necessary (Chen et al., 2023). This approach called unplugged, does not require the use of computers. Del Olmo-Muñoz et al. (2020) contend that using programming or computers is not always mandatory to develop CT. For instance, Curzon et al. (2014) proposed a story-based pedagogy approach to introduce CT concepts to teachers while Bers et al. (201) successfully introduced CT concepts to foundational phase learners through game-based learning with games that foster CT. Brackmann et al. (2017) conducted a quasi-experimental study using unplugged activities to develop CT among primary school learners in Spain. The experimental group was involved in a number of non-programming activities underpinning CT skills such as decomposition, pattern recognition, algorithmic thinking and abstraction. The results showed that the CT unplugged activities enhanced student's CT skills significantly. In a similar study, Tsarava et al. (2017) used playful unplugged activities like a treasure hunt board game to teach the CT concept of variables as information containers.

Although computer programming remains the most direct and mainstream approach for CT development, Chen et al. (2023) observe that the unplugged approach is the most feasible option for schools that do not have computer science trained teachers, technology resources, internet infrastructure or source of electrical power. Additionally, unplugged approaches are gaining popularity in the field of CT due to their user friendliness. Curzon et al. (2014) note that unplugged activities are ideal for novice programmers as they are non-threatening and boost learners' confidence.

2.2.4 Approaches for Integrating Computational Thinking in Education

The European Commission's JRC systematic review on developing CT in compulsory education identifies three main approaches used to introduce CT in schools (Stefania et al., 2022). The first approach involves introducing CT in the form of a separate subject. This is the approach adopted by the DBE in South Africa and many other countries worldwide. In this approach, CT is introduced through subjects such as coding and robotics, programming, or computer science. The rationale behind this approach is that besides equipping learners with problem-solving skills, the subject prepares them for digital careers in the 21st-century. These careers span various fields, including education, business, information technology, and engineering, where computer literacy is increasingly recognized as essential skill for success. As emphasized in the European e-Skills Manifesto, "The world is going digital, and so is the labour market; skills like coding are the new literacy. Whether you want to be an engineer or a designer, a teacher, nurse, or an entrepreneur, you will need digital skills"(McCormack, 2014, p. 57).

The second approach is to integrate CT across the entire curriculum (cross-curricula theme), where CT concepts are addressed in all subjects. This approach aims to support learning and understanding of content and problem-solving across various subject areas. By integrating CT across disciplines, students can apply CT skills in real-life contexts and enhance their overall learning experience (Tiruneh et al., 2018).

The third approach involves integrating CT within specific subject curricula, such as mathematics and Technology. In this approach, CT concepts are integrated into selected subject curricula, not all subjects, to enhance students' problem-solving abilities within those specific subjects. It is important to note that in many cases, a combination of two or three approaches is used to introduce CT in schools (Stefania et al., 2022). The choice of the approach may depend on various factors such as educational policies, curriculum frameworks, and available resources.

2.2.5 Assessment of Computational Thinking

The literature presents several CT assessment tools. According to Román-González et al. (2017), there are five main categories of CT assessment tools developed from different perspectives.

I. **CT summative tools** are the first category of CT assessment tools and they include the aptitude-based Computational Thinking Test, the test for assessing programming concepts and skills (Mühling et al., 2015), and the Commutative Assessment Test (Weintrop & Wilensky, 2015), which assesses learner's understanding of programming concepts with respect to modalities of programming platforms.

II. **CT formative-iterative tools** aim to provide students with real-time automatic feedback, helping them to improve their code and develop CT skills. A popular example in this category is Dr Scratch (Moreno-León et al., 2015), a web based application that allows teachers and students to automatically analyse projects developed in Scratch.

III. **CT skill transfer tools** measure the ability of students to apply CT skills in real-life problem-solving situations. These tools consist of tasks that assess CT skills beyond programming or computer science settings. A good example of this tool is the international Bebras competition (Dagiene & Futschek, 2008).

IV. **CT perception-attitude scales** are tools that measure teachers and students' attitudes towards computational thinking (Rich et al., 2021).

V. **CT Vocabulary Assessment** include tools that measure elements and dimensions of CT expressed verbally by students (Grover, 2011).

To ensure a robust and holistic assessment of CT, the researcher adopted three CT assessment mechanisms in this study; summative, skill transfer, and vocabulary assessments. These mechanisms were implemented through the creation of test items that assessed CT programming concepts, application of CT skills in solving real-life problems, and participants' understanding of CT through verbal expression. The summative assessment items evaluated the pre-service teachers' understanding of CT programming concepts and their ability to apply these concepts in visual programming contexts. The CT skill transfer assessment items were designed to evaluate participants' ability to apply CT skills and concepts in resolving real-world problems thereby demonstrating the transferability of CT. Vocabulary assessment measured the pre-service teachers' ability to define or articulate CT concepts and terminology accurately, reflecting their depth of understanding. By combining the three CT assessment mechanisms, the researcher intended to obtain rich and multifaceted feedback that evaluated the pre-service teachers' conceptual understanding, practical application, and verbal proficiency in CT concepts and skills, providing a comprehensive overview of their CT development. Román-González et al. (2017) conducted a study in Spanish middle schools to assess CT from the perspective of a Computational Thinking test (CTt), a summative assessment. The aim of the investigation was to measure the convergent validity or how closely CTt (summative test) is related to Bebras Tasks, skill-transfer assessment and Dr Scratch, a formative-iterative assessment. The study involved an 8-week coding course in Scratch environment. Pre-Post CT tests were administered and Scratch projects were analysed with Dr. Scratch. Results revealed that the three assessment mechanisms are convergent but cannot be used instead of any other. This implies that the three tools are complementary and should be combined for a balanced and robust CT assessment.

Perez & Valladares (2018) argue that the CT assessment mechanisms should not be implemented exclusively as each mechanism has its unique strengths and weaknesses. For instance, summative assessments are important as they align closely with the CT curriculum, and they are suitable for large-scale screenings as well as early detection of students with high abilities for programming skills (Cutumisu et al., 2019). However, they are often criticised for focusing on CT concepts and ignoring CT skills and attitudes or perspectives (Brennan & Resnick, 2012). Another weakness that is associated with summative assessments is that they are decontextualised, not reflecting the practical application of CT in real-world scenarios (Poulakis & Politis, 2021). Román-González et al. (2017) posit that to address these weaknesses, skill-transfer assessments, such as Bebras Tasks, should be included to offer a contextualised and naturalistic assessment of CT. These assessments not only measure CT skills but also foster them by putting students in realistic problem-solving situations. Despite the cited strengths, skill-transfer assessments have their own weaknesses, one of which is being too tangential to the core CT concepts and skills (Román-González et al., 2017). They may stray from the fundamental concepts and skills, making it difficult to ensure a comprehensive evaluation of students' CT development. Some CT advocate scholars suggest a balanced approach that harmoniously combines various CT assessment mechanisms to provide a holistic and comprehensive evaluation (Poulakis & Politis, 2021).

Molina-Ayuso et al. (2022) conducted an investigation where they examined the degree of CT development in pre-service primary teachers in the context of Scratch. The study was conducted in Spanish primary school pre-service teachers training institutions. It involved control and experiment groups with pre-post tests administered in both samples. The CT tests consisted of computational concepts (summative), CT application (skill-transfer) questions, and attitude scales. Cetin (2016) utilized achievement and practice tests and a computer programming attitude scale to investigate the effect of Scratch-based instruction on pre-service teachers' development of CT. The achievement test consisted of programming concepts and

CT terminology questions, whereas the practice test was formed from Bebras Tasks, and skill-transfer assessment items. The computer programming attitude scale measured the participants' attitude towards computer programming. Yadav et al. (2014) conducted a study at a teacher training institution in the United States of America (USA) to assess the impact of a computational thinking module on pre-service teachers' understanding of CT concepts. To assess the pre-service teachers' understanding of CT, the scholars used a "Computational-Thinking Quiz" consisting of open-ended questions about CT concepts, programming and skill-transfer questions. It was observed that the combination of CT vocabulary, summative, and skill-transfer assessments provided a comprehensive evaluation.

2.2.6 Benefits of Integrating Computational Thinking in Education

The literature presents academic, economic, and professional benefits of integrating CT in education. There are three main benefits that are mostly cited by governments as a basis for introducing CT in their education systems: 1) improving students' academic performance, 2) facilitating acquisition of important problem-solving skills applicable beyond computer science, and 3) preparing students for the 21st century careers.

I. Improvement in academic performance

Computational thinking and students' academic achievement have been proven to have a positive correlation (Arfé et al., 2020; Code.org, 2018; Code.org et al., 2022; Hunsaker, 2020; Scherer et al., 2019). The studies demonstrate that the development of CT skills foster problem-solving skills, logical reasoning, and critical thinking, all of which contribute to improved performance across various academic disciplines. Moreover, students gain content-related problem-solving skills that help them easily understand the subject matter (Yadav, Good, et al., 2017). Additionally, scholars have discovered that integration of CT into a specific discipline increases students' interest and engagement in the subject, which may also translate into improved performance (Rich et al., 2017). For instance, scholars from the National Taichung University of Science and Technology in Taiwan, conducted research to investigate how using Scratch affects students' development of CT, their enjoyment of learning and their academic improvement (Ou Yang et al., 2023). The results showed that the use of Scratch lead to increased students' enjoyment of learning and improved academic achievement.

II. Acquisition of important problem-solving skills

Douglas (2022) posits that developing CT equips learners with important skills for creating solutions across various professional domains in future. The transferability nature of CT allows learners to apply problem-solving skills in solving complex problems that are beyond computer science context (Mohaghegh & McCauley, 2016). A systematic review done by Malaysian Computer Education scholars revealed that CT skills enable learners to systematically solve complex problems across a broad range of domains (Zaharin et al., 2018).

III. Preparation for future careers

Hunsaker (2020) noted that computer technology is becoming more embedded in the fabric of every industry, as such, professionals in every domain need computer literacy to leverage the technology to solve future problems. Jones & Flannigan (2006) argued that every learner needs digital skills to be a strong candidate for the 21st century jobs. According to the United States of America Bureau of Labor Statistics report, each year there are far more computing jobs added than there are computer literacy graduates, with significant job growth projected for the future (U.S. Bureau of Labor Statistics, 2018). This suggests that Computational thinking equips learners with digital skills that are in high demand and help them to navigate and thrive in today's technology-driven industry. It is argued that as computer technology is becoming woven into every aspect of our lives, it is imperative that young people develop CT skills so that they can play a meaningful role in their communities and prepare for the changing nature of work (Angevine et al., 2017).

2.2.7 Integration of Computational Thinking in Teacher Education

The available literature on teacher education in CT in South Africa and developing countries is limited. As such, most of the literature in this section is drawn from international studies conducted in developed countries, particularly the USA, Europe, and Australia. Research on CT education has explored training in both in-service and pre-service programs.

Regarding in-service programs, the trainings typically involve weeklong professional development workshops, where non-programming or unplugged approaches are employed. For instance, Blum & Cortina (2007) conducted a study in the United States that investigated the impact of a weeklong workshop on computer science in-service teachers' perceptions of CT. The participants engaged in various non-programming activities that involved applying computer science concepts to solve real-life problems. The findings showed a shift in teachers' perceptions, as they begun to view computer science as a subject relevant to everybody. In a similar study, Prieto-Rodriguez & Berretta (2014) conducted a study in Australia that involved a 3-day workshop aimed at exploring the perceptions of digital technology teachers regarding computer science and CT. The workshop provided an overview of computer science and CT disciplines, showcasing their applications without the use of programming. The results indicated a positive change in teachers' perceptions.

Knie et al. (2022) designed a blended in-service teacher training program on CT to foster interdisciplinary competencies in the field of Science, Technology, Engineering and Mathematics (STEM) in Germany. The blended program alternated between the online and face-to-face phases. During the online phase, participants were introduced to CT concepts and practices such as algorithmic thinking, decomposition, pattern recognition, and abstraction and their real-life applications using visual programming. Additionally, flowcharts were used as a technique for visualizing algorithms. The face-face phase involved hands-on experiments on energy and environment topics using the introduced CT concepts and practices. To evaluate the intervention, a similar questionnaire was used before and after the intervention. Results of the study indicated a significant improvement in teachers' understanding of CT and use of CT-related practices in their classrooms.

Teachers felt that the CT concepts and practices are relevant and have higher practicability in science classrooms. Çakır et al. (2021) investigated technology acceptance, self-improvement in technology use in education, and computational thinking skills of teachers who participated in basic robotic coding in-service training in Amasya, Turkey. Two research scales were used: self-improvement in technology use in education scale and computational thinking skills scale. Results showed higher levels of teacher self-improvement in technology use and technology acceptance. Additionally, there was a statistically significant difference between attending in-service training and not attending the training in teachers' computational thinking skills. Ketelhut et al. (2020) conducted an investigation on how to best support in-service elementary teachers in learning to integrate CT into their science teaching through a CT training program in USA. The CT professional development (PD) program consisted of two parts: a professional development workshop and a science teacher inquiry group.

The aim of the study was to understand how teachers' views on integrating CT in their teaching changed due to their participation in the yearlong professional development program. During PD workshops, teachers were involved in various unplugged activities that foster CT skills and looked at how they can be incorporated in their classrooms. The data collected from the study showed that teachers were excited to incorporate CT into their classrooms. The study findings also indicated that teachers found CT engaging to students. Teachers also noted that CT-infused activities had the potential to be motivating, interesting and relatable to students as they seem to deviate from traditional way of teaching science. There was also an observation that CT has the potential to spark creativity in solving problems. However, a survey done by Cabrera et al. (2018) examined the impact of professional development workshops on the number of qualified CT teachers in USA. The findings revealed that discrete non-programming workshop experiences were not sufficient to meet the demand for qualified CT teachers. Consequently, the inclusion of a CT course with a programming-oriented approach in teacher education programs was recommended to ensure a steady supply of well-qualified CT educators.

According to the South African Department of Basic Education website, by 2022, the Teachers Union Collaboration (TUC) had provided coding and robotics trainings to Grade R to 9 teachers through a 15-week online workshop. The workshops focused on unplugged coding activities that do not require the use of a computer (Greyling, 2023). The four computational thinking skills of decomposition, pattern recognition, algorithmic thinking, and abstraction were covered in the planned CT activities.

In contrast to in-service teacher training, there are limited studies available on CT education for pre-service teachers, as noted by Jaipal (2017). However, most pre-service teacher training programs adopt a programming-oriented approach. For example, Molina-Ayuso et al. (2022) conducted a study that examined the development of CT skills in pre-service primary school teachers. The study involved programming activities using Scratch, with 149 Spanish pre-service teachers participating as part of their mathematics education program. A CT test consisting of 30 questions was administered and the results showed a statistically significant enhancement in test scores and a shift in teachers' attitudes toward CT. A similar study by Cetin (2016) investigated the impact of visual programming instruction on pre-service teachers' understanding and attitudes toward CT. The participants engaged in programming exercises using the Scratch platform. Data collection involved achievement tests, attitude surveys and interviews, which revealed that the intervention improved teachers' CT skills and dispositions. Umutlu (2022) conducted an exploratory study to investigate pre-service teachers' CT and programming skills in an educational technology course that introduced visual programming using Scratch.

The pre-service teachers were involved in programming challenges and projects. Based on the findings, it was concluded that computing education for pre-service teachers is necessary to prepare teachers for 21st-century classrooms. Consequently, the addition of more programming courses to teacher training programs was recommended. Using a design-based research methodology, M Esteve-Mon et al. (2019) designed an intervention through educational robotics for the development of CT of pre-service teachers within a technology course at a Spanish university. The intervention consisted of four stages that involved plugged and unplugged activities. The first two set of activities were called “unplugged” and “playing” involving unplugged initial activities with puzzles, crayons, and printable materials. In addition, pre-service teachers had to design their own unplugged activities using the provided materials. These activities were adopted to introduce the pre-service teachers to CT concepts and skills and their real-life applications. In the last two sets of activities called “making” and “remixing”, the participants were involved in programming activities with Scratch and mBot robots in which they could program their own robots (mBots) to do something like following a black line drawn on a ground and overcome obstacles above the line.

All activities were performed in group settings except for a few individual programming tasks. Self-perception of digital competency and computational thinking tests were administered before and after to evaluate the intervention. Questions in the CT test covered seven CT programming concepts as given by Román-González et al. (2017). The results confirmed the effectiveness of the intervention through educational robotics in the development of CT of the pre-service teachers. In a similar study, Tankiz & Atman Uslu (2023), examined the pre-service teachers’ computational thinking skills and self-efficacy towards teaching CT after attending instructional technologies courses in Turkey. The implementation of the study was conducted in three stages. In the first stage, pre-service teachers were introduced to CT components, their applications and Scratch. In the second stage, the participants developed Scratch programs like games related to their subject areas. In the last stage, they developed lesson plans teaching CT using the Scratch programs they developed in the second stage. Pre and post CT tests were administered to evaluate pre-service teachers’ CT skills and their perception towards its teaching. Findings indicated a significant increase in the participants’ CT skills and self-efficacy perceptions towards the teaching of CT.

The existing literature offers an array of approaches that have proven effective in developing CT skills for pre-service teachers. Findings indicate the effectiveness of the visual programming approach in CT development in pre-service teacher training programs and Scratch offers the platform of choice for many educators in CT development. Additionally, CT assessment tools such as computational thinking tests are widely used. The literature also highlights the growing importance of CT education and the pressing need for research and interventions in pre-service teacher training, especially in the context of developing countries like South Africa. However, most of the studies focus on investigating the effectiveness of an approach in CT development in pre-service teacher training programs. Limited research has been done in exploring how the CT development process takes place through visual programming approach in South African teacher education programmes. Furthermore, it is apparent that not much has been researched on the facilitating and constraining factors in CT development process, particularly in the context of a developing country. Several critical questions related to the integration of CT in teacher education in South Africa remain unanswered. The lack of empirical studies addressing such important issues is what prompted this research study.

2.3 Theoretical and Analytical Framework

2.3.1 Introduction

Theory is a critical component of a research process and Osanloo & Grant (2016) defined a theoretical framework as a blueprint that provides a foundation for a study, guiding it and defining the philosophical, epistemological, and methodological approaches. It provides a structure for understanding a phenomenon under investigation (Yuksel & Yildirim, 2015). Grant & Osanloo (2014) add that a theoretical framework helps researchers focus their inquiries and develop a clear direction for their research. On the other hand, an analytical framework is an approach used to systematically evaluate a phenomenon (Rodrigues, 2010). It serves as a theoretical lens through which data is collected and interpreted. In this particular study, the theoretical framework is built upon the social cultural theory by Vygotsky (1978). This theory serves as the foundation and underpins the study. In addition, Jocius et al. (2020) and Lee et al. (2011) have developed relevant Professional Development (PD) and pedagogical models, respectively, which are informed by this SCT perspective. These models are adopted and integrated within the theoretical framework. Father, Brennan & Resnick (2012) proposed a three dimensional (3D) CT model, which provides a structured approach to analysing and evaluating CT concepts and skills in the context of a visual programming platform. This model complements SCT as theoretical framework used in this study, providing a systematic way to assess the phenomenon under investigation.

2.3.2 Theoretical Framework: Social Cultural Theory

Sociocultural theory is based on the work of Soviet Psychologist Lev Vygotsky, who perceived cognitive development of a child as predominantly a product of peers, caregivers, parents and the culture (Vygotsky, 1980). Vygotsky's theory identifies the critical role that social interactions and culture play in the process of knowledge construction. Even though Vygotsky's ideas, produced between 1924 and 1934, were intended to understand higher psychological processes in children, they have many implications and applications for learners of all ages (Polly et al., 2017).The SCT theory stipulates that the human development process is socially mediated (Vygotsky, 1980). Essentially, asserting that children's cognitive development depends on their interactions with others and the environment that society provides to help them make-sense of the world (Akpan et al., 2020). Vygotsky stressed that meaningful learning takes place through social interactions with individuals, materials, or cultural artifacts.

In this study, the sociocultural theory was employed as a theoretical lens to understand and mediate the development of CT through social interactions and the use of *Scratch,* a tool within the pre-service teachers' social environment. Within this theory, two concepts are used: *mediation of learning*, and *Zone of Proximal Development* (ZPD).

2.3.2.1 Mediation of learning

Mediation of learning is one of the fundamental concepts in Vygotsky's SCT theory. Vygotsky (1980) posited that the process of learning or cognitive development is mediated by the use of 'tools' that are within the individual's cultural context. He argued that psychological tools facilitate the construction of knowledge and mediate individual functioning. The tools could be humans or physical objects (artifacts), which can aid a learner to make-sense of the phenomenon.

According to Vygotsky (1978), learning takes place through social interactions with peers and more knowledgeable others (MKO), which can be parents or teachers. He contended that cognitive development originates from the social context of a child. In his own words, he explained, "Every function in the child's cultural development appears twice: first on the social level, and later on the individual level; first between people and then inside the child" (Vygotsky, 1978, p. 57). In other words, individuals acquire knowledge at the social level first through engaging with people in social contexts like collaborative environments. Later, this knowledge is internalised at the individual level. It is through working with others (peers, parents, teachers) with different levels of skills or knowledge on a variety of tasks that the learner adopts socially shared experiences and acquires useful knowledge (Scott & Palincsar, 2013). In this study, the interactions of the pre-service teachers with each other and the researcher were intended to mediate the development of CT. During the CT sessions where pre-service teachers were engaged in CT plugged (programming) and unplugged activities, the researcher intentionally organised the participants to work in groups of three to four people to facilitate knowledge sharing. Additionally, the programming tool that was used in this study, Scratch, provided a platform where the participants were able to collaborate remotely with other users or access sample programmes from across the global on various visual programming projects. This allowed participants to interact and learn CT programming concepts from peers who are more knowledgeable and skilled in Scratch programming. Moreover, the researcher (Knowledgeable Other) was available throughout the activities to provide help to the participants whenever needed. This sociocultural perspective on learning served as a theoretical lens to demonstrate the effectiveness of mediation of CT development through visual programming, given the pre-service teachers' low levels of familiarity with the Scratch technology.

As advocated by Vygotsky, the other aspect of mediation involves the use of physical objects or artefacts. These material objects and artifacts include cultural practices, language, symbols, and objects such as technology (Vygotsky, 1980). The concept of using physical objects or artifacts as tools for the mediation of learning emanated from Hegel's philosophy that tools served humans to achieve their purposes (Hegel, 2002). Hegel perceived the tool of technology as an extension of human capabilities and realization of human potential. He posits that technology helps humans to transform nature. Building on this concept, Vygotsky encouraged the use of artefacts to transform individual's cognitive development. The theory emphasizes the use of artifacts that are within the cultural context of the learner as individuals make-sense of the world through the lenses of their cultural elements (Vygotsky, 1978). In this digital era, technology has become an integral part of today's culture. And recognizing that the participants in this study are "digital natives," the researcher used a digital tool called "Scratch" to mediate the development of CT among the first-year FP pre-service teachers. Participants used Scratch to create visual programs thereby learning CT concepts and developing essential CT skills. The use of Scratch in this study leveraged the cultural relevance of technology in today's society and aligns with Vygotsky's emphasis on mediated learning through culturally significant artifacts.

2.3.2.2 Zone of proximal development

The ZPD is an essential concept in the sociocultural theory. The zone of proximal development is defined as the distance between the actual development level of a learner and the level of potential development under the guidance of a more knowledgeable other (Borthick et al., 2003). This notion stipulates that a learner needs guidance from a more knowledgeable other, who can be a teacher, parent, or an expert in the area, to achieve more. Other theorists have validated the notion of ZPD. For example, according to Main (2023), studies have shown that learner's mental processes and exam anxiety are significantly influenced by the availability or absence of someone to offer help when needed. Shabani et al. (2010) contend that cognitive development within the ZPD can be facilitated not only by human mediation tools (MKO) but also by various tools such as technology and pedagogical approaches. These non-human tools can play a crucial role in activating learning and helping learners achieve their highest potential within the ZPD (See Figure 2). Figure 2 is a visual representation of the ZPD.

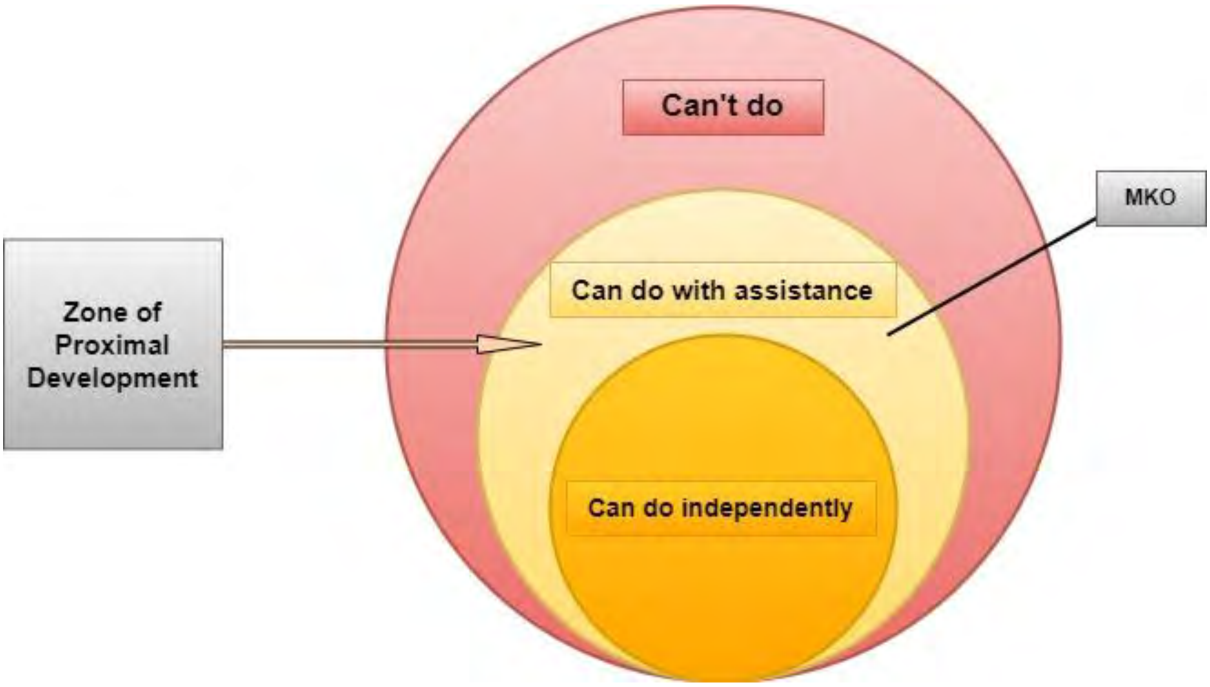


Figure 2: Zone of Proximal Development

In this study, the professional development model (see Section 2.2.3.1) and pedagogical model (see Section 2.2.3.2) were adopted to activate learning in the ZPD, in a teacher professional development setting, focusing on CT. Furthermore, to ascertain the learning process within the ZPD, an adapted formative assessment (see appendix A1) was employed. This assessment aimed to validate whether learning or cognitive development had indeed taken place within the ZPD. Vygotsky's SCT theory emphasizes the pivotal role that a MKO plays in scaffolding or helping an individual achieve their full potential. In light of this, the study hinges on the view that technology acted as a MKO, as the use of technology was able to scaffold PSTs through learning. This notion allowed this study to explore how to use visual programming technology to develop CT during social interactions in the form of CT sessions. The researcher (MKO) demonstrated and guided the FP pre-service teachers on how visual programming could be used to mediate the development of CT. This helped the pre-service teachers who were less knowledgeable in this area to reach or get into their highest potential level of CT development. Therefore, after the intervention, it was revealed that most pre-service teachers understood CT concepts and gained CT skills more than before the intervention (Refer to Chapter 4 and 5). The theory also provided insights about the role of technology, peers and more knowledgeable others in cognitive development.

Like any other theory, sociocultural theory is subject to criticism. Critics have highlighted the lack of specificity as one of the major weaknesses of Vygotsky's theory (Ameri, 2020). They argue that SCT is vague and lacks clear definitions and operationalization of its key concepts, making it difficult to apply the theory in research and practice. Furthermore, Vygotsky's SCT theory is seen as inconsistent with Piaget's cognitive development theory (Matusov & Hayes, 2000). Although Piaget emphasised the individual's active role in the process of knowledge construction through interaction with the environment (DeVries, 1997), Vygotsky emphasised social and cultural factors, assigning the individual a more passive role. Constructivists promote instruction approaches that are learner-centred especially in the context of skill-based content as individuals internalise practical skills effectively through these methods (Amineh & Asl, 2015).

To mitigate these weaknesses, this study and the researcher adopted a professional development model specific to CT training for pre-service teachers, with three well defined stages of CT professional development process (see Section 2.2.3.1). In addition, a specific pedagogical model tailored for CT training with visual programming, was adopted. This pedagogical approach is grounded in both constructivist and sociocultural perspectives (see Section 2.2.3.2), balancing reliance on social environment with individual's active role in the process of knowledge construction. Through incorporating these models, the study aimed to complement sociocultural theory, highlighting its strengths while addressing its weaknesses.

2.3.3 Professional Development and Pedagogical Model

2.3.3.1 "Code, Connect, and Create" (3C) Model

The "Code, Connect and Create" (3C) framework, proposed by Jocius et al. (2020), is a professional development model aimed at supporting content area teachers in integrating CT into their classrooms. This framework is designed to capacitate teachers to incorporate CT both as a separate subject, such as coding and robotics and across various subject areas. The 3C model comprises three key stages: code, connect and create.

The *Code* stage is the initial phase of the model, where teachers are introduced to a visual programming platform to learn coding. The primary goal of this stage is for teachers to develop programming skills and, more importantly, to concretize CT concepts and skills. The model emphasizes the use of visual programming tools like Snap! or Scratch, which shift the focus from programming syntax to higher-order CT concepts and skills. In line with the sociocultural theory, pedagogies that promote social interactions are believed to be effective facilitators of learning. Therefore, during the programming sessions ("code" stage), participants (pre-service teachers) collaborated with each other and the researcher to familiarize themselves with the programming tool and gain hands-on programming experience from the perspective of a student through pedagogies such as pair programming and project-based learning(PBL) (Jocius et al., 2020).

The pre-service teachers worked on the CT programming tasks in groups of three to four individuals for collaboration. Additionally, participants were introduced to the fundamental CT skills such as algorithmic thinking, decomposition, pattern recognition and abstraction and CT concepts such as sequences, loops, conditionals and more as outlined in Brennan & Resnick (2012)'s CT framework. Scratch platform, which was employed in this stage, allowed PSTs to gain visual programming skills by creating visual programs like animations, videogames, music and simulations. This stage is crucial as it establishes the foundation for developing CT skills. To effectively implement this pivotal stage, the study incorporated the pedagogical model coined as "Use, Modify, and Create", proposed by Lee et al.'s (2011). For a visual representation of the design of this model and its alignment with the pedagogical approach, refer to Figure 3.

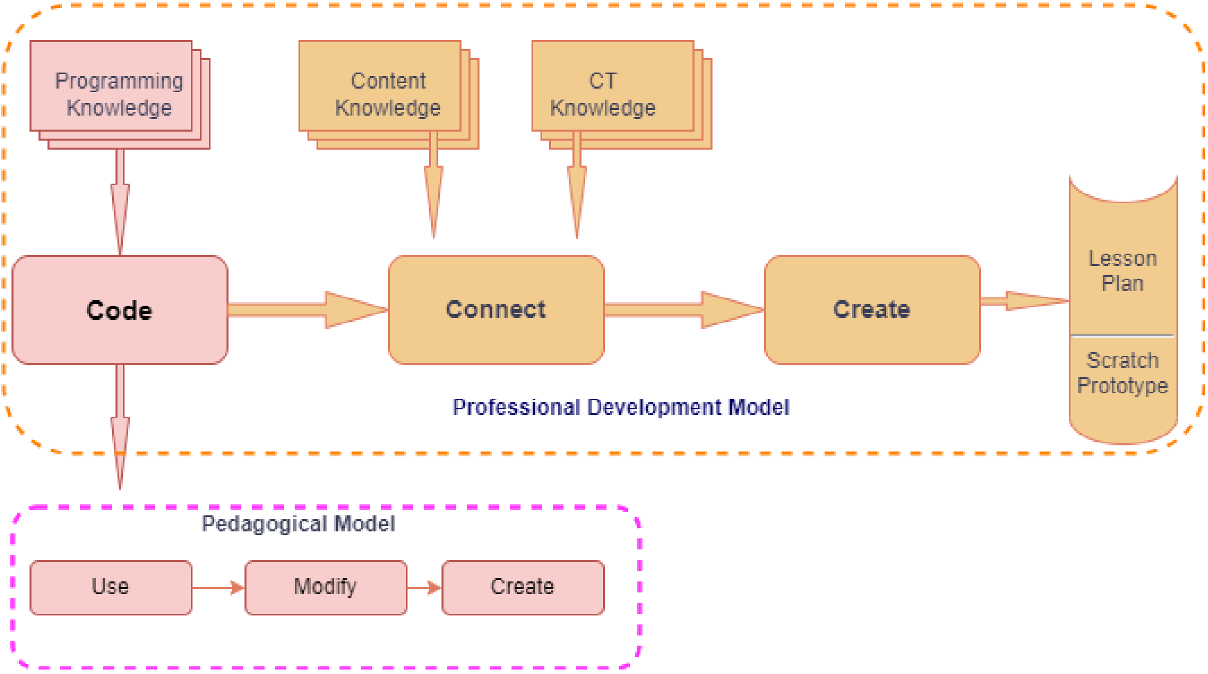


Figure 3: Professional and Pedagogical Models Design

The *Connect* component of the PD model serves to enable participants to identify the CT concepts and skills discussed in the coding stage and explore their application in different disciplines or subject areas. This stage supports teachers in recognising opportunities for integrating CT across various subjects and demonstrating the connection between CT and different disciplines. Participants engage in collaborative activities such as brainstorming and explaining how CT can be integrated into the teaching of different subject areas.

The *Create* stage represents the final component of the model, where participants work individually or in groups to develop their own learning segments. These learning segments can take the form of coding prototypes or non-programming lessons that infuse CT. This stage encourages participants to apply their learning from previous stages and actively create educational materials that incorporate CT into their teaching practice.

2.3.3.2 "Use, Modify, and Create" Model

Lee et al. (2011) presented a pedagogical framework called "Use, Modify, Create" model, designed to support novice programmers in developing CT through coding. This framework has three progression phases, representing participants' cognitive and practical development in CT. This model has been proven to be effective in supporting and enhancing the acquisition of coding and CT skills among programming beginners (Stefania et al., 2022).

In the initial *use* stage, students consume or engage with programming artifacts or code created by others. For instance, they can run a pre-made animation program. The objective of this stage is to help coding novices understand how the visual programming platform functions while building confidence and reducing anxiety. Once students have gained an understanding of how programs are created, they move on to *modify* stage. Here, they begin making changes to existing programs, with increasing levels of complexity. For example, a student may change the colour of a character's shirt in an animation program. Later, they might decide to modify the program's code to change the character's behaviour, potentially by adding new code.

As students gain more skills and confidence through program modifications, they are encouraged to enter the *create* stage, where they generate and implement their own project ideas. In the final stage, it was assumed in this study that participants may apply the key CT concepts and skills they had acquired to design and implement projects of their liking. However, the model emphasizes the importance of maintaining an appropriate level of challenge that supports growth while limiting anxiety during the progression. As such, it is important to acknowledge that the movement from one stage to another is not always strictly linear. Throughout all three stages, a collaborative learning approach is promoted. Moreover, the "use, modify, create" framework was implemented in the programming sessions, as it is well-suited for the PSTs who had no prior programming experience.

In this study, the PSTs first explored various examples of visual programs such as games, animations, stories and music created by different Scratch users across the global and are available online. These programs are accessible to all users under the "explore" tab in Scratch. For example, participants opened a Pong game program created by a user with the username "aiysha88". They played the game, which involved moving a bar to intercept a ball, scoring points upon contact. Beyond playing the game, the PSTs were able to access and examine the script or code of the game by selecting "inside" button, allowing them to understand the programming concepts behind it. The researcher provided explanations of the code and concepts used. This stage enabled participants to observe how CT concepts are used in real programming scenarios. Following this stage, participants were asked to make small changes to the program and observe the effects of their changes. They first adjusted the speed of the moving ball to a slower pace and then changed the score recorded when the ball hit the bar. Finally, they added sound effects that the program made when the ball and the bar got in contact. After these two stages, the PSTs were tasked with creating their own similar game programs, applying the same CT concepts and skills they had learned.

Following this three-stage pedagogical model was crucial for several reasons. First, it allowed PSTs to build a solid foundation by familiarizing themselves with the created programs and understanding the underlying concepts before attempting modifications. This approach ensured that they comprehended the basics thoroughly before moving on to more complex tasks. By examining and modifying existing programs, participants gained hands-on experience with real-world applications of CT concepts, reinforcing their learning and boosting their confidence. Additionally, this gradual progression from exploration ("use") to modification ("modify") to creation ("create") fostered a deeper understanding and retention of CT skills. It allowed PSTs to see the immediate effect of their changes, thereby making the learning process more engaging and interactive. Ultimately, this structured approach equipped them with the necessary skills and confidence to develop their own programs, ensuring a comprehensive and effective learning experience.

2.3.4 Brennan & Resnick (2012) CT Framework

Brennan & Resnick (2012) designed a framework for studying and assessing the development of CT in the context of a visual programming environment, specifically Scratch. Their focus was on understanding ways in which programming of interactive media support the development of CT in learners. The framework is structured around three dimensions (3D), which include computational concepts, computational practices, and computational perspectives. I now present how the three dimensions of the framework were adopted and used in this study.

2.3.4.1 Computational concepts

Computational concepts are defined as CT concepts that learners employ as they write computer programs (Brennan & Resnick, 2012). The 3D Framework highlights seven computational concepts that are highly useful in Scratch projects and common across many programming and non-programming contexts. These CT concepts are sequences, loops, conditionals, events, parallelism, operators, and data. The scope of this study focused on the first three fundamental computational concepts.

Sequences are a key concept in programming that involves expressing a task or an activity as a series of instructions or steps for a computer to execute (Brennan & Resnick, 2012). The order of these steps is of utmost importance, as any modification can directly change the outcome or behaviour of the program. A solid understanding of this concept enables the participants to write programs that function as intended. Figure 4 is a visual program created by one of the participants groups during the programming sessions. This visual program is a sequence of computer instructions written in Scratch programming language. The computer executes these instructions in the exact sequence in which they have been written. For instance, when the green flag is clicked, the digital character, which is called a *sprite* in Scratch, moves 10 steps, then moves another 10 steps and finally moves another 10 steps before saying the word 'Hello' for 2 seconds. The sprite can never say the word 'Hello' before moving the 10 steps three times.

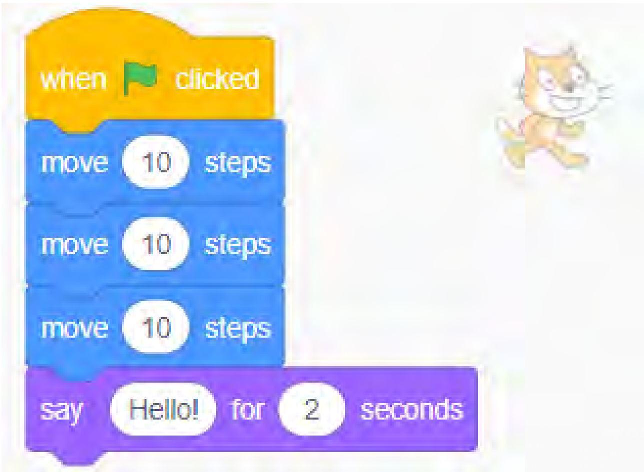


Figure 4: Sequence of code blocks representing computer instructions in Scratch

Loops represent another key computational concept in computer programming. They serve as a mechanism for executing a single set of instructions multiple times, as required (Brennan & Resnick, 2012). This eliminates the need to duplicate the same set of instructions each time an action needs to be reiterated. Consequently, mastering this computational concept equips participants with the ability to write visual programs that are not only scalable but also easy to manage. Figure 5 is an extract of a program created by one of the groups. This program is similar in functionality to the one in Figure 4. What distinguishes them is the loop implemented by a repeat block in in Figure 5. In this script, the three "move step" blocks have been replaced by a repeat block with the number of times the 'move step' instruction should be repeated. Clearly, loops reduce duplicate code which can make scalability and management significantly difficult.

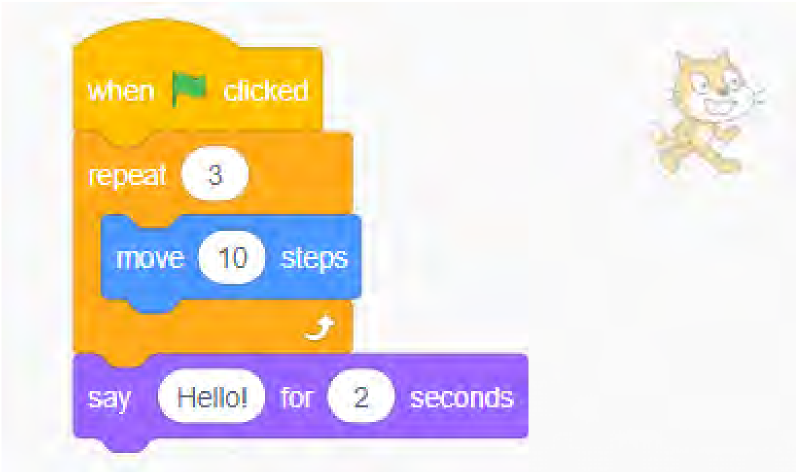


Figure 5: Scratch program with a repeat block (loop)

Conditionals represent another computational concept that enables computer programs to execute actions based on certain conditions (Brennan & Resnick, 2012). This allows programmers (participants) to write programs that support the expression of multiple outcomes. The Scratch program in Figure 6 is a visual program that one of the groups created when they were tasked to design and implement their own visual program like a Pong game that was explored initially. This program is an example of if-else (conditional) statement where an instruction (action) is executed based on the outcome of a condition. In this case, the 'basketball bounce' sound effect can only be played when the ball touches the green paddle, otherwise, the ball goes to a random position.

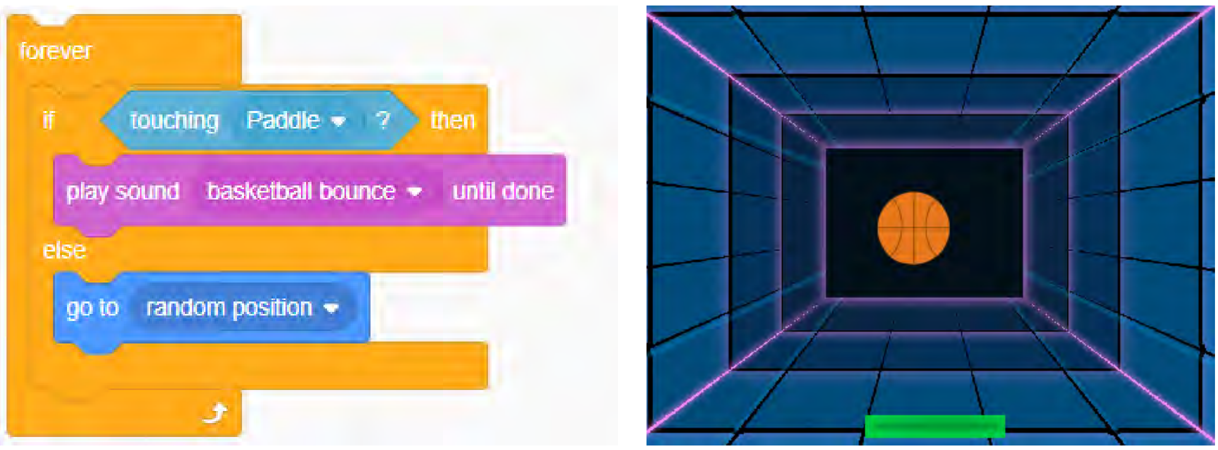


Figure 6: Scratch program with an if-else (conditional) statement

This concept is implemented in most of the computer systems that are used daily. An automated teller machine (ATM) is a good example in which the machine can execute one of the two possible actions; displaying ‘insufficient funds’ message or dispersing the cash, based on the outcome of a condition that the entered amount is less or equal to the available balance.

The 3D computational thinking framework provided essential CT concepts that participants used to create visual programs like animations, music, stories, and games. These CT concepts are so crucial to visual programming; without a solid understanding of them and their application in programming scenarios, it is impossible to develop a functional visual program (Brennan & Resnick, 2012). Every program relies on fundamental CT concepts such as sequences, loops, and conditionals. Therefore, understanding these concepts was crucial for creating and executing coherent, manageable, and functional visual programs.

2.3.4.2 Computational practices

As learners actively participate in programming using computational concepts, they gain *CT practices* (Brennan & Resnick, 2012). Resnick (2017) argued that these practices, often referred to as skills in other frameworks, are transferable. According to the 3D framework, learners develop four main practices while designing projects. The first practice involves being *incremental and iterative*. In the context of visual programming, learners recognize that project designing is not a straightforward, linear process. Instead, it is adaptive, involving stages like conceptualizing a project solution, devising a design plan, and then implementing the design through coding. This CT practice equips participants with the determinism to tackle projects step by step, minimising discouragement from failing to complete them all at once. This practice is relevant even in everyday problem-solving scenarios.

The second computational practice that learners develop is *testing and debugging*. This practice refers to developing strategies for anticipating and dealing with problems while programming (Brennan & Resnick, 2012). This is particularly important as things rarely work just as imagined, especially in computer programming. These practices are developed through trial and error, knowledge transfer - also called pattern recognition and support from knowledgeable others. This practice equips participants with skills to diagnose and resolve programming challenges.

Reusing and remixing is another fundamental CT practice that aid learners to find ideas and code to build upon. This approach enables them to potentially develop programs much more complex than they could have developed on their own. Moreover, it enhances the development of critical code-reading. Participants can actively engage with other people’s work.

The last computational practice involves *abstracting and modularizing.* This entails building something large by putting together collections of smaller parts (Brennan & Resnick, 2012). Abstracting simplifies complex projects or problems by focusing on relevant information and disregarding irrelevant details (Dong et al., 2019). Modularizing, on the other hand, involves breaking down complex processes or problems into smaller, more manageable components (Csizmadia et al., 2015). In visual programming, learners employ abstracting and modularizing at multiple levels, from the initial stage of conceptualizing the problem to translating the concept into code. By employing abstraction, participants can create efficient, easily maintainable, and scalable visual programs. It is worth noting that abstracting and modularizing are not unique to programming; they facilitate learning and aid problem-solving in everyday contexts as well.

In this study, participants worked on visual programs incrementally, realizing that completing some programs required multiple stages. Some groups took two to three sessions to finish their programming projects, starting with problem conceptualization, followed by designing and coding. When faced with coding errors, the PSTs were able to debug the errors on their own as well as by seeking guidance from the researcher (MKO). The researcher ensured that participants first explored all available debugging mechanisms, like use of prior knowledge (pattern recognition) and trial and error mechanism, before intervening. This approach was intended to enhance their independent testing and debugging skills. Allowing PSTs to use existing code segments enabled them to create complex visual programs efficiently. Given the time constraints, participants would have struggled to develop complex visual programs without incorporating other people's pieces of code. Furthermore, to manage the complexity of visual programs, participants developed programs, component by component. For example, in developing a complete Pong game, PSTs initially programmed the moving ball component, followed by the moving paddle component. These separate programs were later integrated and additional components, such as scoring points and sound effects were later added. This practice enabled the PSTs to focus on each task, ensuring that it is carried out efficiently and effectively. These CT practices were crucial in helping participants develop complex programs efficiently and effectively. Moreover, the skills gained through these practices are applicable to real-life problem-solving contexts as argued by Brennan & Resnick (2012), demonstrating the broader value of the activities provided in this study.

2.3.4.3 Computational perspectives

The dimension of perspectives within the 3D Framework describes the shifts that take place in the learners' understanding of themselves, their relationships with others, and the technological world around them (Brennan & Resnick, 2012). The first perspective called *expressing*, entails developing a mindset where computation is viewed as a tool for expressing oneself. As PSTs designed visual programs, their perspective evolved from perceiving themselves as mere consumers of interactive media to becoming active producers. The second perspective is termed *connecting*. Scratch programmers realise that creativity and learning are deeply social practices. The Scratch platform facilitates the collaborative creation of interactive media among learners. *Questioning* is the last perspective that learners develop as they engage in visual programming. They can ask questions to negotiate with the realities of technologies around them. During the programming sessions, PSTs experienced Scratch as a tool for expressing their own knowledge development and life experiences. For example, some participants created visual programs that depicted their personal feelings. The nature of the visual programming platform also facilitated collaboration among the PSTs by exposing them to a worldwide online community of visual programmers where programming ideas and skills are freely shared. In addition, the intervention allowed the participants to relate the CT practices to their everyday realities. Some PSTs felt that the practices are applicable beyond the classroom setting.

2.4 Conclusion

This chapter provided a comprehensive overview of the key concepts and frameworks relevant to the study. It began by defining computational thinking and identifying some of the recursive CT skills found in literature. Given the study's focus on visual programming, the chapter then reviewed relevant programming frameworks. Plugged and unplugged CT development approaches were explained and approaches for integrating CT were also explored, followed by a discussion on assessment mechanisms to understand how CT is evaluated. The chapter highlighted the benefits of integrating CT and examined various integration strategies in teacher education. The second part of the chapter introduced Vygotsky's sociocultural theory, which served as the primary theoretical foundation for the study. This section also covered professional development and pedagogical models, essential for the training of pre-service teachers involved in the intervention. Finally, the chapter concluded by presenting the Brennan & Resnick's (2012) CT framework. The next chapter discusses the research design and methodology of this study.

CHAPTER THREE
RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

The previous chapter presented literature and explored theories that underpin this study. This chapter presents the research design and methodology used in this study. According to Roller & Lavrakas (2015), research design is a systematic plan for conducting a study, outlining the procedures necessary to address research questions. Research design can either be qualitative or quantitative. On the other hand, research methodology is a strategy that is employed to implement a research plan. It guides the process of formulating a research problem, purpose, data collection, and the presentation of results from the collected data (Sk et al., 2020). This chapter is structured as follows: (3.1) Introduction, (3.2) Qualitative research design, (3.3.) Qualitative case study, (3.4) Research site and participants, (3.5) Research instruments, (3.6) Computational thinking sessions, (3.7) Data analysis, (3.8) Data triangulation, (3.9) Research evaluation, (3.10) Reflexivity and positionality, (3.11) Ethical considerations, and (3.12) Conclusion. The following research evaluation aspects were discussed: validity, credibility, transferability, and reliability. The research instruments used in this study were CT reflective tool, semi-structured interviews, focus group discussions and reflective journals.

3.2 Qualitative Research Design

Research design as defined by Akhtar (2016), is a deliberative overall plan which is systematically put together to effectively carry out a research study. Cohen et al. (2017) rightly observed that research is not an arbitrary enterprise, it is an exercise that precedes rigorous and scrupulous planning. A research design encompasses decisions about the research methods, data collection methods, sampling techniques, data analysis and ethical considerations among others. The literature presents three common categories of research designs; quantitative, qualitative and mixed research designs. Watson (2015) defines quantitative research designs as research designs that involve systematic investigation of phenomenon using statistical or numerical data. On the other hand, qualitative research designs are concerned with social enquiries that involve analysing non-numerical data to understand naturally occurring cases in detail (Pathak et al., 2013). Mixed research designs combine quantitative and qualitative research methods to answer research questions (Schoonenboom & Johnson, 2017). This study adopted the qualitative design located within the interpretive paradigm using a case study methodology. Qualitative research design that is located within an interpretive paradigm focuses on understanding the world from the individuals' experiences and interpretations

(Sandelowski, 2001). An interpretive qualitative design aims to form a detailed social enquiry that is driven by the goal of understanding a phenomenon within its context (Hammersley & Traianou, 2012). In this design, the meanings, or interpretations (subjectivity) that people make within their contexts form valuable knowledge. This research intended to understand the development of computational thinking from the perspective of pre-service teachers within their authentic learning environment. Qualitative research connotes a set of data collection methods that use words rather than numbers as were mostly used in this research (Bryman et al., 2008). According to Cohen et al. (2017), there are several purposes of qualitative research, such as, explanation, description, reporting, testing, theory generation and creation of key concepts. In this research, the qualitative research design was employed to primarily understand and explain the development of CT among pre-service teachers using visual programming within the specified context.

The qualitative research design was adopted in this study owing to its suitability for investigating contextual factors and complexities that influence human experiences. As previous studies have shown the effectiveness of developing CT among pre-service teachers through visual programming, particularly in developed countries (European Commission. Joint Research Centre., 2016; Stefania et al., 2022), it's important to note, as Gerring (2007) pointed out, that what works in one context may not necessarily yield the same outcomes elsewhere. Thus, the primary objective of this research was to uncover the specific contextual factors, both facilitating and restraining, that need to be considered when developing CT among pre-service teachers using visual programming, in a developing country context. Qualitative research prioritizes peoples' perspectives, and experiences, thereby empowering participants to contribute to knowledge creation. As observed by Moloi et al.(2023) the product of a well-executed qualitative research design is a set of contextually grounded insights that can inform practice, theory and policy making. It is expected that the findings of this research may inform the integration (practice) of CT into teacher education at Rhodes University.

This research exhibits several fundamental characteristics of a qualitative research design. First, qualitative research is premised on participants' interpretations. People are regarded as meaning-making beings who can make sense of their world. Qualitative researchers believe that meanings (knowledge) are a product of social situations, interactions and negotiations (Cohen et al., 2017). It is against this understanding that the views of the participants in this study formed the primary source of data. In addition, the researcher engaged in meaningful interactions with the participants in order to make-sense of the phenomenon under investigation. For example, the researcher conducted semi-structured interviews and focus group discussions, in which both the researcher and the participants actively contributed. Furthermore, both the researcher and the participants were writing reflective journals to capture their sense-making. Second, in qualitative research, the researcher is a critical instrument of the research process (Wa-Mbaleka, 2020). They immerse themselves in the research processes so that the distinction between them and the participants is blurred. This allows the researcher to identify with or understand the experiences, interpretations, or meanings that people make. This is why the researcher was actively involved in the entire study from data collection to analysis, while making sure that there was no clear distinction between him and the participants. Third, qualitative design perspectives stipulate that knowledge is greatly influenced by culture and is context bound (Yilmaz, 2013). This notion posits that knowledge is not independent of the space and culture within which it is situated. Therefore, it is important to take the cultural elements of the participants into account when planning and conducting a study. The researcher used a technological tool (Scratch)to understand the development of CT on the premise that technology is part of today's society, especially among PSTs who participated in this study (Szymkowiak et al., 2021). Another fundamental characteristic of qualitative research is a methodological perspective that social research should be conducted in natural, real-world settings as context has strong implications on meaning (May & Perry, 2022). This study was conducted within the case of BEd Foundational phase first year students, where the researcher was able to observe the development of CT among pre-service teachers within the natural context of a classroom environment.

3.3 Qualitative Case Study

Cohen et al. (2017) note that there are many definitions of a case study in the literature and argue that attempting to arrive at a single definition is elusive and unnecessary. However, Bassey (1999) describes a case study as a detailed investigation of an individual, group, or a phenomenon (unit) to uncover interdependencies and emerging patterns within the unit. One of the broadly cited advantages of a case study, as highlighted by Yin (2009) is its ability to provide a rich and detailed understanding of a phenomenon that may be lost in other research methods. Case studies are characterised by thick descriptions and are implemented to answer "how" and "why" questions (Shambare et al.,2022). In alignment with the goal of this study, which seeks to explore the development of CT in the context of pre-service teacher training at Rhodes University, the utilization of a case study methodology is deemed appropriate. The case study methodology allowed this study to explore a comprehensive exploration of specific enabling and constraining factors that should be considered when developing CT in the context under investigation. The chosen case study methodology was particularly significant in this research because the effectiveness of an approach that has been successful in integrating CT in developed countries may yield different outcomes in a developing country context. Additionally, the developing country context may present unique facilitators and challenges to the integration of CT. Therefore, by employing this methodology, the study was able to explore the specific context and provide valuable insights into the intricate details and complexities of CT development.

The literature presents an array of limitations that are associated with a case study methodology. Wellington (2015) commented that case study approaches are not replicable due to their reliance on context. He adds that case studies may not be representative or generalizable. Polit & Beck (2010) define generalization as drawing broad inferences from particular observations. It entails extrapolation of representative sample findings to an entire population. The researcher understands that the unit of analysis (case) in this study is unique and not a representative sample of a large population. Therefore, the findings of this study only apply to the year one BEd FP pre-service teachers who participated in the study at Rhodes University and cannot be simplistically generalised to a larger population. However, Yin (2013) argues that despite the lack of generalization, findings from a case study provide insights into other similar cases, thereby helping the interpretation of similar situations. It is against this argument that the researcher holds the view that the findings of this research inform other similar cases within Rhodes University and beyond.

3.4 Research Site and Participants

3.4.1 Research Site

According to Cohen et al. (2017, p.154), a "decision on what and where to do research can arise from a problem that has emerged in the locality, perhaps due to government policy or practices to local developments." The lack of trained teachers to teach the newly introduced subject of coding and robotics in the Sarah Baartman District is what inspired the conception of this research idea, as stated in Section 1.4. As such, the goal of this research was to explore the development of CT among first year BEd FP preservice teachers using visual programming at Rhodes university. For this reason, Rhodes University, a teacher training institution situated within the Sarah Baartman District Municipality in the Eastern Cape Province of South Africa, was identified as an ideal site for the research. The intervention was implemented within the Department of Primary and Early Childhood Education at the institution.

3.4.2 Selection of Participants

In traditional research, sampling is often understood as defining a subset of a population to obtain data that is representative of the entire population (Levy & Lemeshow, 2013). However, in case studies and qualitative research in general, sampling takes on a different meaning. Uprichard (2013) observes that in case studies, sampling involves selecting a population or specific cases from which in-depth knowledge will be gained. The focus is not on generalization to a larger population but on gathering detailed insights into the specific case being investigated. Guided by the objectives of this study, the researcher used purposive sampling to select the participants. As described by Conger et al.(2017), purposive sampling involves deliberately selecting cases or participants based on the specific needs or characteristics that align with the research objectives. In this study, the selection of pre-service teachers was purposefully done.

Regarding sample size, as argued by Cohen et al. (2017), the question of sample size has no clear-cut answer, as the size depends on a variety of factors in a qualitative research. One such factor is the purpose of the research. As stated in Section 1.4, the purpose of this interventionist form of research was to equip pre-service teachers with computational thinking skills, preparing them for the teaching of the newly introduced coding and robotics subject in their future classrooms. Therefore, all the willing first-year pre-service teachers who enrolled for the *Introduction to Computers* module took part in the study. The total number of the pre-service teachers who consented to participate in the study was forty-nine (49). However, the researcher purposively selected twelve students out of the total sample for semi-structured interviews and focus group discussions. As argued by Boddy (2016), in qualitative research, the sample size can be influenced by the available time and resources.

The following requirements (i) and (ii) were used as criteria for selecting the first 49 participants and requirement (iii) was used as criterion for selecting the 12 participants.

i. Must have been Bachelor of Education (BEd.) foundation phase pre-service teachers.
ii. Must have enrolled in the introduction to computers module.
iii. Must have been attending the CT sessions regularly.

Since the introduction to computers module is not credited and attendance was optional, the researcher used purposive sampling to select 12 participants who regularly attended CT sessions and were willing to participate in the semi-structured interviews and focus-group discussions. This approach ensured that the data gathered gave a comprehensive and accurate reflection of the participants' experiences and perspectives (Moser & Korstjens, 2018). Table 1 shows the participants' demography.

Table 1: Participants' Demographic Information

Gender	Age Range	Number of pre-service teachers	Percentage
Male	17 - 24	12	24%
Female	16 - 22	37	76%

3.5 Research Instruments

The use of multiple data collection tools is indeed encouraged in research as it allows for data triangulation, which enhances reliability and credibility of findings (Sepula & Simuja,2024). In this study, three qualitative data collection tools and one quantitative assessment tool were employed to facilitate comprehensive analysis, address research questions from various perspectives, and gain an in-depth understanding of the phenomenon under investigation. The researcher used a CT fostering and evaluative instrument called CT Reflective tool, semi-structured interviews, focus group discussions, and reflective journals to collect data.

3.5.1 CT Reflective Tool

The CT reflective tool is an adapted instrument derived from validated CT tests. Like the traditional CT tests, one of the functions of this tool is to assess the development of CT. However, its overarching goal goes beyond mere evaluation, it seeks to foster the development of CT among participants. Moreover this tool offers objectivity by reducing researcher bias or subjectivity in data interpretation and allows for standardized tracking of changes in research participants (Pandey & Pandey, 2021).

Despite the highlighted advantages, the researcher was very conscious of the shortcomings that are inherently associated with this instrument. For instance, given that it takes the nature of a test, the tool is associated with the 'Hawthorne effect'. Cook (2005) observed that the realization that the participants' knowledge and skills will be tested brings anxiety, which influences their behaviour in the study. To minimise this effect in this study, the researcher intentionally used the name "CT reflective tool" instead of the traditional name "CT test". This not only removed the test anxiety, but it also helped the participants to view the instrument as a fostering other than a measuring tool. Secondly, as noted by Booth (2006), participants are not always clear on what is being asked or required in the activities or questions. Román-

González et al. (2017) add that participants may know the right answer but fail an activity or question because there is lack of clarity. In view of this important factor, the researcher was always available and would walk around the classroom during the CT activities to offer clarification when needed. Furthermore, the researcher would clarify every activity before the participants start working on them. Another critic that is levelled against researchers conducting their own tests is that they may unconsciously resort to simplistic testing rather than richer and more extended assessments (Cohen et al., 2017). This may also lead to the questioning of the test items validity. As a way of mitigating this challenge, the researcher used CT activities from internationally recognized and validated tests. Participants' motivation and interest in the test activities affects performance (Piaw Chua, 2012). The motivation is strengthened through clearly explaining the purpose of the test which the researcher comprehensively did at the beginning of the intervention.

The design of this tool was guided by Brennan & Resnick's (2012) 3D framework, adopted in this study to facilitate and evaluate CT development within the context of a visual programming platform and it comprised three sections. The first two sections consisted of activities that evaluate and augment the first three computational concepts described in the analytical framework, i.e. sequences, loops, and conditionals. Given the participants' limited computing background, the first set of these activities was designed as stand-alone, independent of any computer programming syntax - a format proposed by Zapata-Caceres et al. (2020). The second set of activities engaged the same computational concepts but presented them within a visual programming context, with a focus on the underlying concepts rather than the Scratch syntax. These activities were adapted from a validated and widely used CT test available on the Callysto website (https://callysto.ca/computational-thinking-tests/). The last section contained activities that evaluate and foster the acquisition and transferability of CT practices or skills, as outlined within the analytical framework. These activities were adapted from "Bebras competition," an international test that assesses learners' ability to apply CT in real-life problems (Dagiene & Futschek, 2008). Furthermore, the CT Reflective tool included a qualitative open-ended question to gain insights into how pre-service teachers understand CT.

Following the SCT theory notion that learning is socially mediated, the researcher divided the participants into groups of three to work on the CT activities (Vygotsky, 1980). This ensured that the participants could collaborate and help each other understand and develop CT concepts and skills respectively. As stated in Section 2.2.2.2, the CT reflective tool (see Appendix A1) was administered before and after the intervention to track changes in the participants' CT concepts and skills to ascertain learning in the ZPD and address the first research question i.e. How do pre-service teachers develop computational thinking with visual programming? And triangulate the other methods used to collect data such as Semi-structured interviews, Journal reflections and Focus group discussions.

3.5.2 Semi-Structured Interviews

A semi-structured interview is a data collection method that combines standardization and flexibility to gain rich and detailed data from participants (Ahlin, 2019). In semi-structured interviews, the researcher (interviewer) is guided by pre-determined open-ended questions with room for controlled deviations, prompts, and probes to obtain a deeper understanding of the participant's experiences or perspectives (Grossoehme, 2014). As noted by Simuja(2024;2023), through the probes and asking for elaboration from interviewees, researchers can obtain deeper insights of the contextual factors influencing participant's views. It helps researchers to capture the complexity of a phenomenon within its broader context.

The topics and questions are prepared in advance with some flexibility in terms of the wording and the sequencing of the open-ended questions, tailored to each individual interviewee and the given responses. The framing of the semi-structured interview also considers the possibility of prompts and probes (Ahlin, 2019). Prompts enable the interviewer to clarify by rephrasing or repeating questions, for example, especially when the interviewee has not understood the question. Opoku et al. (2016) caution that prompts should be used carefully as they may falsely assume that the interviewee has not understood the question. The interviewee can also prompt the interviewer to provide more details, clarify or seek more guidance. Whereas probes in semi-structured interviews enable the interviewer to ask the interviewee to provide more detail, elaborate, exemplify, add to, extend or qualify their responses, thereby addressing richness, comprehensiveness, and depth of responses that are hallmarks of a successful interview. Another scholar offers a cautionary note concerning the use of prompts and probes, arguing that over-probing may lead to interview bias and may provoke resentment from the interviewee (Doody, 2013). He further suggests that a successful interview should have controlled prompts and probes. Another crucial element that influences a successful interview is the issue of power dynamics between the interviewer and the interviewee (Mikecz, 2012). The two parties are to work as partners in which knowledge is socially co-constructed.

In this study, the semi-structured interviews followed an interview guide (see Appendix A2) with controlled prompts and probes. To create a good rapport and mitigate power imbalances, the researcher clearly stated at the beginning of every interview that he was going to learn from the interviewees' views and perspectives. In addition, to make interviewees feel equally important and have a sense of power, the researcher made sure that the interviews were conducted at the interviewees' convenience. The nature of this research whose primary goal was to explore how the targeted pre-service teachers develop CT from their perspectives and uncover contextual factors that aid the process, necessitated the use of semi-structured interviews. Adams (2025) posits that semi-structured interviews are suited for gathering data on people's views, perspectives, and experiences, which cannot be collected as richly in other data collection methods. The researcher conducted the interviews right after the intervention and he audio-recorded the interviews and took complementary notes in his notebook. Due to factors beyond the researcher's control, only ten semi-structured interviews were conducted against the planned twelve interviews. During the interviews, two participants who consented to participate in the semi-structured interviews got very busy with academic work, so they were not available for the interviews. Despite this, the remaining sample size of ten for semi-structured interviews is believed to be enough to provide meaningful insights. The audio recordings were later transcribed using the Microsoft O365 transcription tool, a Microsoft office word feature that allows people to transcribe audios into text files for analysis. The transcription files were imported into NVIVO software, a qualitative data analysis computer software package, for analysis. This data collection instrument was intended to address the first and second research questions i.e. *1. How do foundational phase pre-service teachers develop computational thinking with visual programming? 2. What are the enabling and constraining factors that affect the development of computational thinking using visual programming approach among foundational phase pre-service teachers?*

While there were many significant advantages of using semi-structured interviews in this research, the researcher recorded some challenges that were associated with this method. Firstly, the researcher noted that semi-structured interviews are time-consuming. The interviews required more time which was a big challenge especially considering that the interviewees were students who had equally time-demanding academic obligations. The cleaning of the transcribed files also proved to be time-consuming. Secondly, it was noted that due to the rich content responses collected from the interviews, analysis of the data was complex. It required in-depth coding and interpretation, which was labour-intensive and time-consuming. Thirdly, despite the researcher's effort to create a natural and free interview environment where an interviewee provides honest responses, it was observed that some interviewees were pressured to give responses that they thought the interviewer was expecting. Some interviewees were very conscious of not giving wrong answers. Similarly, the researcher's efforts to mitigate power imbalances were limited. Even after clearly stating that the researcher was not more expert and was there to learn from the participants, some interviewees still perceived the researcher as more expert than them. However, despite the recorded disadvantages, the advantages of this data collection method remain valuable for this study and offer triangulation to other methods of collecting data such as Focus Group Discussion.

3.5.3 Focus Group Discussions

Denscombe (2017) describes a focus group discussion as a form of a group interview used by researchers to collect data that emerge from the interaction of the participants on a specific topic of research. Focus group discussions are a powerful tool that allows participants to build upon each other's responses, challenge ideas and explore their perspectives more deeply. Consequently, focus group discussions are known for uncovering new insights that could not otherwise emerge (Cohen et al., 2017). Breen (2006) notes that focus groups yield a collective rather than an individual view and the interactions are between the participants and not with the interviewer. Sepula & Simuja (2003) describe a focus group as an unnatural setting that brings together a chosen sample of a population to discuss a particular given topic, where the interactions lead to data. A facilitator is present in focus groups to steer the discussion and most importantly to keep the members to the focus of the discussion. Gibbs (2012) posits that focus groups are suited for collecting data about the participants' attitudes, perceptions, experiences, opinions, and viewpoints in a way that would not be feasible in other data collection methods. She adds that focus groups are particularly useful when there are power imbalances between the researcher and the participants, which was the case in this study. The researcher (lecturer) in this study and the student participants presented some power imbalances that had the potential of compromising the data quality. Therefore, the focus group setting provided an environment where participants could express themselves freely without feeling obliged to please the lecturer. Moreover, focus groups are well known for their ability to provide a space where participants are able to question each other's perspective, as well as re-evaluate and reconsider their own understandings. This allows the researcher to collect refined perspectives, views, and opinions of the participants regarding the research problem.

Although focus groups presented many advantages, the researcher noted that as with all methods, there are limitations unique to this data gathering method. Firstly, despite providing the topic of discussion to the group, it was noted that the researcher had less control over the data produced than in semi-structured interviews. Sometimes participants would digress from the topic, leading to less focused data. Secondly, it was observed that the group dynamics had a negative impact on the collected data. Some participants who were more vocal, articulate, and confident seemed to intimidate and overshadow the quieter voices in the discussions, skewing the data to their opinions and perspectives. To mitigate some of the limitations inherent in this method, the researcher provided a clear explanation of the purpose of the group and the topic of discussion to minimize tangent discussions. Secondly, the researcher promoted the discussion by asking open-ended questions and challenging the group by drawing out participants' differences on a particular debate. Sometimes the researcher would steer the conversation back on course when there is a drift to ensure the focus of discussion. To ensure that everyone contributed, the researcher would give a chance to less confident and quite participants to speak, allowing the data to be more representative. Lastly, during the entire moderation of the discussions, the researcher avoided giving personal opinions so as not to influence participants towards a particular position.

Regarding the size of the group, MacIntosh (1993) contends that the recommended number of participants per group is six to ten. As such, after conducting the semi-structured interviews with the participants, the researcher held two focus group discussions consisting of six pre-service teachers. The topic of discussion was the participants' understanding of CT, how to develop it, and the enabling and constraining factors that should be considered when developing it using visual programming (see Appendix A2). The participants discussed these areas based on their experiences from the intervention. Just as pointed out by Denscombe (2017), the focus group discussion helped this study researcher capture insights that could not be uncovered during semi-structured interviews and reflective journals. This data collection tool was employed to answer research questions 1 and 2.

3.5.4 Reflective Journals

A reflective journal serves as an authentic written record where researchers and participants document their thoughts, experiences, and reflections in practical settings (Choi, 2020). One significant advantage of using reflective journals as a data collection tool is their ability to provide an audit trail of the research design (Vicary et al., 2017). By utilizing reflective journals, researchers can explicitly map or disclose their role, perspectives, or biases in the study, thereby promoting transparency and reflexivity in the research process.

Russell & Kelly (2002) define a reflective journal as a written record of individual's thoughts, feelings, experiences, and insights throughout the research process. It includes details on what participants did, thought and felt during the study. Reflective journals make the research process visible, which enhances the transparency of the research design. Janesick (1999) comments that reflective journals allow participants to dig deep and reflect on their perceptions. In addition, reflective journals provide a mechanism for tracking change in the perceptions of participants (Simuja,2024). Through the journals that participants write about their perceptions on a regular basis, they are able to see the change in the perceptions. In this study, the participants and the researcher were writing reflective journals after each session to capture thoughts, experiences, and reflections on the development of CT using visual programming. These journal entries (see Appendix A3) were triangulated with other data collection tools to enhance the validation and trustworthiness of the research findings. The data from the journals were used to answer research questions 1 and 2.

3.6 Computational Thinking Sessions

As the lecturer responsible for the "Introduction to Computers" first-year module, the researcher implemented the intervention in the form of a series of lectures within the module. A total of twelve sessions of 60 minutes each over a period of six weeks were conducted in the participants' natural setting of a computer laboratory. The first session served as an introduction to computational thinking for the participants. During this session the rationale for integrating CT in primary education was articulated. In the second session, the CT reflective activities were administered to the participants in groups of three to four. During the following nine sessions, hands-on sessions with *Scratch* programming language were conducted. The CT practical sessions were underpinned by the "use-modify-create" pedagogical model and Brennan and Resnick's CT framework adopted in this research dictated the content (Brennan & Resnick, 2012; Lee et al., 2011).

Firstly, the participants were able to engage with pre-made Scratch programs like games and animations on the online Scratch platform to understand how Scratch programming works (Scratch Foundation, 2023). In the following sessions, participants were able to make changes to the existing programs. After the first two pedagogical stages of 'use' and 'modify', the participants began to create their own Scratch programs as suggested by the model (Lee et al., 2011). The programs were created by employing the computational thinking concepts described in the Brennan's 3D framework given in Section 2.2.4. The programs ranged from moving digital characters (sprites) to games, animations, music, and stories. For example, participants were asked to create a moving sprite using sequences and loops. At some point, participants created a 'Pong game' by employing sequences, loops, and if-else conditional statements. Upon completion of the CT practical sessions, the CT reflective activities described in Section 3.5.1 were administered for the second time. Finally, the researcher and the participants solved the CT reflective activities together. This was intended to help the participants who had challenges with the CT reflective activities, thereby fostering their CT skills.

3.7 Data Analysis

According to Taylor & Gibbs (2010), data analysis involves techniques that allow researchers to move from raw data to understanding, explaining and interpreting the phenomenon under investigation. The aim of data analysis is to make-sense of the data collected in the study and draw conclusions. Khoa et al. (2023) note that data analysis exercise takes different shapes depending on the research design, paradigms, type of data, and research questions used. Adu (2019) characterizes qualitative data analysis as a rigorous process of making meaning of what has been said, written and done. According to Castleberry & Nolen (2018), qualitative data analysis involves several stages such as preparing and organizing data, describing and presenting data, analysing the data and interpreting the data.

During the first stage, the data is put into formats that make for easy analysis. Among others, this process involves transcribing audio data into text files for cleaning and analysis. In this study, the audios from semi-structured interviews and focus group discussions were transcribed using the Ms O365 transcription feature. The transcription files were cleaned where missing texts were filled, and any grammatical errors were corrected. Adu (2019) presents two types of qualitative data analysis which are thematic and content analysis. The later entails generating codes before analysing the data while the former involves generating codes and themes from and during the data coding process. This study adopted the thematic analysis where the researcher began by familiarizing himself with the data. In thematic analysis, the researcher begins by immersing themselves in the data to fully understand its depth (Castleberry & Nolen, 2018). This involves reading and re-reading the data, while noting down initial ideas. Themes were generated during the coding (identification and description of relevant data) process. Figure 7 is a visual representation of the data analysis process employed in this study as suggested by Adu (2019) in his book titled 'A step-by-step guide to qualitative data coding'.

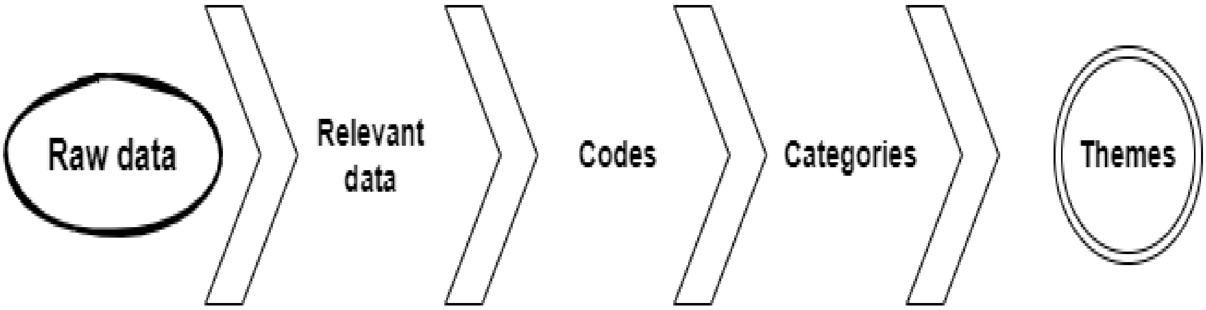


Figure 7: Thematic Data Analysis Process

Following this analysis design, the raw data which included transcription files from semi-structured interviews and focus group discussions, and the journals written by the participants and the researcher were imported into NVivo software. Guided by the research questions and the theoretical frameworks, the researcher identified important information (relevant data). The relevant data were carefully assigned labels (codes). These codes were then put into categories based on their similarities. Finally, the formulated categories were given befitting labels that formed themes. The quantitative data (formative assessment scores) were analysed through the lenses of the 3D framework. Basic descriptive statistics were computed using Microsoft Excel to observe possible shifts in the participants' CT knowledge and skills. The theoretical frameworks adopted in this study played a pivotal role in the analysis and interpretation of data. For example, using the qualitative data, the 3D framework helped the researcher in identifying shifts in the pre-service teachers' computational thinking perspectives. This framework also served as a guide or a marking scheme in scoring the CT reflective activities, ensuring a consistent and accurate evaluation. Furthermore, the sociocultural theory and the professional development and pedagogical models, provided critical lenses for interpreting the qualitative data. These theories facilitated a thorough and comprehensive understanding of CT development through visual programming, contextualizing the data within established educational theories. Employing these theoretical perspectives, the researcher was able to systematically analyse the data, discovering nuanced insights into the cognitive and social dimensions of CT development among pre-service teachers.

3.8 Data Triangulation

In its literal sense, it is said that triangulation is a technique in which military strategists use multiple locational markers to pinpoint a single spot(Carter et al., 2014). Drawing from this original sense, triangulation in qualitative research simply means the use of two or more data sources, methodological approaches, and theoretical perspectives within the same study (Bans-Akutey & Tiimub, 2021). Triangulation is also viewed as a strategy for testing validity of research through the convergency of information from different sources (Carter et al., 2014). According to Noble & Heale (2019), triangulation is a process that increases credibility and validity of research findings by using a variety of methods. The notion of triangulation is premised on the argument that over-reliance on one method may lead to distorted or biased results as research methods are never neutral, they act as filters through which reality is experienced (Donkoh, 2023). Thurmond (2001) highlights that the essence of triangulation is to minimise or counterbalance the deficiency or limitations of a single strategy thereby ensuring validity and trustworthiness. In her definition of triangulation, Thurmond exposes five types of triangulations which include theoretical, methodological, data sources, investigators, and data analysis methods triangulation. However, she was quick to mention that a researcher is not obliged to implement all the types of triangulations in their research as the use of many triangulation approaches does not necessarily guarantee validity and flawless study. Researchers are encouraged to employ triangulation approaches that contribute to the understanding of the research problem.

In this study, the researcher used multiple triangulations (two or more types of triangulations within the same study) to provide more insights to better understand and explain the phenomenon. Firstly, theoretical triangulation was used by employing Vygotsky's social-cultural theory, Brennan's 3D CT framework, 'code-connect-create' professional development model and the 'use-modify-create' pedagogical model to analyse and interpret data for better understanding of the phenomenon. Secondly, method or instrument triangulation which is described by Cohen et al. (2017) as the use of multiple data collection instruments to address the same phenomenon, was also employed in this research. Multiple data collection tools such as semi-structured interviews, focus group discussions, reflective journals and a quantitative data collection method called CT reflective tool were used to address the research questions. A type of data source triangulation where data was collected from two focus groups and the same semi-structured interview questions were administered to eight participants. This was done to confirm and increase the validity and credibility of the research findings.

3.9 Research Evaluation

To ensure that the findings of this study are trustworthy, the researcher considered the research aspects of validity, reliability, reflexivity, and positionality as criteria for research evaluation. As suggested by Ary et al. (2010), validity was addressed by ensuring two research aspects of credibility and transferability.

3.9.1 Validity

Noble & Smith (2015) define validity as the extent to which the findings accurately reflect the data. As argued by Golafshani (2015), validity is an important key aspect of effective research. In other versions, validity is understood as the degree to which an instrument measures what it intends or claims to measure (Junior et al., 2019). Golafshani (2015) comments that validity refers to the extent to which data interpretations or research findings are warranted by the theories and the evidence given. Shadish et al. (2002) identify four main types of validity: construct validity, statistical conclusion validity, internal validity and external validity. Construct validity is about the soundness of inferences made about the nature of theoretical constructs or factors. Whereas statistical conclusion validity is the use of suitable statistical functions to make determinations or conclusions. Internal validity is defined as the validity of the identified relationships between the elements of the research design and outcomes. On the other hand, external validity is about generalizability, transferability and comparability of the research findings.

Going through the literature on validity and reliability, it is evident that these two notions convey different meanings in quantitative and qualitative research. In quantitative research, among others, validity aims to achieve replicability, consistency, predictability, generalizability, and objectivity. While in qualitative research, validity is achieved through credibility (instead of truth value, a quantitative concept of internal validity), transferability (replacing generalizability, a quantitative external validity concept), dependability (replacing consistency, a quantitative concept of reliability), and confirmability (replacing neutrality, a quantitative concept of objectivity) (Ary et al., 2002). This study aimed to meet the described criteria for achieving validity. For instance, to achieve construct validity, the researcher relied heavily on literature in the theoretical and methodological approaches.

3.9.1.1 Credibility

According to Liao & Hitchcock (2018) credibility is an internal validity concept within qualitative research, a replacement of truth value in quantitative research. It entails the ability of the study to report a situation through the eyes of the participants (Liao & Hitchcock, 2018). Simply put, credibility is about the authenticity and trustworthiness of the data. To ensure credibility of the data and research outcomes of this study, the researcher took several deliberate steps. Firstly, the researcher took the findings of the study to the participants to verify if they accurately reflected their perspectives (member checking). During this process, the participants had the chance to add further information and correct factual errors. This also allowed the participants to check the adequacy of the analysis. Secondly, the researcher employed triangulation of data collection methods to gather rich data from which reflective outcomes were drawn. The researcher used CT reflective tool, semi-structured interviews, focus group discussions and reflective journals to understand how pre-service teachers develop computational thinking skills through visual programming. The multiplicity of the data collection methods helped the researcher to verify outcomes that accurately reflect the participants' perspectives.

3.9.1.2 Transferability

As defined by Lincoln & Guba (1985), transferability refers to the extent to which qualitative research findings can be compared and transferred to other contexts. Transferability is the replacement of generalizability, an external validity concept in quantitative research. Ary et al. (2010) argues that the interest in qualitative research is not so much to generalize the findings to a wider context, rather to provide sufficiently rich data for the readers of the research to determine the possibility of transferability. Cambon et al. (2012) comment that it is imperative in qualitative research to provide clear, rich, detailed, and thick descriptions of the context and the participants (situation) so that readers can decide the extent to which findings can be transferable to another situation. Similarly, Slevin & Sines (2013) posit that the role of a qualitative researcher is to provide typicality of the context and the participants of the research so that the reader can identify possible comparisons and determine how data can be translated into different settings and cultures. In this study, the researcher provided enough typicality of the participants and context to allow readers to determine the extent to which findings can be transferable to other situations. For instance, the computer literacy skill set of the participants was determined and provided in this thesis. This allows readers to compare participants and determine the degree of transferability of the research findings to their own participants and context.

3.9.2 Reliability

Reliability is a term that denotes dependability, consistency, and replicability over time, instruments and groups of respondents (Cohen et al., 2017). A research study is said to be reliable if it demonstrates that if it were to be carried out again on the same group of participants and within the same context, it would yield the same results. Essentially, reliability is concerned with precision and accuracy. Ary et al. (2010) note that canons of reliability differ in quantitative and qualitative research. Unlike in quantitative paradigms, where reliability means stability, consistency, and replicability of research, in qualitative research, reliability entails dependability of the research findings. Similar to internal validity, reliability in qualitative research is achieved by member checking, triangulation, and prolonged observations. Further, reliability in qualitative research can be addressed by involving another observer with the same theoretical framework, observing the same phenomenon. In this regard, the researcher involved the Researcher Supervisor especially during coding stage to make sure that the themes that came out were reliable.

3.10 Reflexivity and Positionality

Researchers play a crucial role in the creation of knowledge in research studies, hence, the need to look at themselves and their positionality (Wa-Mbaleka, 2020). It is important to recognize that power dynamics within the researcher-participant relationship are rarely symmetrical, with those possessing more power, resources, and information, studying those with less (Hammersley & Traianou, 2012). These power dynamics along with potential biases, and subjectivity can impact the research process and findings (Bettez, 2015). Thus, it is essential for researchers to acknowledge and address these potential influences. In this study, as a Lecturer of the selected BEd FP pre-service teachers and as a researcher, I was aware that my position could introduce power dynamics that could affect my interaction with the participants, who were my students. To mitigate this risk, I took specific measures to establish rapport and trust with the pre-service teachers. I clearly explained to them that I was conducting the research *with* them, rather than *on* them. This helped to create a collaborative environment and promoted a sense of partnership between the participants and myself. Furthermore, at the onset of the research, I emphasized that I was a co-leaner in this process. I conveyed to the participants that we would explore the phenomenon together, acknowledging that their insights and experiences would contribute to the research. Additionally, I clarified to the participants that their involvement in the study would not be credited or contribute to their class marks. This ensured that their participation felt natural and voluntary, without any pressure of potential consequences.

3.11 Ethical Considerations

Ethics is a set of norms or principles that guide researchers to uphold the rights of participants throughout the research process (Hammersley & Traianou, 2012). It is about carrying out research in a manner that preserves the dignity of participants as human beings (Cohen et al., 2017). Brooks et al. (2014) highlight that ethics run throughout the entire research process and it is contextual. Ethical norms are socially, politically, institutionally, and personally situated. As echoed by Simons & Usher (2000), ethical principles are independent of universalism, they vary in different parts of the world. To uphold and preserve the rights of research participants in this study, the researcher observed all ethical standards and protocols. These standards are according to the policies and procedures stipulated by the *Rhodes University Ethical Standards Committee* and the *Education Faculty Higher Degrees Committee* (see Appendix C2, Ethics Clearance Letter). This study prioritised care and respect for the participants and all ethical issues were upheld. In particular, the researcher focused on the informed voluntary consent, gaining access and confidentiality.

3.11.1 Informed voluntary consent

The research participants were informed about the potential benefits and risks of their participation in the study before their engagement in the study commenced. The letter of information and consent (see Appendix B1, Letter to Pre-service teacher participants) was guided by the Rhodes University ethical standards and requirements. The letter of information and consent outlined the research purpose, intervention structure, procedures, potential benefits and risks, identity protection, use of data, and other information relevant to ensure informed consent. The researcher reviewed the letter of information and consent with the participants before they signed or gave consent. Moreover, the researcher informed the participants that they had a right to decline to answer questions or withdraw from participating at any point in the study. Participants had the chance to review their interview transcripts. Later, the participants will be given copies of all documents and publications connected to this study including this thesis. The participants' engagement in this study was entirely voluntary without any threat or force.

3.11.2 Gaining access

To get permission to conduct the study with the BEd FP pre-service teachers, the researcher wrote access letter requesting permission from the Rhodes University Registrar (see Appendix B2, Letter to the Registrar). The access letter explained the research title, goals, methods, and the potential benefits of the study to the University, schools, pre-service teachers, and future learners. The researcher's obligation to keep participants' identities anonymous and maintain data confidentiality was explained. Lastly, the commitment to provide the registrar's office with an electronic copy of the research publication was made.

3.11.3 Confidentiality

Participants' privacy and confidentiality are safeguarded by the Rhodes University policies and protocols and identifiable information such as names has been anonymised. Participants were given the opportunity to choose their own pseudonyms for anonymity in the study publications. Some participants left the pseudonym choice in the hands of the researcher. This made the researcher search on the internet for names that are common in the participants' tradition and culture. All the collected data from the various instruments have been kept in a safe place at Rhodes University and will be destroyed after five years. All electronic files are kept on a password-protected computer by my research supervisor.

3.12 Conclusion

This chapter defined qualitative research, located within the interpretive paradigm using an interventionist form of case study methodology. It provided the descriptions of the research site, the sample and the basis for their choice. It also discussed the rationale behind the chosen methodologies, including their strengths and inherent shortcomings. The chapter also discussed the data collection methods, which included CT reflective tool, semi-structured interviews, focus group discussions, and reflective journals. The chapter further discussed data analysis, triangulation, research evaluation techniques such as validity and reliability. Two validity aspects were discussed including how they were applied in this study. Reflexivity and positionality as aspects were discussed and the chapter was concluded by addressing ethical considerations of this study. Three ethical elements of informed voluntary consent, gaining access, and confidentiality were explained particularly with regards to how they were upheld in this study. The next chapter discusses the findings of this study.

CHAPTER FOUR
FINDINGS

4.1 Introduction

The previous chapter tackled the research design and methodology used in this study. A qualitative case study approach was employed to allow a contextual and in-depth study of pre-service teachers' perceptions of CT development using visual programming. In this chapter, the researcher presents and analyses the collected data thematically, offering a detailed examination of the emerging themes and insights. The chapter consists of five parts: Chapter (4.1) Introduction provides an overview of the chapter's structure. Chapter (4.2) Research Activities, outlines the specific activities undertaken during the research process, detailing how the study was conducted and the steps involved in data collection. This is followed by Chapter (4.3) Data Generation and Analysis, which discusses the methods and processes used to generate and analyse the data. It explains the thematic analysis approach, highlighting how the data was systematically examined to identify key themes. Chapter (4.4) Characteristics of Research Participants presents a demographic and background profile of the study participants. Finally, Chapter (4.5) Findings of the Study, presents the main findings from the data analysis, the findings are discussed in relation to the research questions and the literature reviewed in Chapter Two.

4.2 Research Activities

The researcher undertook the following research activities in the implementation of this interventionist study.

Phase 1: To determine the participants' computer literacy, which could have an impact on the development of CT, the researcher administered a short questionnaire (see Appendix A4). The questions on this questionnaire focused on the participants' knowledge and ability to use computers and related technologies. The data gathered from this instrument helped to answer the second research question of this study.

Phase 2: Secondly, the researcher administered the CT Reflective tool (see Appendix A1). The participants worked on the activities in sixteen groups with a maximum of four members per group. Most groups managed to complete all the activities in two consecutive sessions. The purpose of this tool was to evaluate and facilitate the development of CT concepts and skills. The data gathered during this phase helped to address the first and second research questions of this study.

Phase 3: Following the CT reflective activities, pre-service teachers participated in coding sessions underpinned by the "use, modify, create" pedagogical framework. In the first session, the researcher articulated the CT concepts and skills, demonstrating how they are applied in everyday life. The subsequent coding sessions involved hands-on visual programming activities using Scratch. Using the web-based Scratch interface, participants were able to use already made programs, modify, and even create their own programs. The Scratch platform presented an array of digital characters called *sprites* that participants used to create animations, games, stories, music, and art. Scratch visual blocks were used to create *scripts* that issued specific computer instructions. The content that was covered in these sessions was guided by the analytical framework adopted in this study (Brennan & Resnick, 2012). Consequently, the researcher focused on CT concepts such as sequences, loops, and conditionals as presented in the CT framework. Figure 8 is a screenshot of the first program that participants created. In the program, the 'move' (motion) visual block makes the digital character '*sprite1*' move 10 steps. The motion block is inside a '*repeat'* loop whose function is to make the sprite repeat the action of moving 10 steps, two times. The last block on the bottom is an instruction that makes the sprite say the word "Hello". It is important to note that the execution of these blocks is based on the sequence in which they are stacked.

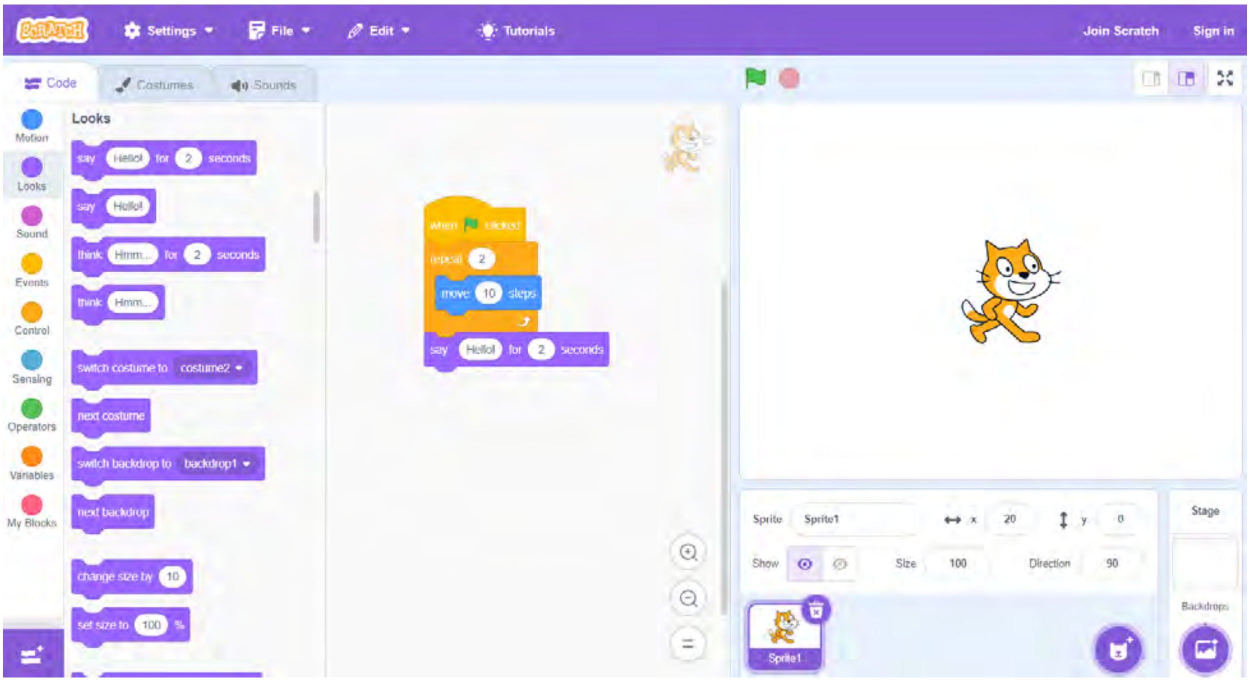


Figure 8: A Scratch visual program that uses 'sequences' and 'loop' CT concepts.

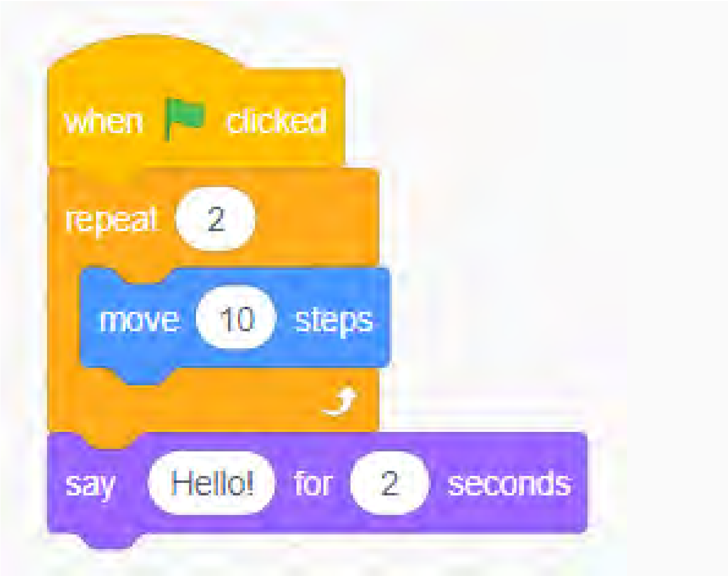


Figure 9: Extracted script from Figure 8 for clarity.

After each session, both the pre-service teachers and the researcher wrote reflections on their experiences. The coding sessions were conducted in a period of four weeks, and eight sessions were covered. Data generated in this phase helped to answer the first, and second research questions.

Phase 4: After the coding sessions, the same CT reflective tool that was administered at the beginning of the intervention, was re-administered. This was done to enhance and identify any potential shifts in the pre-service teachers' CT as a result of the intervention, thereby determining if learning had taken place within the zone of proximal development (Vygotksy, 1978). To ensure consistency and validity, the activities in the tool and the allocated time were the same.

This phase was followed by a comprehensive discussion of each activity in the reflective tool. The purpose of this discussion was to help participants who still had challenges with certain activities understand them, thereby enhancing their CT. The data collected during this phase was intended to answer the first and second research questions of this study. Table 2 shows an outline of the CT session activities throughout the 6-week period of the intervention.

Table 2: Computational Thinking Sessions Outline

Week No.	Activity	Content
Week 1	- Administering CT Reflective Tool	- CT Reflective activities (group work)
Week 2	- Introducing CT concepts and skills and their application in real-life. - Introducing Scratch platform - Exploring Visual Programs on Scratch Web.	- CT concepts and skills - CT Rationale - Scratch Interface (sprites, visual block, etc.) - TIPP & SEE - Sequences
Week 3	- Exploring games, music, art, stories, and animations. - (Use stage)	- Loops - Conditionals - Variables - Events
Week 4	- Modifying Scratch programs (games, music, art, stories, and animations). (Modify stage)	- Sequences, loops, conditionals, variables, events, costumes, sound
Week 5	- Creating visual programs (games, music, art, stories, and animations). (Create stage)	- Group work
Week 6	- Administering CT reflective Tool - Discussing CT reflective activities	- CT reflective activities (group work)

Phase 5: After the intervention, 10 semi-structured interviews and 2 focus group discussions were conducted. Willing and regular pre-service teachers were interviewed and participated in the focus group discussions. The data generated from this phase was intended to answer research questions one, and two of this study.

4.3 Data Generation and Analysis

The data used in this study were gathered from CT reflective tool, semi-structured interviews, focus group discussions and reflective journals. Triangulation of the data collection methods was employed to enhance validity and reliably and address the research questions of this study. Microsoft Office Excel was used to compute basic descriptive statistics such as average scores, frequencies, and plot histogram charts on the quantitative data (scores) from the CT reflective activities. For the rest of the data (qualitative), the researcher made some minor edits especially on the audio transcriptions for clarity. The edits included correction of grammatical errors, removal of long pauses, deletion of filler sounds such as "like", "okay", "um", and "eh", and deletion of repeated phrases. In scenarios where institutional or individual names were mentioned, pseudo names were used instead to ensure anonymity. The data were analysed through the lenses of the research questions, which focus on:

i. The development of foundational phase pre-service teachers' CT with visual programming.
ii. Factors that aid or hinder foundational phase pre-service teachers in developing CT with visual programming.

4.4 Characteristics of the research participants

A total of forty-nine (49) foundational phase first-year pre-service teachers participated in this study. Out of the forty-nine participants, only eight studied ICT or computer related subject in secondary school. However, forty-seven pre-service teachers indicated that they own a smartphone, and they all use social media platforms such as WhatsApp and Facebook. Sixteen participants (33%) owned a laptop or desktop computer. Table 3 shows the participants' computer literacy skills set.

Table 3: Participants computer literacy skills set.

Skill	Number of Competent Participants	Percentage of Competent Participants
Typing on a computer keyboard	45	92 %
Playing videos on a computer	36	73 %
Playing games on a computer	26	53 %
Using Microsoft Office Word	49	100 %
Using Microsoft Office PowerPoint	45	92 %
Using Microsoft Office Excel	46	94%

Programming	3	6%

Despite most participants not owning a laptop or desktop computer, only eight having studied ICT in secondary school, and only three having prior programming experience, the majority were computer literate. Based on the definition of computer literacy given by Safar & AlKhezzi (2013), as the knowledge and ability to use computers and related technology or applications, it can be concluded that the majority of the participants possessed basic computer literacy skills. This is important because basic computer literacy skills are a pre-requisite for visual programming (Davis, 2011), making the implementation of this study feasible. For example, participants needed to know how to use a computer mouse and type on a keyboard to be able to drag and drop visual blocks to create visual programs. Furthermore, having prior knowledge in computer literacy focused the study on the use of visual programming frameworks and not on computer literacy.

4.5 Findings for the Research Questions

4.5.1 The development of computational thinking through visual programming.

According to Brennan & Resnick ’s CT framework, which was adopted as the analytical framework for this study, in the context of a visual programming approach, an individual is considered to have developed CT when they demonstrate understanding and application of CT concepts and practices (Brennan & Resnick, 2012). Therefore, before addressing the “how” question, the researcher decided to evaluate the participants’ computational thinking based on the two CT components (CT concepts and practices). The assessment was administered in the form of CT reflective activities, before and after the intervention. Moreover, from the sociocultural theory’s perspective, it was necessary to evaluate the participants’ CT to determine if learning had taken place in the ZPD.

The use of Scratch visual programming enhances the development of CT

The data collected from this study showed that visual programming approach fosters the development of CT among pre-service teachers. For instance, the first section of the CT reflective tool contained an activity that required the participants to share their perceptions or understanding of CT. Table 4 shows some of the CT definitions from six groups that attempted the activity.

Table 4: Extracted from CT Reflective Tool, Section B, Q.1: Appendix A1 Pre-service Teachers' Understanding of CT before Intervention

Group Name	CT definition
Group A	*"It is a programmed way of thinking".*
Group B	*"It is the knowledge about the structural components of a computer, and it also means functioning like computers".*
Group C	*"It means knowing how computers function and thinking like computers".*
Group D	*"The ability to use computer technology".*
Group F	*"It is the ability to think in a programmed way".*
Group E	*"Our way of thinking that relates to how a computer actually functions".*

From the definitions given, it shows that the pre-service teachers perceived CT as a type of thinking that resembles computers or as simply computer knowledge. Table 5 shows the pre-service teachers' perceptions of CT after the intervention.

Table 5: Extracted from CT Reflective Tool, Section B, Q.1: Appendix A1 Pre-service Teachers' Understanding of CT After Intervention

Group Name	CT definition
Group A	"We *understand that computational thinking has to do with analyzing a problem and finding a solution for it*".
Group B	"*It is a process where we decompose a problem, recognize the pattern, abstract it, and use algorithmic thinking to solve it*".
Group C	"*Computational thinking is a way of thinking or solving problems like computer scientists. It involves breaking down complex problems into smaller and simpler parts, finding patterns, using logic, and creating solutions that can be understood by humans or computers*".
Group D	"*We understand that it is the use of computer skills to deal with everyday problems*".
Group E	"*It is a process which allows people to acquire problem-solving techniques using computer knowledge in order to deal with daily problems*".
Group F	"We *understand computational thinking as a problem-solving approach that involves breaking down complex problems into smaller,*

	more manageable parts and solving them using concepts and methods from computer science. It is essentially a way of thinking and analyzing problems that draws from the principles of computer science. And we also understand that it has several key elements such as decomposition, abstraction, algorithm e.t.c. we also understood that it's not limited to computer science but can be useful in various fields".

Comparing the participants' perceptions before and after the intervention, it is evident that the pre-service teachers' understanding of CT had shifted from perceiving CT as thinking like computers or as computer knowledge to recognizing CT as a problem-solving skill set drawn from computer science principles.

Figure 11 is a histogram that shows the scores of all sixteen groups on the four CT reflective activities in the second section, before and after the intervention. This section contained CT activities that were designed to evaluate the participants' understanding of CT concepts such as sequences, loops, and conditionals, independent of a visual programming platform. Figure 10 is an example of one of the CT activities in Section C of the tool. It requires the application of CT concepts of sequences and conditionals to solve it. In this case, option "D" is the correct sequence that takes the car to the filling station. As observed in this activity, no prior syntactical knowledge of a specific programming language is required to solve it.

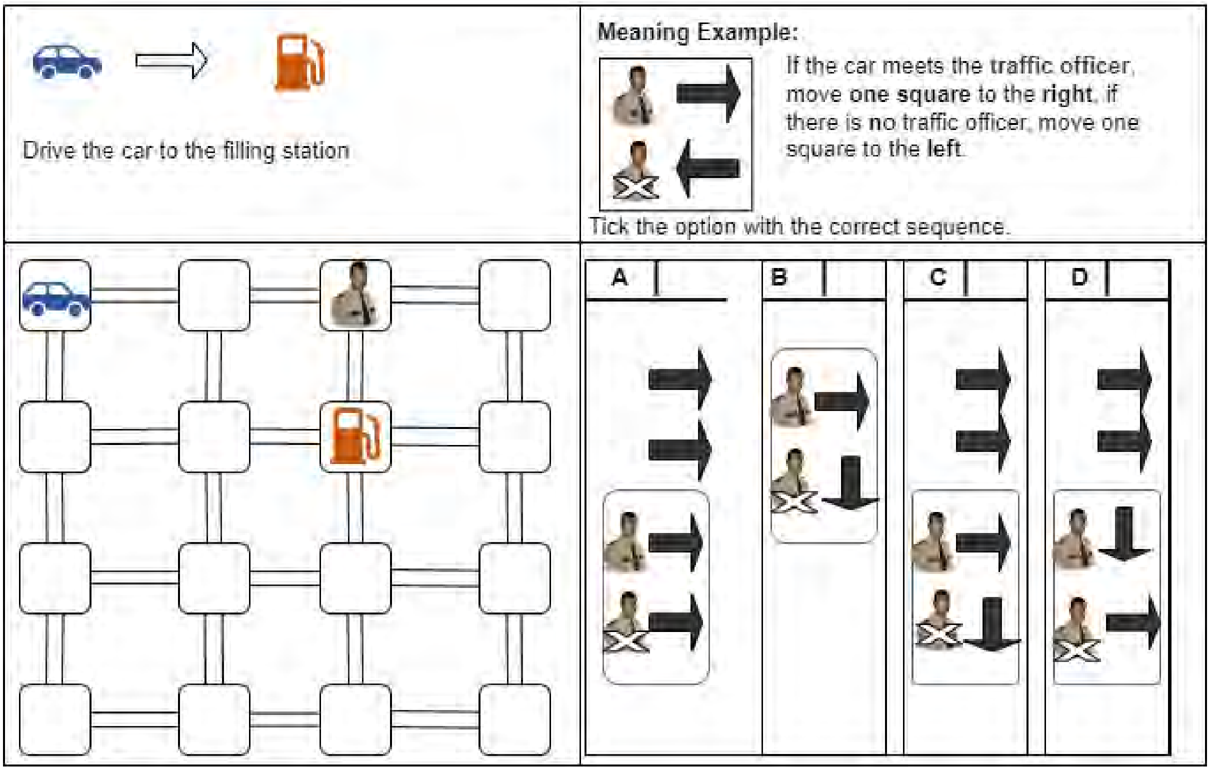


Figure 10: CT Concepts Activity on Sequences and Conditionals

The average score for these activities was 2.3 out of the total score of 4 (1 score for each activity). However, after the intervention, the participants' scores for the same set of CT activities improved as shown in Figure 11. The average score increased from 2.3 to 3.4. Unlike before the intervention, in which most groups (9) scored between 0 and 2, after the intervention, almost all groups (15) scored between 3 and 4 in this section.

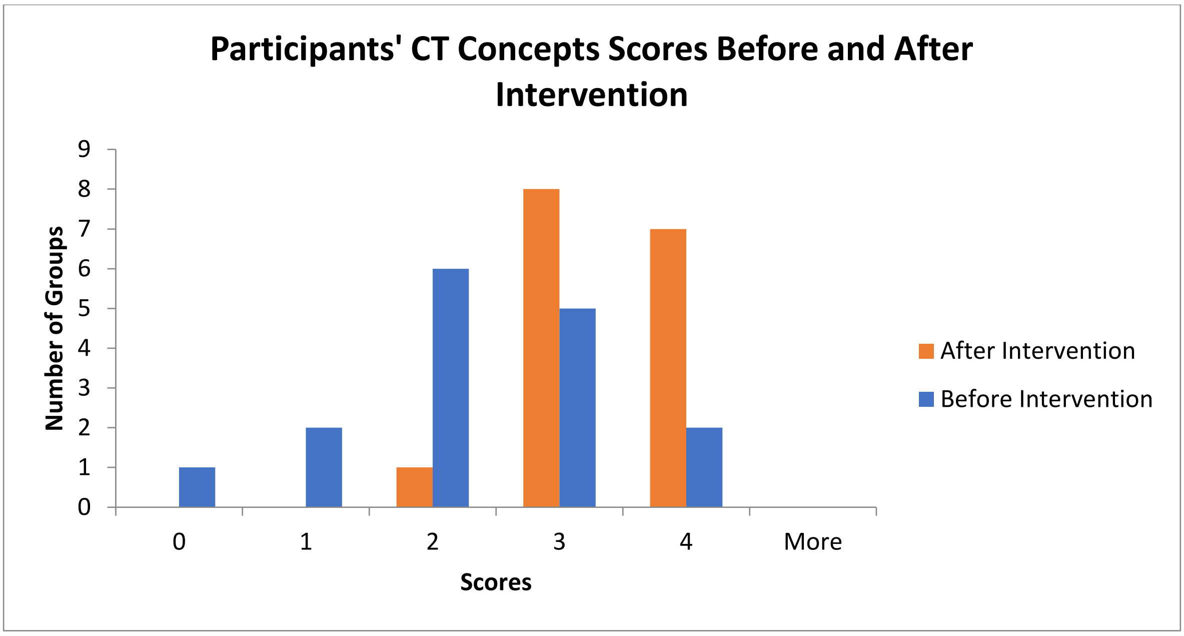


Figure 11: CT Participants' Scores on CT Concepts in Section C

In Section D, participants were asked to solve five CT activities based on the same CT concepts applied in Section C, but dependent on visual programming knowledge. Participants needed to observe visual programming logic to solve the activities. Figure 12 is an extract of one of the activities in this section, assessing participants' understanding of sequences and loops in the context of Scratch. In this activity, option (a) is the correct script, as the repeat loops and the motion blocks are correctly sequenced and take the car to the filling station along the marked-out path.

Activity 2

Which instructions take the car (**B6**) to the filling station (**B4**) using the path marked out?

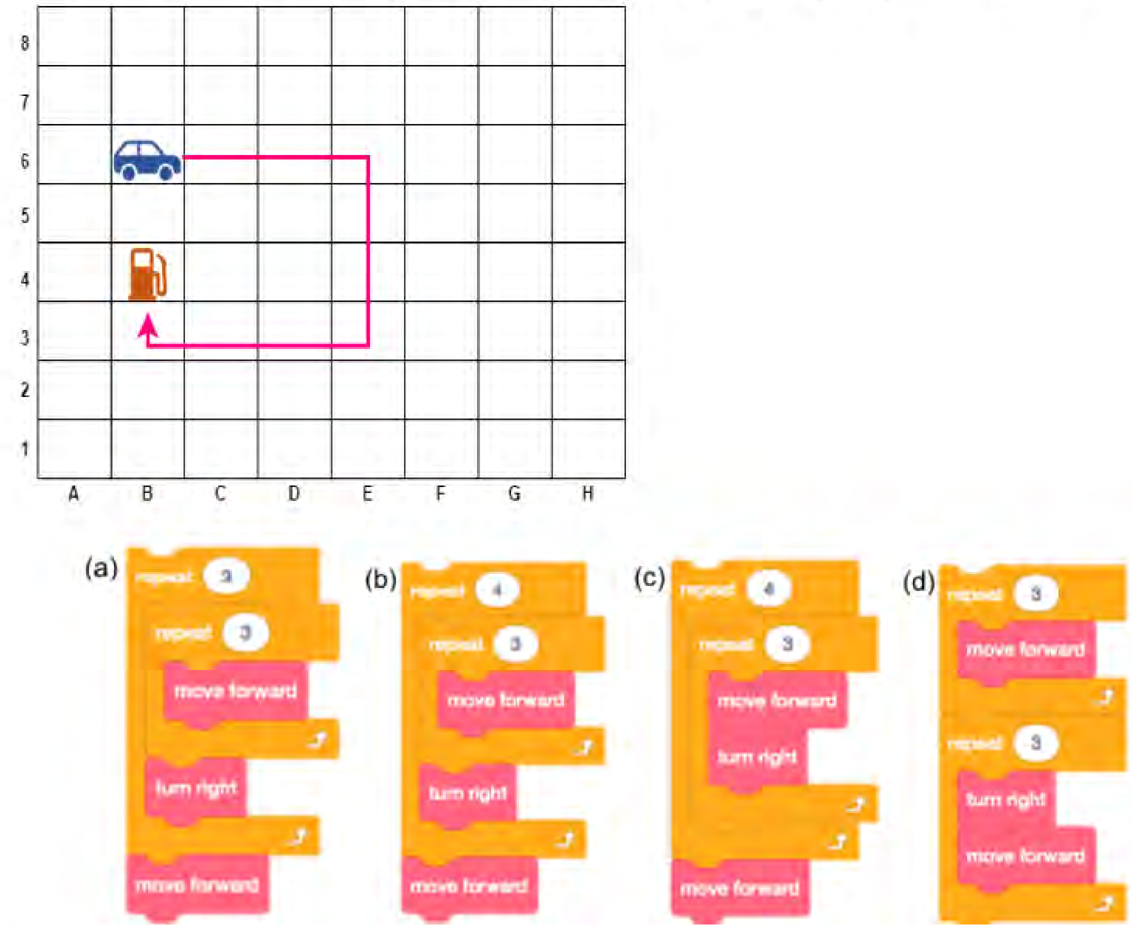


Figure 12: A CT Concepts Activity on Sequences and Loops in Scratch

Figure 13 displays group scores for section D, before and after the intervention. The average score for the sixteen groups before the intervention was 1.5 out of the total score of 5 (1 score for each activity). As observed from the chart, most of the groups scored between 0 and 2. Eight groups managed to solve only one activity correctly. After the intervention, there was a significant improvement, average score increased from 1.5 to 3. Despite this shift, the scores in this set of activities (section D) were relatively low compared to the first set of CT concept activities (section C). This is indicative of the programming challenges that are still present among the pre-service teachers even after the intervention.

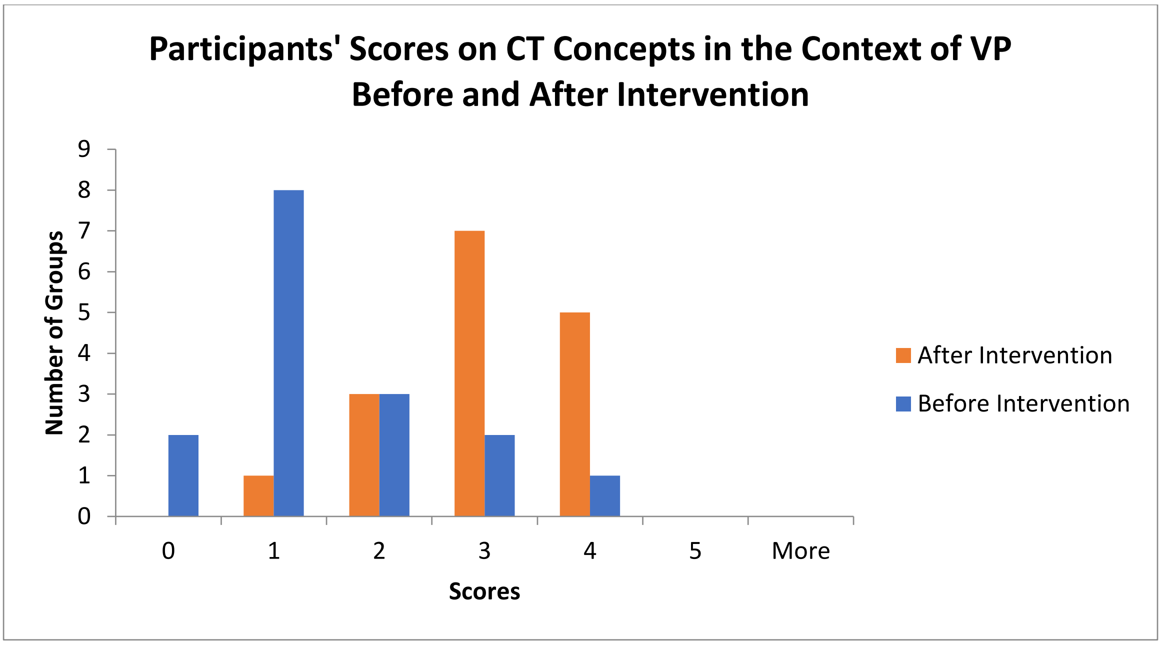


Figure 13: Participants' Scores on CT Concepts in Section D

The last section of the CT reflective tool consisted of activities that focused on the application of CT concepts and practices beyond computer science setting. Figure 14 shows scores of the participants for section E, before and after the intervention. As shown in the chart, participants' performance improved significantly. Before the intervention, ten groups scored between 0 to 1 but after the intervention, twelve groups managed to score between 3 and 4 out of the total score of 5. The average score increased from 0.9 before the intervention to 3.6 after the intervention.

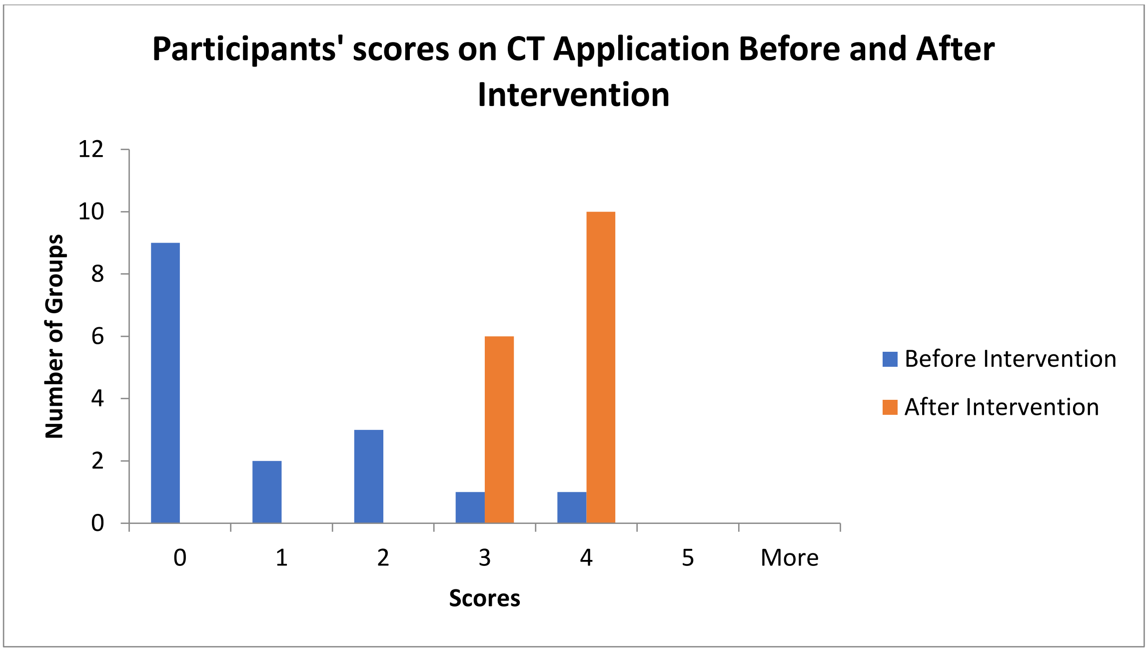


Figure 14: Participants' Scores on CT Application in Section E

Figure 15 is a histogram showing distribution of pre-service teachers' combined group scores for all activities in three sections. The total score was 13 for thirteen activities, 1 score for each activity.

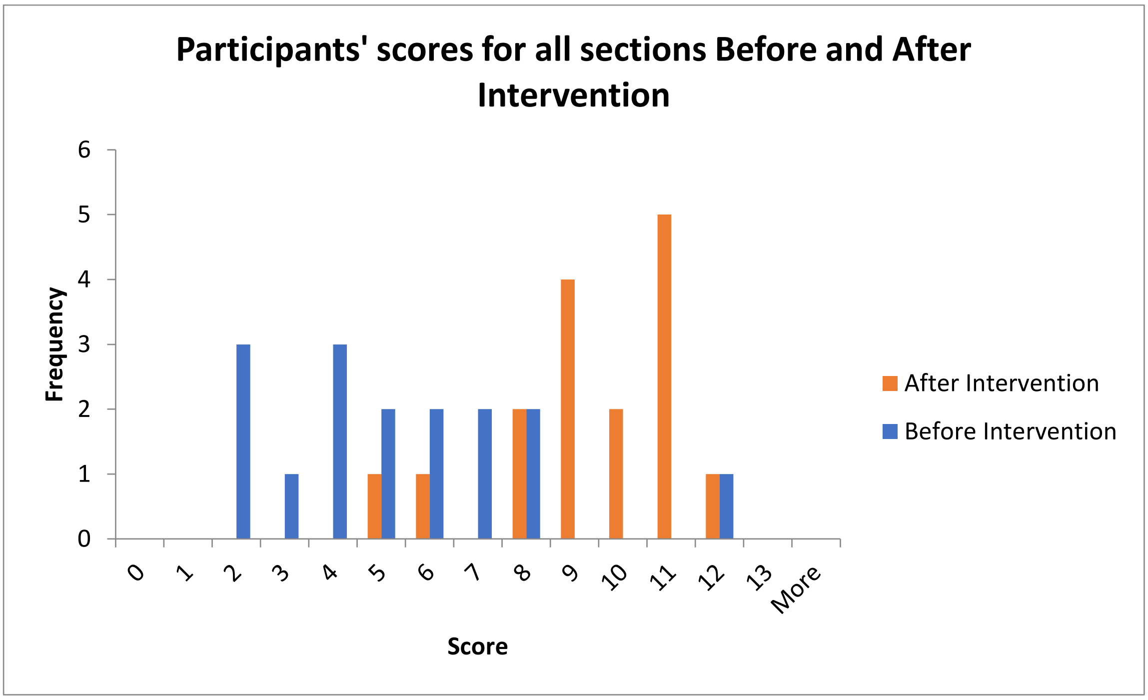


Figure 15: Participants' Scores in all Three Sectons

Overall, the pre-service teachers' scores improved significantly. The average score increased from 5.3 before intervention, to 9.4 after intervention.

During the semi-structured interviews, all the interviewees felt that the coding sessions they conducted fostered their computational thinking skills. For example, Mpho answered the question of how visual programming contributed to their development of CT as follows:

> *"Visual programming contributed to the development of my computational thinking. For instance, if you put a motion block inside and outside a repeat loop, the program behaves differently. If you place it incorrectly in your script, you need to identify the problem and understand why it's not working as expected, then fix it. So, I think this is a perfect way to demonstrate the problem-solving aspect of computational thinking".*

Bandile shared similar perception, answering the same question as follows:

> *"Visual programming contributed to my understanding of computational thinking by revealing the process behind creation of computer games and stories. I learnt that there is a lot that goes into the programming of the computer games we see. For me, it is the thinking of how to do it that is important".*

Some of the pre-service teachers were of the view that the visual programming sessions they conducted had a cognitive effect on them. It enhanced their problem-solving skills. Elaborating further, Mpho had the following to say:

> *"I think I can even apply what I learnt in visual programming in my everyday life, not just in technology. For example, it's easy to solve a problem when you recognise its pattern, like remembering how you solved a similar problem in the past".*

Amahle wrote the following in their reflective journal:

> *"I feel this session was very useful as I learnt a lot of important skills that will benefit me in the future".*

These views suggest that the participants thought that the skills they gained in the coding sessions would be applicable in their future lessons and life situations, beyond computer programming.

Thematic analysis of the data from the semi-structured interviews, focus group discussions and reflective journals showed that visual programming enhances the development of CT by fostering three main themes:

Table 6: Visual Programming Elements that Enhance the Development of CT

Theme	Description
1	Interest
2	Creativity
3	Collaboration

Visual programming fosters interest

Most pre-service teachers found Scratch programming fascinating and fun. From the first coding session, it was evident that the participants were captivated by the platform. I observed that most pre-service teachers were excited both among themselves and to me as researcher during the coding sessions. Moreover, at the end of each Scratch programming session, most participants chose to remain in the computer laboratory to continue exploring the platform. In a semi-structured interview, Ayanda and Thando described their experience of engaging with the platform as follows:

> *"I enjoyed Scratch programming because I have always been interested in playing computer games since I was a child. It was amazing to learn how they are made especially how the scores are stored and incremented".*

Thando commented:

> "*I found coding in Scratch both exciting and enlightening. It was fascinating to create a game and see a digital character behave based on the instructions I had written*".

Another participant, Bonolo shared their experience of Scratch during the semi-structured interviews as follows,

> *"The Scratch platform was very fascinating and since children are highly interested in technology, it's a perfect tool for teaching computational thinking. It will allow us to teach important life skills in a fun and engaging manner without the children even realising it."*

The researcher observed that there was a consensus among the participants that the graphic, engaging and interactive nature of Scratch cultivated their intrinsic motivation. Participants were always looking forward to the CT sessions to explore, modify and create their own visual programs in Scratch.

Visual programming fosters creativity

Most participants described visual programming as creative, allowing them to design and create programs of their liking. Bongani shared their experience with visual programming as follows:

> *"I liked it because it ignited my creativity. I was able to play around with it and create my own things".*

Figure 16 is a visual program that Bongani created during the coding sessions, an animation program that animated her name. The extracted scripts in Figure 15 were designed to change colour, size and spin the first three letters respectively.

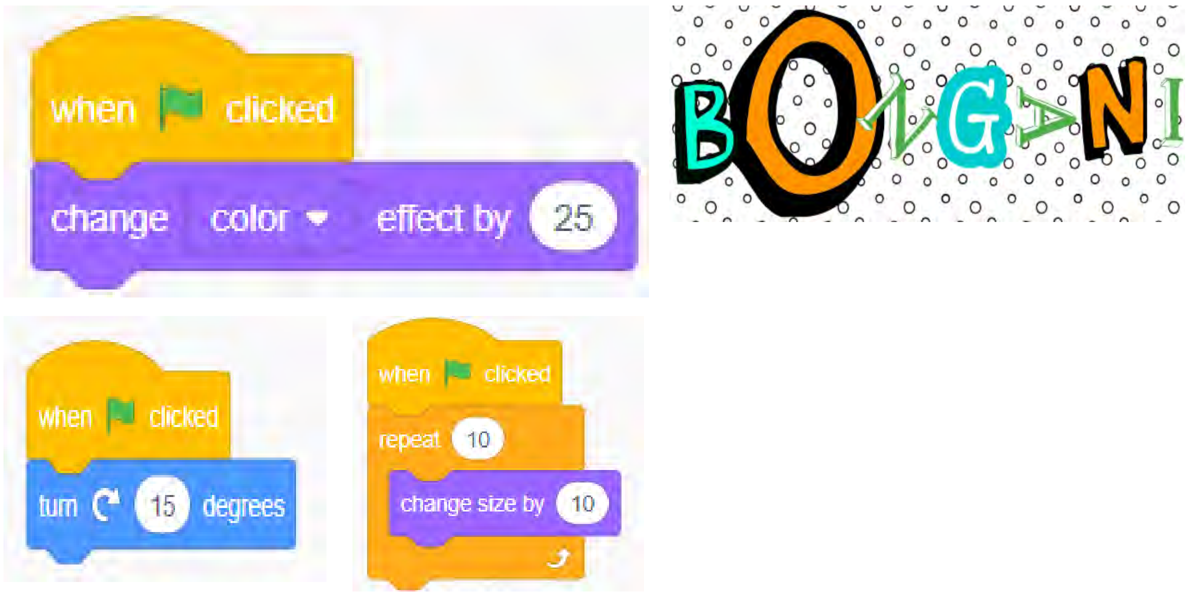


Figure 16: Participant's visual program of animated name

Another participant, Babalwa had the following to say about her experiences with visual programming:

> *"I liked the aspect of creating my own programs because it allowed me to design visual programs that are relevant to me. This also allowed me to express myself through the programs I created".*

Several participants echoed similar views, highlighting that the creativity aspect of the visual programming platform allowed them to express themselves. Arno, one of the participants from group C created the following program (See Figure 17) when they were asked to create a moving digital character during one of the coding sessions.

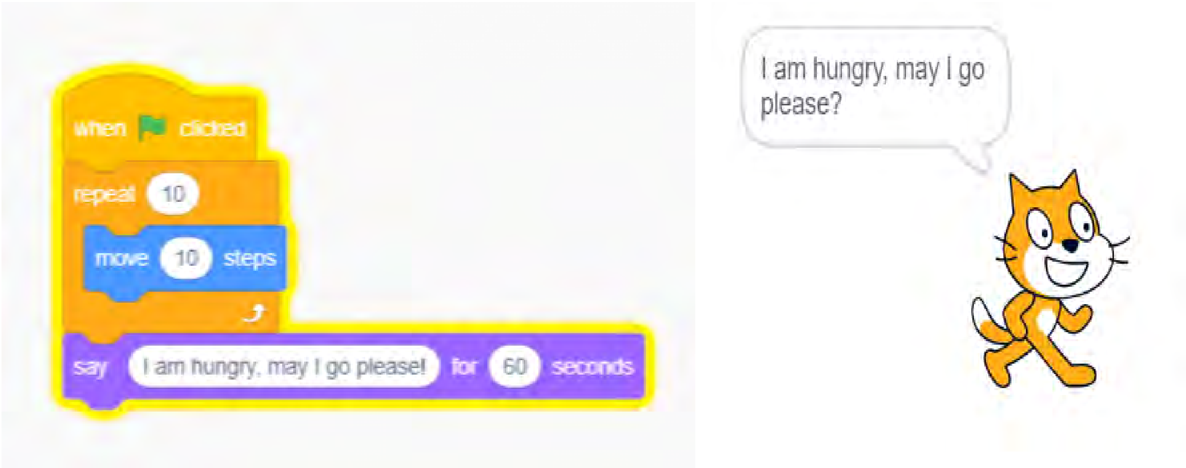


Figure17: Participant's Visual Program

Arno created a visual program where a character moves ten steps and says, "*I am hungry, may I go please?*". When the researcher inquired about the program, it was discovered that the participant was expressing their own feelings through the program. The participant was hungry, and she was indirectly asking the researcher to let her go.

Visual programming fosters collaboration

The pre-service teachers were all tasked to work in pairs and groups throughout the coding sessions. Most participants perceived the sharing of ideas and solving problems collectively as a key aspect that enhanced their understanding of CT. Thando noted in her reflective journal as follows:

> *"Today's session was a bit challenging to me as I couldn't properly implement the conditional statement for the Pong game. However, with the help of the facilitator and my friend, I managed to do it. If I struggle with anything else in the subsequent sessions, I plan to seek help from my friends instead of trying to figure it out on my own and wasting time.".*

Other participants found the collaborative environment of Scratch helpful in their understanding of CT. Cebisa commented:

> *"I liked Scratch because I was able to see how other people implemented their programs across the world. I learnt how to contribute and learn from the online Scratch community".*

These themes are consistent with what Martin et al. (2014) discovered about visual programming. They called it 'creative computing', which they defined as an approach that enhances self-expression, imagination, and personal interest areas.

4.5.2 Factors that aid foundational phase pre-service teachers in developing computational thinking with visual programming

Understanding factors that facilitate the development of CT among pre-service teachers using visual programming was part of the second objective of this study. To address this objective, the researcher phrased a question: What are the enabling and constraining factors that affect the development of computational thinking using visual programming approach among pre-service teachers? (Question 2). Data from the CT Reflective tool, semi-structured interviews, focus-group discussions and reflective journals were analysed to answer the question. The analysis of the data led to the following pedagogical and technological themes which are presented in Table 7.

Table: 7: Factors that aid pre-service teachers in developing CT using visual programming

Theme	Description
1	Contextualized project-based learning facilitates CT development
2	A clear CT rationale facilitates its development
3	Prior-programming experience enhances the development of CT

Contextualised project-based pedagogical method facilitates CT development

Almost all the participants were of the view that the project-based pedagogical approach enabled them to master the visual programming CT concepts. In a similar study, Valls Pou et al. (2022) discovered that Project-Based Learning (PBL) is an effective active learning method for the development of CT among pre-service teachers. In the study, participants were divided into pairs and groups to create contextualized visual programs using Scratch. Findings revealed that PBL is an effective pedagogical approach as it is student-centred and involves creating artifacts that have real-life connections and applications. This also aligns with the findings of Stefania et al. (2022) who conducted a systematic review on the pedagogical approaches that various CT practitioners employ in CT development. They discovered that project-based approach is commonly used in contexts where a programming-oriented CT development approach is adopted, as it engages students in motivating activities. In this study, the following views shared during the semi-structured interviews from Bonolo, Bongani and Mpho are examples of evidence that project-based approach enables the development of CT:

Bonolo commented:

> *"I was thinking that it would be more beneficial if you gave us more assignments where we could develop our own games. This would allow us to practice more, as most of the CT concepts seemed easy in class but proved more difficult when we tried to apply them on our own."*

Bongani shared similar observation and added that contextualised projects enabled them to create programs that were relevant to them as pre-service teachers:

> *"I think what helped me to understand the concepts the most was the opportunity you gave us to create our own programs. We were able to design games that are relevant to us teachers and not just relevant to kids, like Pong Game we practised in class. And when you teach games in class choose games that teachers can also enjoy."*

Mpho echoed the same perspective as follows:

> "*I think the best way to teach this content is how my high school teacher taught us. She gave us tasks to develop our own projects and related the content to a lot of personal occurrences in our lives. In the same way I think the best way to teach CT to pre-service teachers is to get to their level and use Scratch program examples that relate to their everyday life."*

These views were widely shared among the participants. They felt that providing more opportunities to create their own visual programs would significantly enhance their acquisition of CT skills. Furthermore, most pre-service teachers acknowledged that CT skills are more practical and therefore, require hands-on pedagogical approaches for effective development.

Providing a clear CT rationale facilitates its development

Most of the pre-service teachers cited the rationale behind CT as one of the major factors that motivated them to develop CT. This finding resonates with existing literature. For instance, Angeli & Giannakos (2020) found that emphasizing the role of CT in daily activities made the concepts more relatable and easier to understand for students. Similarly, Malčík et al. (2023) revealed the same finding where they discovered that providing a comprehensive CT rationale helped pre-service teachers to see the practical applications of CT which enhanced their motivation to learn and apply the skills. In this study, during semi-structured interviews, Adem commented:

"I was very confused, and I didn't understand what we were doing when you administered the CT reflective activities at the beginning. But when you explained the importance of CT and how it is applied in various fields beyond computer science setting, I changed my attitude, I started liking it and I was able to understand."

During a focus group discussion, participants from FGD1 had the following to say about factors that facilitated their CT development:

"We believe that understanding the importance and application of computational thinking in our everyday lives is what got our attention the most. It helped us realize that CT is not just about computing; it can be used to solve everyday problems, and it can even enhance our learning."

Concurring with these views, Bongani shared the following during semi-structured interviews:

"I think knowing how CT will benefit my future learners motivated me to learn and ignited my interest in developing CT skills."

The researcher also made similar observations during the CT sessions where most participants looked uncomfortable and confused with the content at first when the CT reflective activities were administered. In the following CT session where the CT rationale was given, explaining the importance and application of CT skills to everyday life, most pre-service teachers were keen to learn. Their disposition clearly shifted from perceiving CT as the unrelatable skill to a skill they can apply in their daily problem-solving situations.

Prior-programming experience enhances CT development

The data collected from the CT reflective activities, semi-structured interviews, focus group discussions and the reflective journals, demonstrate that prior-programming experience help pre-service teachers to understand CT better. For example, Group K, which comprised of the three pre-service teachers who indicated that they had prior-programming experience during their secondary school education performed better in CT reflective activities than all the other groups. Table 8 shows the scores for Group K before and after the intervention, across the three sections.

Table 8: CT Reflective Activities Scores for Group K

Section	Before Intervention		After Intervention		Total Section Score
	Group 11 score	Average score	Group K score	Average score	
B	4	2.3	4	3.4	4
C	4	1.5	4	3	5
D	4	0.9	5	3.6	5

These pre-service teachers took one subject of IT during secondary school education. According to the South African education curriculum, these IT subjects are designed to provide a robust foundation in understanding and utilizing the integrated components of a computer system, shaping students into digitally literate individuals (Basic Education, 2011). The curriculum emphasizes practical techniques to efficiently apply IT knowledge in solving everyday problems. Furthermore, the IT curriculum is also intended to lay some foundation for computer programming where students are introduced to basic programming concepts using programming languages like HTML.

When probed if there were any challenges during the sessions, one of the participants from the group, Mpho had the following to say:

> *"I didn't find anything particularly difficult. It was super fun and very interesting. We coded in HTML in high school, so even though some aspects of Scratch are different, I noticed that the underlying concepts are the same."*

Moreover, the researcher observed that pre-service teachers who owned a laptop also demonstrated a slightly better understanding as compared to the pre-service teachers who did not own a laptop. These findings are concordant with the existing literature, for instance, Kılıç et al. (2021) conducted a study in Turkey to determine CT skills of first year university students who were going through a visual programming course. It was observed that CT skills of students who had low, middle, and high-level prior programming experience differed significantly. Students with high-level prior programming experience demonstrating high levels of CT skills than those with low prior programming skills. In a similar study, Cruz Castro et al. (2021) analysed CT practices in a first year engineering course. The goal of the study was to identify the differences in students' CT practices at the beginning of an undergraduate introductory programming course and to understand how these differences correspond to acquisition of more advanced CT practices. The results showed that previous knowledge or exposure to different IT practices could result in different progression of more complex CT practices, emphasizing the relevance of prior programming experience for a successful and efficient progression of CT skills.

4.5.3 Factors that hinder foundational phase pre-service teachers in developing computational thinking with virtual programming

The second part of the objective of this study was to uncover factors that may hinder the development of CT in the adopted approach of virtual programming among foundational phase pre-service teachers. Using data from the researcher's observations, CT reflective tool, semi-structured interviews, focus-group discussions, and reflective journals, the researchers found the following themes.

Table 9: Factors that hinder pre-service teachers in developing CT using visual programming

Theme	Description
1	Lack of a clear rationale affects CT development
2	Lack of prior programming experience impedes CT development
3	The multimedia nature of Scratch hinders CT development

Lack of clear CT rationale affects CT development

Based on the data from semi-structured interviews, reflective journals, and the researcher's observations, it became clear that failure to provide a comprehensive rationale for CT first, negatively affects CT development among pre-service teachers. Although a well-articulated CT rationale positively affects CT development, its absence has a negative effect. In this study, some participants who missed the first sessions where the researcher provided the CT rationale and highlighted the real-world applications of CT, had challenges understanding the content. For example, Arno wrote the following in her reflective journal:

"I don't enjoy and understand everything we have covered in the sessions so far because I fail to understand the main reason for the new content. Maybe it would be helpful if the sessions included explanations of the underlying reasons and objectives."

Adem made the following similar suggestion in their reflective journal:

"I think it would be more appealing and engaging if the sessions clearly explained how learning this will benefit us. Some of us don't yet see the significance of computational thinking, so highlighting its real-world applications and advantages could make the content more relevant and motivating."

This finding resonates with what Angeli & Giannakos (2020) discovered about the role of CT rationale in the acquisition CT practices among students. They found that failure to understand the role of CT in daily activities made the CT concepts unrelatable and difficult to understand.

Lack of prior programming experience affects CT development

Another major factor that impedes CT development as observed by the participants and the researcher, is a lack of prior-programming experience. Similar findings were discovered by Li (2021), who investigated the challenges associated with CT education. The scholar discovered that most teachers who lacked prior programming background struggled to understand CT. In this study, this phenomenon was also confirmed, most participants who had no prior programming skills had challenges understanding the CT concepts. During one of the coding sessions, Buhle, made the following comment which the researcher noted in his reflective journal:

"This content is just making our lives difficult. The last semester content was easy. May be if we had done something similar in high school, we could understand it better."

One focus group discussion made the following observations:

"Scratch programming is challenging, especially for those of us who never studied ICT in high school. The concepts and logic required are new and can be difficult to understand without prior exposure. We believe that our experience would have been significantly different if we had done similar programming tasks in high school. Having a foundational understanding would have made the learning curve less steep and more manageable."

The general view among the pre-service teachers was that even though visual programming is user friendly and relatively easy to understand, lack of previous exposure to programming concepts made the learning curve or the transition too steep. Most participants failed to assimilate or relate the CT concepts to anything they had learnt previously.

The multimedia nature of Scratch hinders the pre-service teachers' understanding of the underlying CT concepts

The data analysed in this study shows that the multimedia nature of Scratch (Bers et al., 2014) diverts the participants' attention from the underlying concepts. Participants appeared more captivated by the Scratch graphics, which led them to overlook the underlying concepts they needed to understand. For example, FGD2 had the following to say when they were asked about the challenges they faced during the intervention:

> *"When you first introduced the Scratch web interface, it looked fascinating and easy. We got so excited and motivated to create our own games and animations. However, when we attempted to create the first Scratch program, we realised that it was not as easy as it seemed. And we knew that our initial attitude had costed us the underlying principles we needed to understand in order to create our own visual programs."*

Bonolo, shared the following about visual programming:

> *"Visual programming is like mathematics; it is exciting when you know how to do it but very frustrating when you don't understand it. At first glance, it looks like it's just a matter of moving a visual block from one place to another, it looks too easy, and because of that attitude, you tend to miss the underlying principles. Then, when you try to do it, that's when you realise that it is not as easy as it seemed."*

Similar findings were found by Mayer & Moreno (2003) who investigated the impact of graphical programming environments on CT development. They discovered that multimedia or pictorial interventions may not always have the desired effect on participants' CT conceptual development. Some learners in the study could not build new knowledge beyond the pictorial information.

Conclusion

This chapter presented research findings from the collected data from forty-nine research participants. Data collected from CT reflective tool, semi-structured interviews, focus-group discussions, reflective journals and the researcher's observations during the study were at the centre of this chapter. The presentation of the findings was guided by the two research questions, which focused on three areas: (1) the development of CT among pre-service teachers using visual programming, (2) factors that aid the development of CT in visual programming approach among foundational phase pre-service teachers, and (3) factors that hinder the development of CT in visual programming approach among foundational phase pre-service teachers. The chapter correctly presented the results according to the themes that emerged from the analysis of the data, guided by the research questions. In summary, findings of this study showed that visual programming approach enhances CT development by fostering participants' interest, creativity and collaboration. Project-based pedagogical approach, CT rationale, and prior programming knowledge were found to be facilitators of CT development. Finally, the lack of a clear CT rationale, prior programming experience, and the multimedia nature of Scratch were identified as factors that hinder CT development in visual programming approach among foundational phase pre-service teachers. The next chapter presents a discussion of these findings.

CHAPTER FIVE
DISCUSSION OF THE FINDINGS

5.1 Introduction

The previous chapter presented the findings of this study. This chapter provides a discussion of the findings. Discussion of research findings involves summarizing key findings of the study (Lim, 2010). This includes highlighting the major outcomes and any significant data points that were found. Secondly, discussion of research findings involves interpreting the results in the context of existing literature. This involves comparing the results with other similar studies to identify similarities and differences. As argued by Shabani & Lam (2013), comparing study findings across other studies helps to evaluate validity of the results, ensure reliability of results and enhance overall understanding of the research topic. Moreover, study comparisons help to mitigate biases that may be present in individual studies, which leads to more accurate and reliable conclusions (Junior et al., 2019). In this chapter, the discussion of findings provides insights and interpretation of the results based on the existing literature that was reviewed in Chapter Two. This chapter is divided into three parts: (5.1) an introduction, (5.2) discussion of key findings, and (5.3) a conclusion.

5.2 Discussion of Key Findings

Overall, the objective of the study was to explore the development of computational thinking among foundational phase pre-service teachers using visual programming approach. To achieve this objective, two research questions were phrased. Table 10 below presents a summary of the questions and the corresponding research instruments.

Table 10: Summary of Research Questions and the Corresponding Research Instruments

Research Question	Research Instrument
How do foundational phase pre-service teachers develop computation thinking with visual programming?	CT reflective tool, Semi-structured interviews, Reflective journals
How does an intervention that uses a visual programming approach aid or hinder foundational phase pre-service teachers in developing computational thinking?	CT Reflective Tool, Semi-structured Interviews, Focus group discussion, Reflective journals

The reported findings in Chapter Four are discussed based on the research questions summarised above.

5.2.1 The development of computational thinking through visual programming

The findings of this study indicate that the participants were able to experience the development of computational thinking through visual programming. As shown in the previous chapter, the majority of pre-service teachers believe that visual programming enhances development of CT. This perception is supported by their scores on CT reflective activities (see Section 4.5.1), which show a marked improvement across all three sections of the assessment tool. In Section B, the average score increased from 2.3 to 3.4; In Section C, from 1.5 to 3; and in the last section, from 0.9 to 3.6. The average score for all activities increased significantly from 5.3 to 9.4 out of total score of 13. These increases reflect significant progress in the participants' CT development. Additionally, there was a noticeable shift in the pre-service teachers' perception of CT; from viewing it merely as thinking like a computer or having computer knowledge, to recognizing it as a broader, transferable problem-solving skill set derived from computer science principles.

These findings are consistent with the findings of Peracaula-Bosch & González-Martínez (2022), İlic (2021), Cetin (2016), Fagerlund et al. (2020), Maraza-Quispe et al. (2021), Molina-Ayuso et al. (2022), Umutlu (2022), Yadav (2017), and Stephenson, et al. (2017). These scholars focused on investigating the extent at which visual programming approaches enhance CT development and reported considerable development of CT among pre-service teachers. For instance, Peracaula-Bosch & González-Martínez (2022) conducted a study at a teacher training institution in Spain to develop computational thinking among pre-service teachers using Scratch. The scholars proposed a CT training similar to this intervention where pre-service teachers were involved in Scratch programming activities to enhance their CT. A training involving 37 bachelor's degree in Teacher Training students was carried out in a second-year computational thinking and programming course in 12 face-to-face sessions. The computational thinking test validated by Román-González et al. (2017) was applied at the beginning and the end of the training. The CTt results showed a significant improvement of pre-service teachers' CT level from an average score of 19.03 to 22.41 on a scale of 30 points. Further statistical analysis of the differences between the pre-training and post-training CTt scores showed that the differences were statistically significant, confirming that the differences were due to the visual programming experience and not chance.

In a similar study, İlic (2021) investigated the effect of Scratch programming on the computational thinking skills of pre-service teachers in Instructional Technologies course in Turkey. Thirty-three pre-service teachers participated in their study and data were collected from computational thinking test before and after the training. Additionally, a questionnaire and face-to-face semi-structured interviews were conducted at the end of the course. Results indicated that the Scratch programming activities conducted in the study had improved the CT skills of the participants where the post-CTt scores were significantly higher than the pre-CTt scores. Furthermore, pre-service teachers had harmoniously stated that Scratch programming contributed to their development of CT skills. In another similar study, Cetin (2016) employed an experimental design to explore the effects of Scratch-based instruction on pre-service teachers' CT skills and found similar outcomes at a teacher training institution in Turkey. Two samples were used: experiment and control groups. The pre-service teachers' CT skills were evaluated using CT and aptitude tests at the end of the study. Results showed that there was a significant difference in the CTt scores between the control and experiment groups. The experiment group got a mean score of 57.18, while the control group got a mean score of 45.54 with a statistically significant value (P-value) of 0.001, confirming that the difference was due to the Scratch programming training.

The perceptions of pre-service teachers highlight the significant role that visual programming plays in the development of CT. According to the participants, visual programming enhances CT by fostering three key elements: interest, creativity, and collaboration. The engaging nature of visual programming environments like Scratch appears to be more influential, as many pre-service teachers found the platform captivating, which in turn stimulated their interest and creativity. This aligns with existing literature that underscores the importance of engaging and interactive tools in the educational process, particularly in the context of developing CT skills (Bers et al., 2014).

According to the pre-service teachers in this study visual programming significantly fostered their interest by creating and improving their motivation, active learning and fun. The use of a visual programming tool, Scratch, made abstract CT concepts more concrete and relatable, thereby increasing their intrinsic motivation to engage with the material. Participants reported that the interactive nature of visual programming enhanced active learning, as it required them to apply theoretical CT concepts in practical scenarios, hence, deepening their understanding. The gamified elements and creative freedom inherent in visual programming also introduced an element of play and fun, making the learning process enjoyable and stimulating. This combination of increased motivation, active engagement, and enjoyment contributed to a more effective and engaging learning experience, ultimately leading to a deeper and more sustained interest in the development of computational thinking. This aligns with the existing literature which suggests that incorporating elements of play in learning environments can enhance student interest, which consequently improves outcomes among others. In his book titled "Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play”, Mitchell Resnick highlights the principle of play in the development of CT in visual programming environments (Resnick, 2017). He argues that play is not just fun and games, it is a critical educational tool that creates an intrinsic motivation as they are enjoyable and engaging. He further posits that this intrinsic motivation sustains interest over long periods of learning, crucial for deep learning. Sáez-López et al. (2020) who explored the effect of training in visual programming for pre-service teachers in Spain, discovered that visual programming improves pre-service teachers’ interest by increasing motivation, active learning and fun. Saez-Lopez et al. (2022) investigated perceptions of pre-service teachers on visual programming in Columbia. Although the study did not establish a significant improvement in learning, it was discovered that visual programming significantly improves motivation, active learning and fun, translating into enhanced interest in learning among pre-service teachers.

Creativity was the second key element that was cited by most pre-service teachers as a significant enhancer in the use of visual programming for CT development. Participants believed that the opportunity to create their own Scratch programs stimulated their innovation and creative skills. By designing their own projects, pre-service teachers engaged in a process of trial and error, problem-solving, and critical thinking, which are essential components of computational thinking. The open-ended nature of Scratch projects allowed participants to explore various directions and solutions, thereby stimulating creative thinking. Furthermore, the act of creating contextualized visual programs helped pre-service teachers to internalize CT concepts and skills more effectively. The opportunity to create and innovate within visual programming significantly enriched the CT experience, making the acquisition of CT skills both effective and enjoyable. This finding resonates with what other scholars discovered elsewhere. For example, Scherer et al. (2019) investigated the cognitive benefits of visual programming among students in Norway and the results showed that visual programming approaches improve students creativity and performance. In a similar study, Kovalkov et al. (2020) used a computational tool to measure creativity in visual programming environment among pre-service teachers. Upon completion of Scratch programming training, a computational tool was administered to evaluate participants' creativity in thinking and creation of artifacts. Results demonstrated that visual programming approach enhances creativity, which leads to effective acquisition of CT skills among pre-service teachers.

Moreover, the collaborative aspects of Scratch were frequently cited by participants as instrumental in their CT development. Most pre-service teachers attributed their progress in developing CT to the collaborative learning environment fostered by the platform. The opportunities for pair and group programming enabled them to actively engage in the creation and critical evaluation of knowledge, thereby deepening their understanding and mastery of CT concepts and skills. In addition, the Scratch collaborative feature which exposed the participants to a remote visual programming community, allowed participants to connect and learn from others remotely. This not only expanded their perspectives but also fostered their CT skills by exposing them to diverse approaches and solutions. This finding is supported by the view that collaborative learning environments are crucial in the development of complex cognitive skills, such as CT, as they encourage peer interaction, problem-solving, and the sharing of ideas (Brennan & Resnick, 2012; Vygotsky, 1978). Roque et al. (2016) explored the diverse and creative collaboration in the Scratch online community. It was discovered that the Scratch platform enables creative collaboration by allowing participants to design, build, and invent shared artifacts. Pinto & Escudeiro (2014) conducted a study in Portugal in which they investigated learning skills that are promoted by Scratch programming among pre-service teachers. After running a Scratch programming training within Information and Communication Technologies (ICT) course, the scholars found that Scratch promotes collaborative learning, and effective relationships among pre-service teachers.

The combination of these factors; interest, creativity, and collaboration, suggests that visual programming environments like Scratch are well suited to facilitate the development of CT among pre-service teachers. These insights contribute to the broader understanding of how visual programming can be effectively integrated into teacher education programs to enhance the development of CT.

5.2.2 Factors that aid foundational phase pre-service teachers in developing computational thinking with visual programming

The factors influencing the development of CT in visual programming approach among foundational phase pre-service teachers are multifaceted, with both positive and negative influences identified. The study highlights several key elements that contributed to effective CT development.

The most cited positive factor for the development of CT was the use of contextualized project-based pedagogical approaches. Pre-service teachers emphasized that opportunities to create their own visual programs were essential in practising and mastering programming skills. The ability to develop programs that were relevant to their personal contexts was particularly beneficial, supporting the social cultural theory idea that contextual relevance enhances engagement and facilitates deeper learning (Vygotsky, 1978). This finding aligns with Stefania et al. (2022), who, in their systematic review report titled "*Reviewing computational thinking in compulsory education*", reported that project-based approaches are commonly used to engage students in motivating and meaningful activities. By allowing students to apply CT in real-world scenarios, the project-based method not only promoted practical skill acquisition but also fostered a deeper understanding of CT concepts (Yang et al., 2023). Another study conducted by Iskandar et al. (2019) in Nigeria on the effectiveness of Project-Based Learning in visual programming supports this finding. In their study, the findings revealed that the PBL pedagogical model significantly improves students' CT concepts and skills development.

Another critical factor cited by participants was the understanding of the CT rationale. Pre-service teachers who comprehensively grasped the real-life applications of CT reported higher levels of motivation and engagement, which in turn facilitated their CT development. This underscores the importance of making the relevance of CT explicit in educational settings particularly among pre-service teachers who have limited to no computing background. When pre-service teachers understood the practical applications and potential impact of CT on daily lives and future careers, they were more eager to invest the necessary effort in mastering these skills. By making the real-world relevance of CT explicit, the researcher was able to foster a deeper engagement and commitment to learning, thereby facilitating the development of CT skills. The given CT rationale not only motivated participants but also helped them see the value of CT beyond the classroom, making the learning experience more meaningful and effective. As argued by Jang (2008), instruction rationale supports participants' motivation, engagement, and learning. In her study conducted in USA, the author explored factors that support students' motivation, engagement and learning in uninteresting activity. Results showed that a well-articulated externally provided rationale can attract students' attention even on uninteresting activities. The study further revealed that the rationale facilitates engagement and learning because it provides an activity's value and personal benefit. Ryan & Deci (2000) add that such personal relevance information helps participants to identify with and internalize the value of the task, allowing them to engage purposively in the learning activity.

Prior programming experience was also identified as a significant factor influencing CT development in this study. Participants with previous exposure to programming, such as those who had studied it in secondary school, reported a smoother transition and assimilation of CT concepts and skills. This finding is supported by the shared perceptions and performance of pre-service teachers with prior programming experience in CT reflective activities, suggesting that foundational knowledge of programming can significantly enhance the learning of CT concepts. These observations are consistent with Li (2021), who identified prior programming as a significant facilitator of CT development among pre-service teachers. Wilcox & Lionelle (2018) conducted a study to quantify the benefits of prior programming experience in a visual programming course. The visual programming training was divided into two groups: one with prior programming experience in visual programming (P) and another without (N). Both groups were trained with the same curriculum content and assessments. Results showed substantial advantages of prior programming experience, with P group outperforming N group by more than 6% on exams and 10% on visual programming projects.

However, despite existing literature supporting this finding, other studies did not find any significant differences in CT development due to prior programming experience (Kuhail et al., 2021; Savidis, 2024; Wohlin, 2002). These scholars, who conducted their studies in developed countries context, argued that this is because visual programming platforms are designed for people with limited or no programming skills, allowing those without prior experience to learn and develop CT skills as effectively as those with such experience. Considering these conflicting findings, the researcher believes that the unique context of this study, supports the significance of prior programming experience on the ability to assimilate visual programming concepts and skills as discovered by Li (2021) and Wilcox & Lionelle (2018). The implication is that scaffolding CT education with prior programming knowledge could be beneficial, particularly for those new to the field.

5.2.3 Factors that hinder the development of CT among foundational phase pre-service teachers in visual programming approach

As the provision of CT rationale is instrumental in facilitating CT development, its absence has negative implications. When the rationale behind CT is not well-articulated, it negatively affects engagement and learning. Pre-service teachers who struggled to understand the rationale behind CT for various reasons found it challenging to fully engage with the material. Essentially the difficulty in developing CT skills stems from not understanding the content value and its relevance to everyday life and future careers. When pre-service teachers do not understand the applications or benefits of CT activities, they lack the motivation to engage with the material effectively. This reveals the importance of educators to clearly and comprehensively provide the benefits and applications of CT in various real-life contexts. As highlighting how CT skills can be applied in real-world scenarios and future professional settings, educators can cultivate intrinsic motivation and foster a deeper understanding of CT concepts and skills among foundational phase pre-service teachers. As argued by Jang (2008), this approach not only enhances engagement but also encourages the practical application of CT skills, thereby enriching the overall CT development experience.

Lack of prior programming experience was identified as a significant hindrance to CT development among the foundational phase pre-service teachers. During the CT sessions, most participants attributed their difficulties in understanding CT concepts and skills to their lack of prior exposure to programming. They felt that assimilating new and unfamiliar CT concepts was challenging without a foundational background in programming. This observation aligns with Li's (2021) findings, where the author identified the lack of prior programming experience as a major barrier to the development of CT among educators. Furthermore, Wilcox and Lionelle (2018) quantified the benefits of prior programming experience, demonstrating that pre-service teachers with a programming background performed better in visual programming course, outperforming their peers in exams and projects by substantial margins. This finding is particularly pertinent in the context of foundational phase pre-service teachers who participated in this study, as their background did not include programming subjects, for most of them.

The study also identified the multimedia nature of the Scratch visual platform as a hindrance to the understanding of the underlying CT concepts among foundational phase pre-service teachers. As Scratch is designed to be an engaging and accessible tool for learning programming, some participants reported that its graphical interface created the illusion of simplicity, leading them to underestimate the complexity of the underlying CT concepts. This observation aligns with the findings of Mayer and Moreno (2003), who discovered that multimedia platforms could induce cognitive overload, where learners focus on surface-level features at the expense of deeper conceptual understanding. The participants' experiences suggest that while Scratch is effective in engaging learners, its visual design may inadvertently distract them from the core principles of CT. Underscoring the need for instructional strategies that balance engagement with rigorous conceptual learning. Literature shows that while visual programming tools can enhance CT concepts and skills, they may not always achieve the intended educational outcomes comprehensively (Koh et al., 2010; Kuhail et al., 2021; Wu & Su, 2021). For instance, Koh et al. (2010) conducted a study in Colorado, USA, to evaluate the CT concepts and skills that pre-service teachers acquire through visual programming. By employing semantic analysis, the researchers evaluated participants' visual projects to identify the existence of CT concepts and skills from game designing and coding. The study revealed that although visual programming tools offered motivational benefits, participants struggled to clearly articulate and demonstrate the specific CT concepts they had gained. The authors concluded that while visual programming approach can foster interest and engagement, it may fall short in ensuring a deep understanding and clear demonstration of CT concepts and skills without additional conceptual instructional support or scaffolding.

5.3 Conclusion

In this chapter, the findings of the study were discussed from the lenses of the existing literature reviewed in Chapter Two. Encouragingly, the findings of this study align with the existing literature. Most pre-service teachers were of the view that visual programming enhance CT development by fostering interest, creativity and collaboration. This chapter also discussed factors that aid or hinder the development of CT among pre-service teachers in the context of a visual programming approach. Moreover, the discussion highlighted the critical aspects of CT development. It was evident that the pre-service teachers' engagement and motivation were significantly influenced by the Scratch interactive interface, and CT rationale. Additionally contextual project-based pedagogical approach and prior programming experience were part of the discussion about factors that aid CT development. Additionally, lack of a clear rationale, prior programming exposure and the multimedia nature of Scratch platform were discussed as factors that may hinder the development of CT among foundational phase pre-service teachers. The next chapter presents the conclusions and recommendations of the study, offering insights into best practices for developing CT using visual programming among foundational phase pre-service teachers and suggesting avenues for future research to address the identified gaps.

CHAPTER SIX
CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

The previous chapter provided a discussion of the findings of this study. This last chapter offers an overall summary of the study. It starts with section (6.1) Introduction, followed by (6.2) Summary of the study, which highlights the key objectives, theories and research design and methodology adopted in the study. Section (6.3) Summary of the study's Chapters provides a recap of each chapter, highlighting their individual contributions to the study. Next, (6.4) Summary of the research findings presents an overview of the key findings, relating them back to the research questions. This is followed by (6.5) Limitations of this study, where the challenges faced during the research process are examined. The chapter then moves to (6.6) Recommendations for practice, providing practical suggestions based on the results of this study, to improve the application of visual programming in CT development among foundational phase pre-service teachers. Finally, section (6.7) Recommendations for Further Studies outlines potential avenues for future research, suggesting areas where additional investigation could yield significant insights and contributions to the broader field of computational thinking education.

6.2 Summary of the Study

This study aimed to explore the development of computational thinking among pre-service teachers using a visual programming approach. The study was contextualised within the Rhodes University Foundation Phase Pre-service Teachers. To achieve the objective of the study, the following two research questions were phrased:

1. How do foundational phase pre-service teachers develop computational thinking with visual programming?
2. What are the enbling and constraining factors that affect the development of computational thinking using visual programming approach among pre-service teachers?

The study was designed as a qualitative inquiry utilizing a case study methodology and located within an interpretivist paradigm. The researcher clearly discussed and justified the reasons for choosing the qualitative design as well as the case study methodology. A purposive sampling was employed to select forty-nine first-year foundational phase pre-service teachers. The rationale and criteria for selecting the research site and participants were sufficiently explained. The following research instruments were employed in this study: CT reflective tool, semi-structured interviews, focus group discussions, and daily reflective journals. The data collection tools were underpinned by Vygotsky (1978)'s social cultural theory constructs, the "code, connect, create" professional development model and the "use, modify, create" pedagogical model as theoretical frameworks. The Brennan & Resnick (2012)'s computational thinking framework was adopted as the framework for determining and analysing CT concepts and CT concepts and skills. The basis for the choice of employing these frameworks in this study was demystified and justified. Thematic analysis was used, and emerging themes were identified for qualitative data. Basic descriptive statistics were computed for quantitative data.

6.3 Summary of the study chapters

Chapter One of the study laid a foundation by defining important terms to provide their precise contextual application. In this chapter, the context and background of the study were provided. The problem statement was comprehensively given with clarity, highlighting the central issue addressed by the study, approaching it from the global to local context. The goal, purpose and significance of the study were stated and justified, establishing the relevance and potential impact of the study. Furthermore, the chapter clearly presented the research questions, which were designed to guide the investigation and ensure a structured approach to addressing the research problem. An overview of this thesis was also outlined, offering a roadmap for the reader to understand the structure and progression of the study. This comprehensive introduction set the stage for the subsequent chapters, ensuring that the reader has a clear understanding of the study's intent, scope, and context.

Chapter Two reviewed the latest and the most relevant literature related to this study. The aim of the study and the research questions guided the literature review process. All literature reviewed in this study was drawn from credible, authentic and authoritative electronic databases such as Google Scholar, Scopus, ERIC, Web of Science and Science Direct, while ensuring that all publications are from peer-reviewed journals. The chapter began by demystifying CT and visual programming as defined in the literature. Plugged and unplugged approaches in CT development were explained. Next, literature on CT integration approaches, CT assessment mechanisms and the benefits of integrating CT was reviewed. Due to the lack of literature on CT teacher education in South African context, most literature was drawn from international context. The literature reviewed on CT teacher education covered both pre-service and in-service teacher training. The literature highlighted the growing importance of CT education and the pressing need for research and interventions in pre-service teacher training, especially in the context of developing countries. Furthermore, the existing literature showed that visual programming is the practical approach that enhances the development of computational thinking among pre-service teachers.

In addition, this chapter discussed the theoretical and analytical frameworks that underpinned the study. The Vygotsky's (1978) social cultural theory was the main theory that underpinned this study. Informed by this theory, two models guided the intervention. The "code-connect-create" professional development model was discussed where only the first phase of the model was implemented in this study. The "use-modify-create" pedagogical model was also presented in this chapter. Finally, the Brennan and Resnick (2012)'s CT framework was discussed as an analytical framework for interpreting data and deducing meaning. The basis for adoption of these theories and models were adequately explained and justified.

Chapter Three presented the research design and methodology adopted in this study, located within the interpretivist paradigm using a case study method. The rationale for selecting this research design and methodology was clearly given and justified, ensuring alignment with the study's objectives. Detailed criteria for sampling and the determination of sample size were also given. Furthermore, the strengths of the data collection tools used in this study were highlighted while acknowledging their inherent shortcomings. Data collection methods used were CT reflective tool, semi-structured interviews, focus group discussions and reflective journals. Each tool was chosen for its ability to elicit rich data relevant to the study's objectives. The data collection tools were comprehensively described, providing insight into how data was collected and ensuring the credibility of findings. The chapter also presented the thematic data analysis process, illustrating how themes were coded, identified and analysed to derive meaningful insights. The use of triangulation was emphasized to foster the validity and reliability of the findings by cross-verifying data from multiple sources, methods and theories. Additionally, the chapter included a section on research evaluation, discussing criteria such as credibility, transferability, dependability, and confirmability, followed by reflexivity and positionality. This section underscored the rigorous process employed to maintain the integrity of the research. The chapter concluded by addressing ethical considerations of the study, ensuring that the study adhered to ethical guidelines throughout the research process. Issues such as informed consent, gaining access, and confidentiality were carefully handled to protect the participants' rights and well-being. This approach not only reinforced the study's methodological rigor but also its ethical integrity.

Chapter Four presented the results from the study. The data from the CT reflective tool, semi-structured interviews, focus group discussions and the reflective journals were central to this chapter. Each data source was pivotal in uncovering the pre-service teachers' experiences and perspectives regarding the development of CT through visual programming. The chapter identified and categorized the emerging themes from the collected data based on the research questions. These themes included the impact of visual programming on fostering interest, creativity, and collaboration, as well as various factors that facilitated or hindered the development of CT among the foundational phase pre-service teachers. In presenting the results, the chapter maintained a high level of dependability by quoting participants verbatim. This approach not only added authenticity to the findings but also allowed for deeper understanding of the participants' viewpoints. The direct quotes provided rich, qualitative insights into how the pre-service teachers perceived and engaged with the visual programming platform in developing CT.

Chapter Five outlined a thorough discussion of the findings of the study. The findings were compared and contrasted with prior literature reviewed in Chapter Two, highlighting both consistencies and divergencies. The existing findings that aligned with the findings of this study were cited to demonstrate the reliability and validity of the outcomes. For example, the role of visual programming in enhancing CT through fostering interest, creativity, and collaboration among pre-service teachers was found to be consistent with the literature (Koh et al., 2010; Li, 2021). The discussion of the findings touched on the specific ways in which visual programming influenced the development of CT. for instance, it was noted that by engaging with the platform, participants reported higher levels of motivation and creativity, which aligns with existing theories on the benefits of interactive and engaging learning environments (Papert, 1980). Additionally, the chapter addressed the various factors that aided or hindered CT development. Moreover, the discussion of the findings included suggestions and recommendations that are contextually relevant. These included the need for more hands-on, practical approaches in CT education and the importance of clearly articulating the rationale behind CT to enhance engagement and learning outcomes. This discussion emphasized the practical applications of CT in real-life problem-solving and future career contexts, encouraging pre-service teachers to integrate these aspects into their future teaching. The research questions were comprehensively addressed in this chapter, with each finding discussed in relation to the specific question(s) posed at the beginning of the study.

Chapter Six is the last chapter of this study, offering a comprehensive overview of the entire research process. This chapter begins by revisiting the study's objectives, and methodology, providing a cohesive summary of the research design and approach. It then outlines the major research findings discussed in the previous chapters, highlighting key insights and patterns that emerged throughout the study. The chapter also draws conclusions based on these findings, highlighting the implications for both theory and practice. For instance, it underscores how visual programming approach can significantly enhance CT among foundational phase pre-service teachers, particularly when these tools are used in a collaborative and creatively stimulating environment. These conclusions are grounded in the data collected through the CT reflective tool, semi-structured interviews, focus group discussions, and reflective journals. Furthermore, Chapter Six provides actionable recommendations for educators, curriculum developers, and policymakers. Among others, these recommendations emphasize the importance of integrating visual programming into teacher education programs to foster CT skills, the need for providing clear CT rationale to enhance engagement, the need to employ contextualized learning approaches, and the value of prior programming experience as a facilitator of CT development. The chapter also identifies gaps in the current research that present opportunities for future studies.

6.4 Summary of Key Research Findings

The key findings for this study are presented based on the research questions.

6.4.1 The development of CT using visual programming

This study found that visual programming significantly enhances the development of computational thinking. The pre-service teachers' CT performance scores showed notable improvement. Additionally, their perception of CT evolved from viewing it as mere computer knowledge to understanding it as a broad, transferable problem-solving skill set underpinned by computer science principles. Furthermore, the pre-service teachers believed that visual programming enhances CT by fostering interest, creativity, and collaboration.

6.4.2 Factors that aid the development of CT using visual programming

Several key factors were identified as facilitators of CT development using visual programming. These include:

- Contextualised project-based pedagogical approach: This approach allowed pre-service teachers to practice and master programming skills by creating relevant visual programs. By working on projects that are meaningful and contextually relevant, pre-service teachers were able to apply CT concepts in practical scenarios, thereby deepening their understanding and skills.

- Provision of CT Rationale**:** Clearly explaining the real-life applications and importance of CT motivated participants to engage with and develop their CT skills. Understanding the practical benefits and relevance of CT helped to increase their motivation and commitment to learning.
- Prior programming experience: Pre-service teachers with prior programming experience demonstrated better CT development. Their foundational knowledge allowed them to assimilate new CT concepts more easily and effectively, highlighting the advantage of prior experience in programming education.

6.4.3 Factors that hinder the development of CT using visual programming

This study also identified several constraining factors that impede the development of CT among foundational phase pre-service teachers when using a visual programming approach. The following key factors were identified:

- Lack of a clear CT rationale**:** the absence of a clearly articulated rationale for CT hinders pre-service teachers' understanding and engagement with CT concepts and skills. Without a solid foundation in the importance and applications of CT, PSTs struggle to see its relevance, which negatively impacts their motivation and learning.
- Lack of prior programming experience: participants without prior exposure to programming found it challenging to grasp CT concepts and skills. The unfamiliarity with basic programming principles made it difficult for them to assimilate new CT-related knowledge, highlighting the need for scaffolding strategies.
- Distraction due to multimedia elements: The graphic and multimedia nature of Scratch, while engaging, often diverted participants' attention away from the underlying CT concepts. This led to cognitive overload and a superficial understanding of CT concepts, as participants focused more on the visual and interactive aspects of the tool rather than the underlying computational logic.

This study has made significant contributions to the research on teacher training in CT using visual programming within the context of a developing country. The findings underscore the importance of integrating visual programming in CT teacher training programs into CT teacher training programs, highlighting both the benefits and challenges associated with its use. The research provides a rich understanding of how visual programming can effectively foster CT skills among foundational phase pre-service teachers, particularly through fostering interest, creativity, and collaboration. it also identifies key factors that either aid or hinder the development of CT, such as project-based pedagogical approaches, provision of clear CT rationale, prior programming experience, and the potential cognitive overload caused by the graphic multimedia nature of tools like Scratch.

Moreover, the study's context-specific insights emphasize the need for tailored approaches in developing countries, where educational resources and prior exposure to computing may vary significantly from developed regions. The implications of these findings suggest that CT teacher training programs must be designed with an awareness of these contextual factors to maximise their effectiveness. Addressing both the enabling and hindering factors, this study offers valuable recommendations for future research and practical applications in the field of CT education.

The next sections present the limitations of the study and provides recommendations practice and directions for further studies.

6.5 Limitations of the study

Although significant efforts were applied in the preparation and conduct of this study, certain limitations should be acknowledged:

- Though the sample size was adequate, the use of purposive and convenient sampling limits the applicability of the findings to the specific participants of this study. Therefore, the results cannot be generalized to all teacher training institutions in South Africa.
- The study focused on the use of specific visual programming tool (Scratch), and the findings may not be applicable to other tools of programming environments.
- The availability of resources and support during the study might not reflect typical conditions in all teacher training institutions, potentially impacting the generalizability of the results.

These limitations should be taken into account when interpreting the study's findings and considering their broader implications.

6.6 Recommendations for practice

Based on the findings of this study, the researcher offers the following recommendations:

- Pre-service teacher training programs should integrate visual programming early in the curriculum to enhance computational thinking skills. Early exposure can build a strong foundation and foster a deeper understanding of CT concepts.
- Prese-service teacher education programme needs to adopt contextualised learning approaches that allow pre-service teachers to apply their CT skills in real-world scenarios. This can help bridge the gap between theoretical knowledge and practical application.
- Although visual programming is engaging and fosters interest, it is essential to balance this with rigorous conceptual learning. Ensuring that pre-service teachers understand the underlying principles of CT. Instruction tools like CT Reflective activities adopted in this study can be used to foster conceptual learning of CT.
- The study proposes the need to provide ongoing support and resources to pre-service teachers to reinforce their CT development and address any challenges they may face. This could include access to mentoring, additional practice materials, and opportunities for collaborative learning in visual programming.
- **Teacher education programme need to** implement robust assessment and feedback mechanisms to monitor pre-service teachers' progress in developing CT skills. Regular feedback can help identify areas for improvement and ensure continuous growth.

6.7 Recommendations for further studies

- **As** this study explored the use of visual programming for CT development from the perspective of pre-service teachers, it did not delve into learners' experiences. Further studies should focus on exploring learners' perceptions of CT development using visual programming.
- Research on the effectiveness of visual programming in improving learner achievement is still in its infancy, particularly in South African context. Future research should evaluate the impact of visual programming on learner achievement at foundational, primary and secondary levels.
- Explore the integration of visual programming with other disciplines to see how it can be used to enhance interdisciplinary learning and problem-solving skills across the curriculum in South African context.

It should be noted that this piece of research has just laid a foundation for future research on the use of visual programming in the development of computational thinking. It is hoped that more research will be conducted to provide deeper insights into the effectiveness and broader applicability of visual programming in developing CT skills in various educational contexts.

REFERENCES

Adu, P. (2019). *A step-by-step guide to qualitative data coding*. Routledge.

Ahlin, E. (2019). *Semi-Structured Interviews With Expert Practitioners: Their Validity and Significant Contribution to Translational Research*. SAGE Publications Ltd. https://doi.org/10.4135/9781526466037

Akhtar, I. (2016). Research Design. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2862445

Ameri, M. (2020). Criticism of the Sociocultural Theory. *Budapest International Research and Critics Institute (BIRCI-Journal): Humanities and Social Sciences*, *3*(3), 1530–1540. https://doi.org/10.33258/birci.v3i3.1082

Amineh, R. J., & Asl, H. D. (2015). *Review of Constructivism and Social Constructivism*.

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Angevine, C., Cator, K., Roschelle, J., Thomas, S. A., Waite, C., & Weisgrau, J. (2017). *Computational Thinking for a Computational World*. http://hdl.handle.net/20.500.12265/62

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, *148*, 103807. https://doi.org/10.1016/j.compedu.2020.103807

Ary, D., Jacobs, L. C., Razavieh, A., & Ary, D. (2010). *Introduction to research in education* (8th ed). Wadsworth.

Bans-Akutey, A., & Tiimub, B. (2021). *Triangulation in Research*. 3392. https://doi.org/10.20935/AL33922

Bassey, M. (1999). *Case study research in educational settings*. Open University Press.

Bazhouni, M. (2018). Integrating Critical Thinking Skills in Higher Education. *Education and Linguistics Research*, *4*(1), 65. https://doi.org/10.5296/elr.v4i1.12964

Bell, T., Duncan, C., & Rainer, A. (2017). What is coding? In *Creating the Coding Generation in Primary Schools*. Routledge.

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

Bettez, S. C. (2015). Navigating the complexity of qualitative research in postmodern contexts: Assemblage, critical reflexivity, and communion as guides. *International Journal of*

Qualitative Studies in Education, *28*(8), 932–954. http://dx.doi.org/10.1080/09518398.2014.948096

Blum, L., & Cortina, T. J. (2007). CS4HS: An outreach program for high school CS teachers. *ACM SIGCSE Bulletin*, *39*(1), 19–23. https://doi.org/10.1145/1227504.1227320

Boddy, C. R. (2016). Sample size for qualitative research. *Qualitative Market Research: An International Journal*, *19*(4), 426–432. https://doi.org/10.1108/QMR-06-2016-0053

Booth, A. (2006). Clear and present questions: Formulating questions for evidence based practice. *Library Hi Tech*, *24*(3), 355–368. https://doi.org/10.1108/07378830610692127

Borthick, F., Jones, D., & Wakai, S. (2003). Designing Learning Experiences within Learners' Zones of Proximal Development (ZPDs): Enabling Collaborative Learning On-Site and Online | Journal of Information Systems | American Accounting Association. *Journal of Information Systems*, *17*(1), 107–134. https://doi.org/10.2308/jis.2003.17.1.107

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72. https://doi.org/10.1145/3137065.3137069

Breen, R. L. (2006). A Practical Guide to Focus-Group Research. *Journal of Geography in Higher Education*, *30*(3), 463–475. https://doi.org/10.1080/03098260600927575

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*.

Brooks, R., Riele, K. te, & Maguire, M. (2014). *Ethics and Education Research*. SAGE.

Brown, E. A., & Brown, R. S. (2020). *The Effect of Advanced Placement Computer Science Course Taking on College Enrollment*. http://www.westcoastanalytics.com/uploads/6/9/6/7/69675515/longitudinal_study_-_combined_report_final_3_10_20__jgq_.pdf

Bryman, A., Becker, S., & Sempik, J. (2008). Quality Criteria for Quantitative, Qualitative and Mixed Methods Research: A View from Social Policy. *International Journal of Social Research Methodology*, *11*(4), 261–276. https://doi.org/10.1080/13645570701401644

Bull, G., Garofalo, J., & Hguyen, N. R. (2020). Thinking about computational thinking. *Journal of Digital Learning in Teacher Education*, *36*(1), 6–18. https://doi.org/10.1080/21532974.2019.1694381

Cabrera, K., Morreale, P., & Li, J. J. (2018). Computer science + education: An assessment of CS professional development. *Journal of Computing Sciences in Colleges*, *33*(3), 141–147.

Caeli, E. N., & Yadav, A. (2020). Unplugged Approaches to Computational Thinking: A Historical Perspective. *TechTrends*, *64*(1), 29–36. https://doi.org/10.1007/s11528-019-00410-5

Çakiroğlu, Ü., & Çevik, İ. (2022). A framework for measuring abstraction as a sub-skill of computational thinking in block-based programming environments. *Education and Information Technologies*, *27*(7), 9455–9484. https://doi.org/10.1007/s10639-022-11019-2

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

Cambon, L., Minary, L., Ridde, V., & Alla, F. (2012). Transferability of interventions in health education: A review. *BMC Public Health*, *12*(1), 497. https://doi.org/10.1186/1471-2458-12-497

Carter, N., Bryant-Lukosius, D., DiCenso, A., Blythe, J., & Neville, A. J. (2014). The Use of Triangulation in Qualitative Research. *Oncology Nursing Forum*, *41*(5), 545–547. https://doi.org/10.1188/14.ONF.545-547

Castleberry, A., & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds? *Currents in Pharmacy Teaching and Learning*, *10*(6), 807–815. https://doi.org/10.1016/j.cptl.2018.03.019

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

Chen, P., Yang, D., Metwally, A. H. S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education*, *10*(1), 47. https://doi.org/10.1186/s40594-023-00434-7

Choi, K. (2020, December 3). Reflective journals in qualitative inquiry. *QualPage*. https://qualpage.com/2020/12/03/reflective-journals-in-qualitative-inquiry/

Code.org. (2018, April 20). Code.org + resourceful teachers = higher student achievement! *Medium*. https://codeorg.medium.com/code-org-resourceful-teachers-higher-student-achievement-8be1efdec06e

Code.org, CSTA, & ECEP Alliance. (2022). *2022 State of Computer Science Education: Understanding Our National Imperative*. https://advocacy.code.org/stateofcs

Cohen, L., Manion, L., & Morrison, K. (2017). *Research Methods in Education* (8th ed.). Routledge. https://www.routledge.com/Research-Methods-in-Education/Cohen-Manion-Morrison/p/book/9781138209886

Computational Thinking – Callysto. (2023). https://www.callysto.ca/computational-thinking-tests/

Conducting Semi-Structured Interviews. (2015). In W. C. Adams, *Handbook of Practical Program Evaluation* (1st ed., pp. 492–505). Wiley. https://doi.org/10.1002/9781119171386.ch19

Conger, S., Krauss, K. E., & Simuja, C. (2017). New pedagogical approaches with technologies. *International Journal of Technology and Human Interaction* (IJTHI), 13(4), 62-76.

Cook, D. L. (1967). *The Impact of the Hawthorne Effect in Experimental Designs in Educational Research. Final Report*. https://eric.ed.gov/?id=ED021308

Cruz Castro, L. M., Magana, A. J., Douglas, K. A., & Boutin, M. (2021). Analyzing Students' Computational Thinking Practices in a First-Year Engineering Course. *IEEE Access*, *9*, 33041–33050. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3061277

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking A guide for teachers*. Computing at School. https://eprints.soton.ac.uk/424545/1/150818_Computational_Thinking_1_.pdf

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

Cutumisu, M., Adams, C., & Lu, C. (2019). A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

Dagiene, V., & Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. *Springer*, *5090*, 19–30. https://doi.org/10.1007/978-3-540-69924-8_2

Davis, G. J. (1991). An introduction to computer programming for students with no programming experience. *Proceedings Frontiers in Education Twenty-First Annual Conference. Engineering Education in a New World Order*, 748–752. https://doi.org/10.1109/FIE.1991.187592

DBE. (2022). *Deputy Minister Mhaule participates in panel discussions during 2022 GovTech Conference*. https://www.education.gov.za/ArchivedDocuments/ArchivedArticles/DM-participates-at-GovTech.aspx

DBE & PED. (2022). *Provinces are gearing up for the implementation of Coding and Robotics in SA schools*. https://www.education.gov.za/ArchivedDocuments/ArchivedArticles/Coding-And-Robotics-In-School-922.aspx

de Groot, J. (2018). *Teaching Computational Thinking-What do our educators need?* http://resolver.tudelft.nl/uuid:112f3fe3-b568-4555-a68e-3b599603c45d

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Denscombe, M. (2017). *EBOOK: The Good Research Guide: For Small-Scale Social Research Projects*. McGraw-Hill Education (UK).

Department of Basic Education. (2021). *Grade R-3 Coding and Robotics Draft CAPS Final 19Mar2021.pdf*. https://www.education.gov.za/Portals/0/Documents/Legislation/Call%20for%20Comments/draftcodingandroboticscurriculum/Grade%20R-3%20Coding%20%20and%20Robotics%20Draft%20CAPS%20Final%2019Mar2021.pdf?ver=2021-03-24-164612-000

Department of Basic Education. (2022). *Annual Performance Plan 2022/2023*. https://www.education.gov.za/Portals/0/Documents/Reports/DBE%20APP%20202223.PDF?ver=2022-04-06-140038-883

DeVries, R. (1997). Piaget's Social Theory. *Educational Researcher*, *26*(2), 4–17. https://doi.org/10.3102/0013189X026002004

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 906–912. https://doi.org/10.1145/3287324.3287431

Donkoh, S. (2023). Application of triangulation in qualitative research. *Journal of Applied Biotechnology & Bioengineering*, *10*(1), 6–9. https://doi.org/10.15406/jabb.2023.10.00319

Douglas, S. (2022, October 4). Benefits of Explicitly Teaching Computational Thinking. *SAM Labs*. https://samlabs.com/us/benefits-of-computational-thinking/

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, 60–69. https://doi.org/10.1145/2670757.2670774

European Commission. Joint Research Centre. (2016). *Developing computational thinking in compulsory education: Implications for policy and practice.* Publications Office. https://data.europa.eu/doi/10.2791/792158

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2020). Assessing 4th Grade Students' Computational Thinking through Scratch Programming Projects. *Informatics in Education - An International Journal*, *19*(4), 611–640.

Finke, S., Kemény, F., Sommer, M., Krnjic, V., Arendasy, M., Slany, W., & Landerl, K. (2022). Unravelling the numerical and spatial underpinnings of computational thinking: A pre-registered replication study. *Computer Science Education*, *32*(3), 313–334. https://doi.org/10.1080/08993408.2022.2079864

Fragapane, V., & Standl, B. (2021). Work in Progress: Creative Coding and Computer Science Education – From Approach to Concept. *2021 IEEE Global Engineering Education Conference (EDUCON)*, 1233–1236. https://doi.org/10.1109/EDUCON46332.2021.9453951

Frank, H., & Hatak, I. (2014). Doing a research literature review. In A. Fayolle & M. Wright (Eds.), *How to Get Published in the Best Entrepreneurship Journals*. Edward Elgar Publishing. https://doi.org/10.4337/9781782540625.00012

Gerring, J. (2007). *Case Study Research: Principles and Practices*. Cambridge University Press. https://books.google.mw/books?hl=en&lr=&id=CbetAQAAQBAJ&oi=fnd&pg=PR9&dq=a+strategy+that+has+worked+in+one+context+does+not+necessary+imply+that+it+will+work+everywhere+in+the+same+way+case+study+methodology&ots=kdy0FMPWuJ&sig=JKhZmu3f59qZbBKgTSK6gtgaKh4&redir_esc=y#v=onepage&q&f=false

Gibbs, A. (2012). *Focus groups and group interviews*. *186*, 192.

Golafshani, N. (2015). Understanding Reliability and Validity in Qualitative Research. *The Qualitative Report*. https://doi.org/10.46743/2160-3715/2003.1870

Google for Education. (2022). *Google for Education: Computational Thinking*. https://edu.google.com/resources/programs/exploring-computational-thinking/

Goulding, C. (2002). *Grounded Theory: A Practical Guide for Management, Business and Market Researchers*. 1–186.

Grant, C., & Osanloo, A. (2014). UNDERSTANDING, SELECTING, AND INTEGRATING A THEORETICAL FRAMEWORK IN DISSERTATION RESEARCH: CREATING THE BLUEPRINT FOR YOUR "HOUSE." *Administrative Issues Journal*, *4*(2). https://dc.swosu.edu/aij/vol4/iss2/4

Greyling, J. (Ed.). (2023). Coding Unplugged—A Guide to Introducing Coding and Robotics to South African Schools. In *Transforming Entrepreneurship Education* (p. 155). Springer International Publishing. https://doi.org/10.1007/978-3-031-11578-3

Grocott's Mail. (2023, April 18). *Makana Schools ICT Day*. https://grocotts.ru.ac.za/2023/04/18/makana-schools-ict-day/

Grossoehme, D. H. (2014). Overview of Qualitative Research. *Journal of Health Care Chaplaincy*, *20*(3), 109–122. https://doi.org/10.1080/08854726.2014.925660

Grover, S. (2011). *Robotics and Engineering for Middle and High School Students to Develop Computational Thinking*.

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

Hammer, B., Virgili, E., & Bilotta, F. (2023). Evidence-based literature review: De-duplication a cornerstone for quality. *World Journal of Methodology*, *13*(5), 390–398. https://doi.org/10.5662/wjm.v13.i5.390

Hammersley, M., & Traianou, A. (2012). *Ethics in Qualitative Research: Controversies and Contexts*. SAGE.

Hegel, G. W. F. (2002). *Hegel's Philosophy of Nature: Volume I Edited by M J Petry*. Taylor and Francis.

Hsu, T.-C., Chang, S.-C., & Hung, Y.-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/10.1016/j.compedu.2018.07.004

Hunsaker, E. (2020). Computational Thinking. *The K-12 Educational Technology Handbook*, 91–109.

Hydén, L.-C., & Bülow, P. (2003). Who's talking: Drawing conclusions from focus groups—some methodological considerations. *International Journal of Social Research Methodology*, *6*(4), 305–321. https://doi.org/10.1080/13645570210124865

İlic, U. (2021). The Impact of Scratch-Assisted Instruction on Computational Thinking (CT) Skills of Pre-Service Teachers. *International Journal of Research in Education and Science*, *7*(2), Article 2. https://doi.org/10.46328/ijres.1075

Iskandar, Jalinus, N., Ridwan, Jama, J., Wakhinuddin, Irfan, D., Efendi, R., & M.Ropianto. (2019). Competence Improvement of Visual Basic Programming Through Project-Based Learning. *International Journal of Scientific & Technology Research*. https://www.semanticscholar.org/paper/Competence-Improvement-of-Visual-Basic-Programming-Iskandar-Jalinus/619f1ae87864071c52c6d8ef6078187b66c2cd10?utm_source=consensus

Jacob, S. R., & Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, *1*(1). https://doi.org/10.26716/jcsi.2018.01.1.1

Janesick, V. J. (1999). A Journal About Journal Writing as a Qualitative Research Technique: History, Issues, and Reflections. *Qualitative Inquiry*, *5*(4), 505–524. https://doi.org/10.1177/107780049900500404

Jang, H. (2008). Supporting students' motivation, engagement, and learning during an uninteresting activity. *Journal of Educational Psychology*, *100*(4), 798–811. https://doi.org/10.1037/a0012841

Janssen, E. M., Mainhard, T., Buisman, R. S. M., Verkoeijen, P. P. J. L., Heijltjes, A. E. G., Van Peppen, L. M., & Van Gog, T. (2019). Training higher education teachers' critical thinking and attitudes towards teaching it. *Contemporary Educational Psychology*, *58*, 310–322. https://doi.org/10.1016/j.cedpsych.2019.03.007

Jocius, R., Joshi, D., Dong, Y., Robinson, R., Cateté, V., Barnes, T., Albert, J., Andrews, A., & Lytle, N. (2020). Code, Connect, Create: The 3C Professional Development Model to Support Computational Thinking Infusion. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 971–977. https://doi.org/10.1145/3328778.3366797

Jones, B. R., & Flannigan, S. L. (2006). *Connecting the Digital Dots: Literacy of the 21st Century*.

Junior, P., Abib, G., & Hoppen, N. (2019). The Qualitative Report Validity in Qualitative Research: A Processual Approach. *Qualitative Report*, *24*, 98–112. https://doi.org/10.46743/2160-3715/2019.3443

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher Change Following a Professional Development Experience in Integrating Computational Thinking into Elementary Science. *Journal of Science Education and Technology*, *29*(1), 174–188. https://doi.org/10.1007/s10956-019-09798-4

Khoa, B. T., Hung, B. P., & Hejsalem-Brahmi, M. (2023). Qualitative research in social sciences: Data collection, data analysis and report writing. *International Journal of Public Sector Performance Management*, *12*(1–2), 187–209. https://doi.org/10.1504/IJPSPM.2023.132247

Kılıç, S., Gökoğlu, S., & Öztürk, M. (2021). A Valid and Reliable Scale for Developing Programming-Oriented Computational Thinking. *Journal of Educational Computing Research*, *59*(2), 257–286. https://doi.org/10.1177/0735633120964402

Knie, L., Standl, B., & Schwarzer, S. (2022). First experiences of integrating computational thinking into a blended learning in-service training program for STEM teachers. *Computer Applications in Engineering Education*, *30*(5), 1423–1439. https://doi.org/10.1002/cae.22529

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning. *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, 59–66. https://doi.org/10.1109/VLHCC.2010.17

Kovalkov, A., Segal, A., & Gal, Y. (2020). In the Eye of the Beholder? Detecting Creativity in Visual Programming Environments. *ArXiv*. https://www.semanticscholar.org/paper/In-the-Eye-of-the-Beholder-Detecting-Creativity-in-Kovalkov-Segal/fde40c8a49afd7ba8c91b94520b58287811d9756?utm_source=consensus

Kuhail, M. A., Farooq, S., Hammad, R., & Bahja, M. (2021). Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review. *IEEE Access*, *9*, 14181–14202. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3051043

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Levy, P. S., & Lemeshow, S. (2013). *Sampling of Populations: Methods and Applications*. John Wiley & Sons.

Li, Q. (2021). *Computational thinking and teacher education: An expert interview study*. *2*(3). https://doi.org/10.1002/hbe2.224

Liao, H., & Hitchcock, J. (2018). Reported credibility techniques in higher education evaluation studies that use qualitative methods: A research synthesis. *Evaluation and Program Planning*, *68*, 157–165. https://doi.org/10.1016/j.evalprogplan.2018.03.005

Lim, J. M.-H. (2010). Commenting on research results in applied linguistics and education: A comparative genre-based investigation. *Journal of English for Academic Purposes*, *9*(4), 280–294. https://doi.org/10.1016/j.jeap.2010.10.001

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic Inquiry*. SAGE. https://books.google.com/books/about/Naturalistic_Inquiry.html?id=2oA9aWlNeooC

Lodi, M. <1988>. (2020). *Introducing Computational Thinking in K-12 Education: Historical, Epistemological, Pedagogical, Cognitive, and Affective Aspects* [Doctoral Thesis, Alma Mater Studiorum - Università di Bologna]. https://doi.org/10.6092/unibo/amsdottorato/9188

Lu, J. J., & Fletcher, G. H. L. (2009). Thinking about computational thinking. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 260–264. https://doi.org/10.1145/1508865.1508959

M Esteve-Mon, F., Adell-Segura, J., Ángeles Llopis Nebot, M., 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

MacIntosh, J. A. (1993). Focus groups in distance nursing education. *Journal of Advanced Nursing*, *18*(12), 1981–1985. https://doi.org/10.1046/j.1365-2648.1993.18121981.x

Main, P. (2023). *Sociocultural Theory*. https://www.structural-learning.com/post/sociocultural-theory

Malčík, M., Rakowski, R., Miklošíková, M., Zemčík, T., & Feber, J. (2023). Philosophical Background of Computational Thinking. *International Journal of Emerging Technologies in Learning (iJET)*, *18*(17), Article 17. https://doi.org/10.3991/ijet.v18i17.41139

Maraza-Quispe, B., Maurice, A., Melina, O., Marianela, L., Henry, L., Cornelio, W., & Ernesto, L. (2021). Towards the Development of Computational Thinking and Mathematical Logic through Scratch. *International Journal of Advanced Computer Science and Applications*, *12*(2). https://doi.org/10.14569/IJACSA.2021.0120242

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.

Martin, W., Brennan, K., Tally, W., & Cervantes, F. (2014). *Identifying and Assessing Computational Thinking Practices*. https://repository.isls.org//handle/1/1035

Matusov, E., & Hayes, R. (2000). Sociocultural critique of Piaget and Vygotsky. *New Ideas in Psychology*, *18*(2), 215–239. https://doi.org/10.1016/S0732-118X(00)00009-X

May, T., & Perry, B. (2022). *Social Research: Issues, Methods and Process*. McGraw-Hill Education (UK).

Mayer, R. E., & Moreno, R. (2003). Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist*, *38*(1), 43–52. https://doi.org/10.1207/S15326985EP3801_6

McCormack, A. (2014). *The-e-Skills-Manifeso*. European Schoolnet. https://www.ademccormack.com/wp-content/uploads/2017/07/The-e-Skills-Manifeso.pdf

Mikecz, R. (2012). Interviewing Elites: Addressing Methodological Issues. *Qualitative Inquiry*, *18*(6), 482–493. https://doi.org/10.1177/1077800412442818

Mohaghegh, M., & McCauley, M. (2016). *Computational Thinking: The Skill Set of the 21st Century*. *7*.

Molina-Ayuso, Á., Adamuz-Povedano, N., Bracho-López, R., & Torralbo-Rodríguez, M. (2022). Introduction to Computational Thinking with Scratch for Teacher Training for Spanish Primary School Teachers in Mathematics. *Education Sciences*, *12*(12), Article 12. https://doi.org/10.3390/educsci12120899

Moloi, T., Matabane, M. E., Simuja, C., Seo, B. I., & Tarman, B. (2023). Constructing a social justice curriculum policy in the 21st century. *Research in Educational Policy and Management*, 5(3), i-iv.

Moreno-León, J., Robles, G., & Román-González, M. (2015). *Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking*. *46*, 1–23.

Moser, A., & Korstjens, I. (2018). Series: Practical guidance to qualitative research. Part 3: Sampling, data collection and analysis. *European Journal of General Practice*, *24*(1), 9–18. https://doi.org/10.1080/13814788.2017.1375091

Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 2–10. https://doi.org/10.1145/2818314.2818320

Mullis, I., Von Davier, M., Foy, P., Fishbein, B., Reynolds, K., & Wry, E. (2023). *PIRLS 2021 International Results in Reading*. TIMSS & PIRLS International Study Center. https://doi.org/10.6017/lse.tpisc.tr2103.kb5342

Nakano, D., & Muniz Jr., J. (2018). Writing the literature review for empirical papers. *Production*, *28*(0). https://doi.org/10.1590/0103-6513.20170086

Noble, H., & Heale, R. (2019). Triangulation in research, with examples. *Evidence-Based Nursing*, *22*(3), 67–68. https://doi.org/10.1136/ebnurs-2019-103145

Noble, H., & Smith, J. (2015). Issues of validity and reliability in qualitative research. *Evidence Based Nursing*, *18*(2), 34–35. https://doi.org/10.1136/eb-2015-102054

Opoku, A., Ahmed, V., & Akotia, J. (2016). Choosing an appropriate research methodology and method. In *Research Methodology in the Built Environment*. Routledge.

Ortlipp, M. (2008). Keeping and Using Reflective Journals in the Qualitative Research Process. *International Journal of Applied Linguistics & English Literature*, *2*(6), 118–128. https://doi.org/10.7575/aiac.ijalel.v.2n.6p.118

Ou Yang, F.-C., Lai, H.-M., & Wang, Y.-W. (2023). Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Computers & Education*, *195*, 104721. https://doi.org/10.1016/j.compedu.2022.104721

Owen Doody, M. N. (2013). Preparing and conducting interviews to collect data. *Nurse Researcher*, *20*(5), 28–32. https://doi.org/10.7748/nr2013.05.20.5.28.e327

Pandey, D. P., & Pandey, D. M. M. (2021). *RESEARCH METHODOLOGY: TOOLS AND TECHNIQUES*.

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies*, *22*(2), 421–443. https://doi.org/10.1007/s10639-016-9475-z

Pathak, V., Jena, B., & Kalra, S. (2013). Qualitative research. *Perspectives in Clinical Research*, *4*(3), 192. https://doi.org/10.4103/2229-3485.115389

Peracaula-Bosch, M., & González-Martínez, J. (2022). Developing computational thinking among preservice teachers. *QWERTY - Interdisciplinary Journal of Technology, Culture and Education*, *17*(1), Article 1. https://doi.org/10.30557/QW000049

Perez, A. D. F., & Valladares, G. M. (2018). Development and assessment of computational thinking: A methodological proposal and a support tool. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 787–795. https://doi.org/10.1109/EDUCON.2018.8363311

Piaw Chua, Y. (2012). Effects of computer-based testing on test performance and testing motivation. *Computers in Human Behavior*, *28*(5), 1580–1586. https://doi.org/10.1016/j.chb.2012.03.020

Pinto, A., & Escudeiro, P. (2014). The use of Scratch for the development of 21st century learning skills in ICT. *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–4. https://doi.org/10.1109/CISTI.2014.6877061

Polit, D. F., & Beck, C. T. (2010). Generalization in quantitative and qualitative research: Myths and strategies. *International Journal of Nursing Studies*, *47*(11), 1451–1458. https://doi.org/10.1016/j.ijnurstu.2010.06.004

Polly, D., Allman, B., Casto, A., & Norwood, J. (2017). *Sociocultural Perspectives of Learning*. https://pressbooks.pub/lidtfoundations/chapter/sociocultural-learning/

Poulakis, E., & Politis, P. (2021). Computational Thinking Assessment: Literature Review. In T. Tsiatsos, S. Demetriadis, A. Mikropoulos, & V. Dagdilelis (Eds.), *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 111–128). Springer International Publishing. https://doi.org/10.1007/978-3-030-64363-8_7

Prieto-Rodriguez, E., & Berretta, R. (2014). Digital technology teachers' perceptions of computer science: It is not all about programming. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 1–5. https://doi.org/10.1109/FIE.2014.7044134

Resnick, M. (2017). *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT Press.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Rich, P. J., Jones, B., Belikov, O., Yoshikawa, E., & Perkins, M. (2017). Computing and Engineering in Elementary School: The Effect of Year-long Training on Elementary Teacher Self-efficacy and Beliefs About Teaching Computing and Engineering. *International Journal of Computer Science Education in Schools*, *1*(1), Article 1. https://doi.org/10.21585/ijcses.v1i1.6

Rich, P. J., Larsen, R. A., & Mason, S. L. (2021). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, *53*(3), 296–316. https://doi.org/10.1080/15391523.2020.1771232

Riley, D. D., & Hunt, K. A. (2014). *Computational Thinking for the Modern Problem Solver*. CRC Press.

Roller, M. R., & Lavrakas, P. J. (2015). *Applied Qualitative Research Design: A Total Quality Framework Approach*. Guilford Publications.

Román-González, M., Moreno-León, J., & Robles, G. (2017). *Complementary Tools for Computational Thinking Assessment*.

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1), 42. https://doi.org/10.1186/s41239-017-0080-z

Roque, R., Rusk, N., & Resnick, M. (2016). Supporting Diverse and Creative Collaboration in the Scratch Online Community. In U. Cress, J. Moskaliuk, & H. Jeong (Eds.), *Mass Collaboration and Education* (pp. 241–256). Springer International Publishing. https://doi.org/10.1007/978-3-319-13536-6_12

Russell, G. M., & Kelly, N. H. (2002). Research as Interacting Dialogic Processes: Implications for Reflexivity. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *3*(3), Article 3. https://doi.org/10.17169/fqs-3.3.831

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, *55*(1), 68–78. https://doi.org/10.1037/0003-066X.55.1.68

Sáez-López, J. M., del Olmo-Muñoz, J., González-Calero, J. A., & Cózar-Gutiérrez, R. (2020). Exploring the Effect of Training in Visual Block Programming for Preservice Teachers. *Multimodal Technologies and Interaction*, *4*(3), Article 3. https://doi.org/10.3390/mti4030065

Saez-Lopez, J.-M., Vazquez-Cano, E., & Sevillano-Garcia, M.-L. (2022). Scratch and visual block programming in mathematics: Perceptions of preservice teachers in Columbia. *Scientia Paedagogica Experimentalis*, *59*(1), 29–50. https://doi.org/10.57028/S59-029-Z1003

Safar, A., & AlKhezzi, F. (2013). Beyond Computer Literacy: Technology Integration and Curriculum Transformation. *College Student Journal*, *47*(4), 614–626.

Sandelowski, M. (2001). Real qualitative researchers do not count: The use of numbers in qualitative research. *Research in Nursing & Health*, *24*(3), 230–240. https://doi.org/10.1002/nur.1025

Savidis, A. (2024). *Programming Experience Requirements for Future Visual Development Environments*. 284–292. https://www.scitepress.org/Link.aspx?doi=10.5220/0011082500003182

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, *111*(5), 764–792. https://doi.org/10.1037/edu0000314

Schoonenboom, J., & Johnson, R. B. (2017). How to Construct a Mixed Methods Research Design. *Kolner Zeitschrift Fur Soziologie Und Sozialpsychologie*, *69*(Suppl 2), 107. https://doi.org/10.1007/s11577-017-0454-1

Scott, S., & Palincsar, A. (2013). *THE HISTORICAL ROOTS OF SOCIOCULTURAL THEORY*.

Scratch Foundation. (2023). *Scratch—About*. https://scratch.mit.edu/

Sepula, C., & Simuja, C. (2024). Pre-service Teachers' Pedagogical Awareness of Technology Integration in the Classroom. *In Global Perspectives on Teaching with Technology* (pp. 93-109). Routledge.

Shabani, D. B., & Lam, W. Y. (2013). A Review of Comparison Studies in Applied Behavior Analysis. *Behavioral Interventions*, *28*(2), 158–183. https://doi.org/10.1002/bin.1361

Shabani, K., Khatib, M., & Ebadi, S. (2010). Vygotsky’s Zone of Proximal Development: Instructional Implications and Teachers’ Professional Development. *English Language Teaching*, *3*(4), p237. https://doi.org/10.5539/elt.v3n4p237

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *EXPERIMENTAL AND QUASI-EXPERIMENTAL DESIGNS FOR GENERALIZED CAUSAL INFERENC*.

Shambare, B., Simuja, C., & Olayinka, T. A. (2022). Understanding the enabling and constraining factors in using the virtual lab: Teaching Science in rural schools in South Africa. *International Journal of Information and Communication Technology Education* (IJICTE), 18(1), 1-15.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Simons, H., & Usher, R. (Eds.). (2000). *Situated ethics in educational research*. Routledge.

Sk, P., Pandey, S., Dwivedi, S., Bajaj, N., & Kunder, S. (2020). Basics of Research Methods: An Overview for Beginners. *The Indian Practitioner*. https://www.semanticscholar.org/paper/Basics-of-Research-Methods%3A-An-Overview-for-Sk-Pandey/f5be572de51454942b335e1274730a098073753a

Simuja, C. (2023). Teachers' Experiences of Technology Integration in Sarah Batman District, Eastern Cape Province. *African Perspectives of Research in Teaching and Learning*, 7(3), 548-565.

Simuja, C.(2024) Exploration of Foundation Phase: Pre-service Teachers' Perspectives on Their Abilities to Integrate Technology in the Classroom. I*n Global Perspectives on Teaching with Technology* (pp. 110-126). Routledge.

Slevin, E., & Sines, D. (2013). Enhancing the truthfulness, consistency and transferability of a qualitative study: Utilising a manifold of approaches—ProQuest. *Nurse Researcher*, *7*(2). https://www.proquest.com/openview/f6ac811c0ca0ba13cd3f91a063a117fd/1?pq-origsite=gscholar&cbl=33100

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, *104*, 333–339. https://doi.org/10.1016/j.jbusres.2019.07.039

Stefania, B., Augusto, C., Panagiotis, K., Valentina, D., Patricia, W., Katja, E., Jeffrey, E., Anna, H. M., Eglė, J., Chiara, M., Vaida, M.-D., & Gabrielė, S. (2022). Reviewing Computational Thinking in Compulsory Education. *JRC Research Reports*, Article JRC128347. https://ideas.repec.org//p/ipt/iptwpa/jrc128347.html

Suri, H. (2011). Purposeful Sampling in Qualitative Research Synthesis. *Qualitative Research Journal*, *11*(2), 63–75. https://doi.org/10.3316/QRJ1102063

Szymkowiak, A., Melović, B., Dabić, M., Jeganathan, K., & Kundi, G. S. (2021). Information technology and Gen Z: The role of teachers, the internet, and technology in the education of young people. *Technology in Society*, *65*, 101565. https://doi.org/10.1016/j.techsoc.2021.101565

Tankiz, E., & Atman Uslu, N. (2023). Preparing Pre-Service Teachers for Computational Thinking Skills and its Teaching: A Convergent Mixed-Method Study. *Technology, Knowledge and Learning*, *28*(4), 1515–1537. https://doi.org/10.1007/s10758-022-09593-y

Taylor, C., & Gibbs, G. (2010). *What is Qualitative Data Analysis (QDA)?* https://onlineqda.hud.ac.uk/Intro_QDA/what_is_qda.html

Thurmond, V. A. (2001). The Point of Triangulation. *Journal of Nursing Scholarship*, *33*(3), 253–258. https://doi.org/10.1111/j.1547-5069.2001.00253.x

Tiruneh, D. T., Gu, X., De Cock, M., & Elen, J. (2018). Systematic design of domain-specific instruction on near and far transfer of critical thinking skills. *International Journal of Educational Research*, *87*, 1–11. https://doi.org/10.1016/j.ijer.2017.10.005

Tiruneh, D. T., Verburgh, A., & Elen, J. (2014). Effectiveness of Critical Thinking Instruction in Higher Education: A Systematic Review of Intervention Studies. *Higher Education Studies*, *4*(1), p1. https://doi.org/10.5539/hes.v4n1p1

Tolboom, J., Rijke, W., Bollen, L., & Eysink, T. (2018). Computational Thinking in Primary School: An Examination of Abstraction and Decomposition in Different Age Groups. *Informatics in Education - An International Journal*, *17*(1), 77–92.

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017). *Training Computational Thinking: Game-Based Unplugged and Plugged-in Activities in Primary School*.

Umutlu, D. (2022). An exploratory study of pre-service teachers' computational thinking and programming skills. *Journal of Research on Technology in Education*, *54*(5), 754–768. https://doi.org/10.1080/15391523.2021.1922105

Uprichard, E. (2013). Sampling: Bridging probability and non-probability designs. *International Journal of Asocial Research Methodology*, *1*(16). https://0-www.tandfonline.com.wam.seals.ac.za/doi/epdf/10.1080/13645579.2011.633391?needAccess=true&role=button

U.S. Bureau of Labor Statistics. (2018). https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Valls Pou, A., Canaleta, X., & Fonseca, D. (2022). Computational Thinking and Educational Robotics Integrated into Project-Based Learning. *Sensors*, *22*(10), 3746. https://doi.org/10.3390/s22103746

Vicary, S., Young, A., & Hicks, S. (2017). A reflective journal as learning process and contribution to quality and validity in interpretative phenomenological analysis. *Qualitative Social Work*, *16*(4), 550–565. https://doi.org/10.1177/1473325016635244

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

Vygotksy, L. S. (1978). *THE HISTORICAL ROOTS OF SOCIOCULTURAL THEORY*.

Vygotsky, L. S. (1980). *Mind in Society: Development of Higher Psychological Processes* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.). Harvard University Press.

Wa-Mbaleka, S. (2020). The Researcher as an Instrument. In A. P. Costa, L. P. Reis, & A. Moreira (Eds.), *Computer Supported Qualitative Research* (pp. 33–41). Springer International Publishing. https://doi.org/10.1007/978-3-030-31787-4_3

Watson, R. (2015). Quantitative research. *Nursing Standard*, *29*(31), 44–48. https://doi.org/10.7748/ns.29.31.44.e8681

Webb, M., Bell, T., Davis, N., Katz, Y. J., Reynolds, N., Chambers, D. P., Sysło, M. M., Fluck, A., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., Zagami, J., Micheuz, P., Chtouki, Y., & Mori, N. (2017). Computer Science in the School Curriculum: Issues and Challenges. *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing*, 421–431. https://doi.org/10.1007/978-3-319-74310-3_43

Webster, J., & Watson, R. T. (2024). *Analyzing the Past to Prepare for the Future: Writing a Literature Review*.

Weintrop, D., & Wilensky, U. (2015). Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 101–110. https://doi.org/10.1145/2787622.2787721

Wellington, J. (2015). *Educational Research: Contemporary Issues and Practical Approaches*. Bloomsbury Publishing.

Wilcox, C., & Lionelle, A. (2018). Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 80–85. https://doi.org/10.1145/3159450.3159480

Wing, J. (2009). Computational thinking. *Journal of Computing Sciences in Colleges*. https://doi.org/10.5555/1529995.1529997

Wing, J. (2011). *Research Notebook: Computational Thinking—What and Why?* Carneige Mellon University, School of Computer Science. https://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf

Wohlin, C. (2002). Is prior knowledge of a programming language important for software quality? *Proceedings International Symposium on Empirical Software Engineering*, 27–34. https://doi.org/10.1109/ISESE.2002.1166922

Wu, S.-Y., & Su, Y.-S. (2021). Visual Programming Environments and Computational Thinking Performance of Fifth- and Sixth-Grade Students. *Journal of Educational Computing Research*, *59*(6), 1075–1092. https://doi.org/10.1177/0735633120988807

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational Thinking as an Emerging Competence Domain. In *Competence-based Vocational and Professional Education* (Vol. 23, pp. 1051–1067). Springer International Publishing. https://doi.org/10.1007/978-3-319-41713-4_49

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), 5:1-5:16. 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

Yang, D., Snelson, C., & Feng, S. (2023). Identifying computational thinking in students through project-based problem-solving activities. *Information Discovery and Delivery*, *51*(3), 293–305. https://doi.org/10.1108/IDD-09-2022-0091

Yazan, B. (2015). Three Approaches to Case Study Methods in Education: Yin, Merriam, and Stake. *The Qualitative Report*. https://doi.org/10.46743/2160-3715/2015.2102

Yilmaz, K. (2013). Comparison of Quantitative and Qualitative Research Traditions: Epistemological, theoretical, and methodological differences. *European Journal of Education*, *48*(2), 311–325. https://doi.org/10.1111/ejed.12014

Yin, R. K. (2009). *Case Study Research: Design and Methods*. SAGE.

Yin, R. K. (2013). Validity and generalization in future case study evaluations. *Evaluation*, *19*(3), 321–332. https://doi.org/10.1177/1356389013497081

Yuan, R. (2023). Cultivating CT-oriented teachers in pre-service teacher education: What is there and what is missing? *Teachers and Teaching*, *29*(7–8), 924–945. https://doi.org/10.1080/13540602.2023.2215719

Yuksel, P., & Yildirim, S. (2015). *Theoretical Frameworks, Methods, and Procedures for Conducting Phenomenological Studies in Educational Settings*. *6*(20), 1–20.

Zaharin, N. L., Sharif, S., & Mariappan, M. (2018). Computational Thinking: A Strategy for Developing Problem Solving Skills and Higher Order Thinking Skills (HOTS). *International Journal of Academic Research in Business and Social Sciences*, *8*(10), Pages 1265-1278. https://doi.org/10.6007/IJARBSS/v8-i10/5297

Zapata-Caceres, M., Martin-Barroso, E., & Roman-Gonzalez, M. (2020). Computational Thinking Test for Beginners: Design and Content Validation. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 1905–1914. https://doi.org/10.1109/EDUCON45650.2020.9125368

Zhang, L., Nouri, J., & Rolandsson, L. (2020). Progression Of Computational Thinking Skills In Swedish Compulsory Schools With Block-based Programming. *Proceedings of the Twenty-Second Australasian Computing Education Conference*, 66–75. https://doi.org/10.1145/3373165.3373173

Appendix A1: CT Reflective Tool

Date: ________________________

Introduction

This is a Computational Thinking (CT) Reflective tool which is part of a MEd research intended to enhance and measure your knowledge and skills on CT. Be assured that under no circumstances will your details including the scores be revealed to a third party. Your responses and scores will be kept completely confidential. Please be aware that you may not know the answer to some of the questions. If you do not know the answer to a question, please answer with your best guess.

Instruction

Read each question carefully. The questions are followed by possible answers from which only one is correct. Circle/tick the appropriate option next to the answer of your choice. Kindly answer **ALL QUESTIONS** to the best of your ability.

Section A: Group Information

Group number: ________________________

Group members:

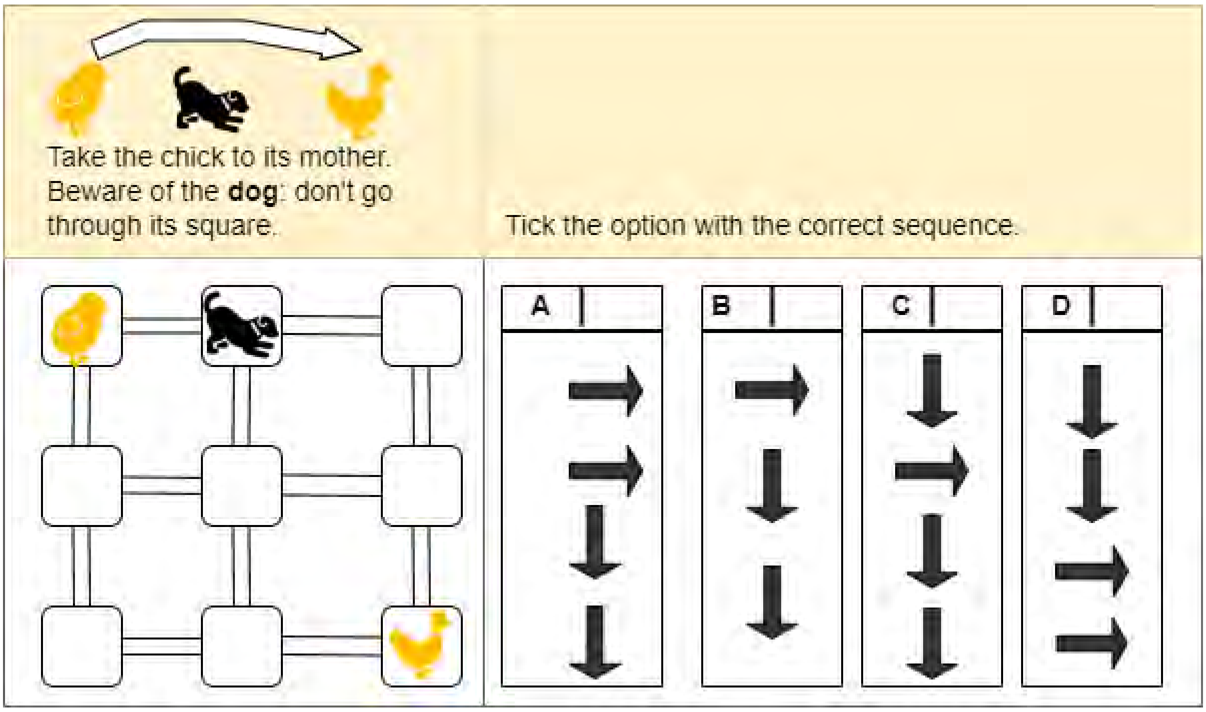
1. ______________________________
2. ______________________________
3. ______________________________
4. ______________________________

Section B: Understanding of CT

1. What is your understanding of the term "computational thinking"?

Section C: CT Concepts

Activity 1



End of Activity 1

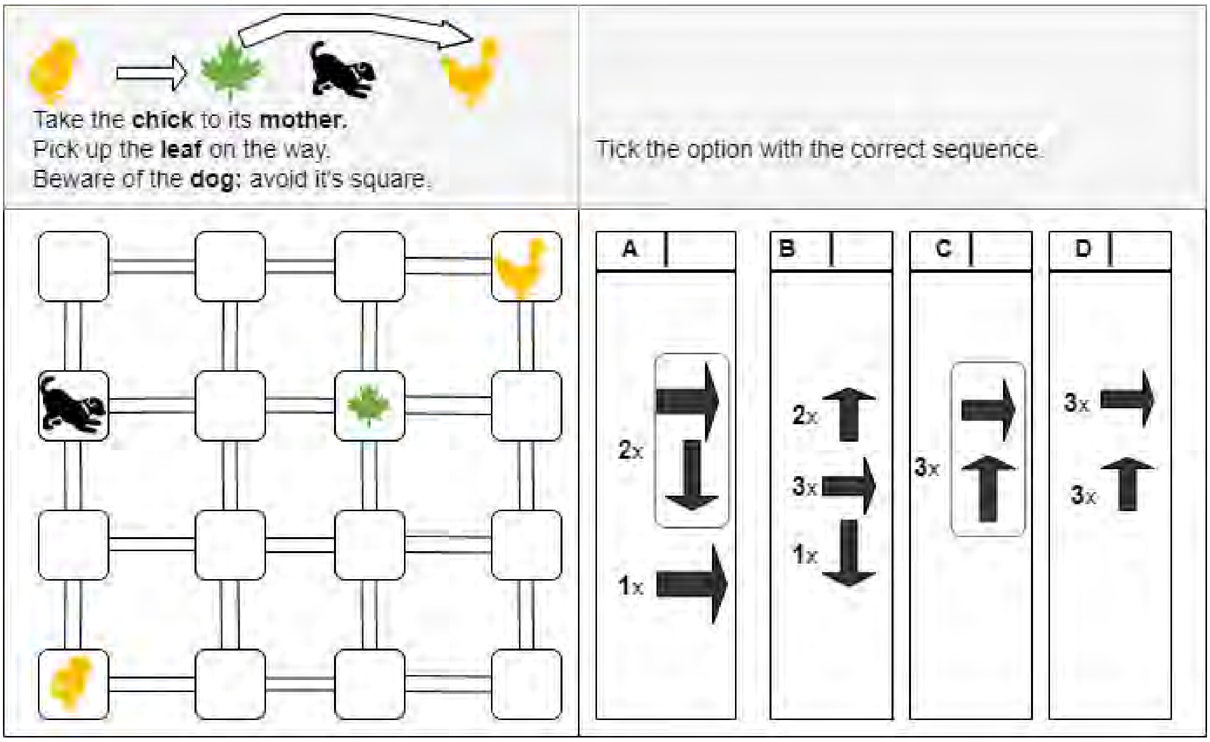
Desired result

D

Computational thinking concept:

- *Sequences*: In programming the concept of "sequences" refers to expressing a task as a series of instructions or steps. These steps are designed for a computer to execute. It is important to note that changing the order of these steps results in varying outcomes or behaviors of the program.

Activity 2



End of Activity 2

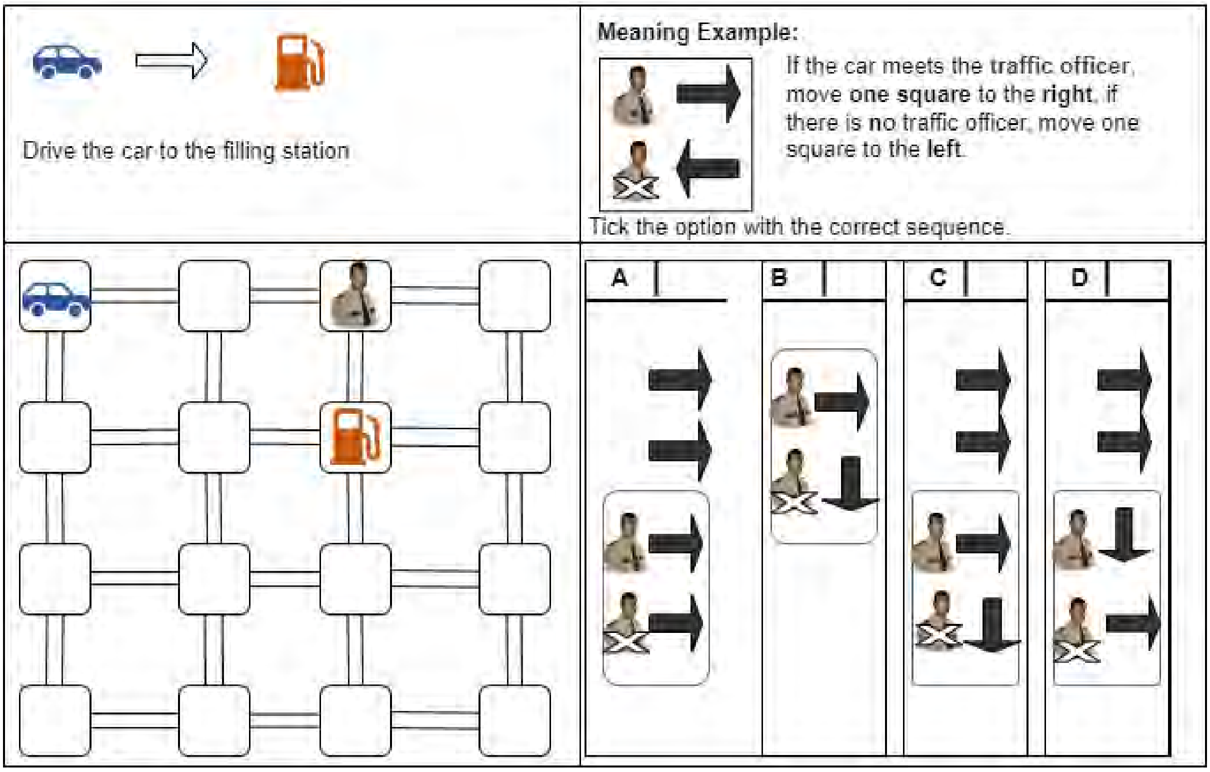
Desired result

C

Computational thinking concept:

- *Loops* are another fundamental concept in programming. They serve as a mechanism for executing a single set of instructions multiple times, whenever necessary. This eliminates the need to repeat the same set of instructions each time an action needs to be reiterated.

Activity 3



End of Activity 3

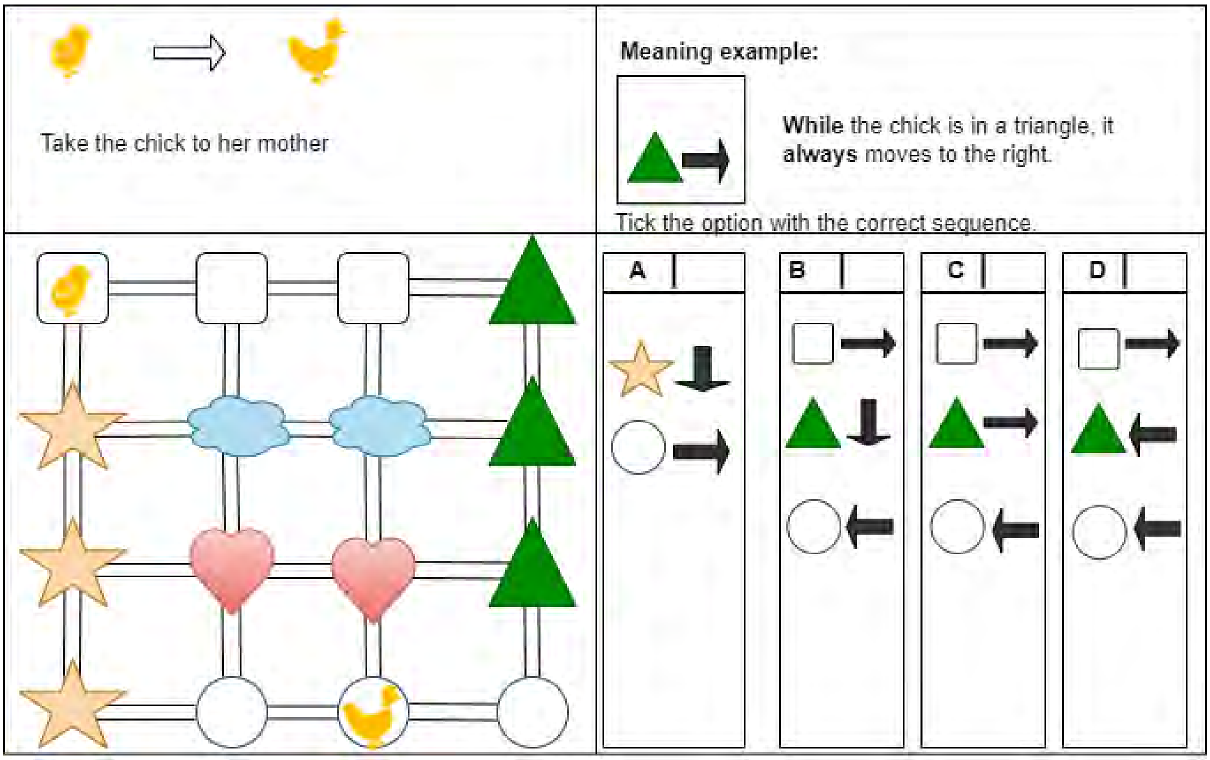
Desired result

D

Computational thinking concept:

- *Conditionals* are a concept in programming that enables computer programs to execute actions based on certain conditions. This concept supports the expression of multiple outcomes.

Activity 4



End of Activity 4

Desired result

B

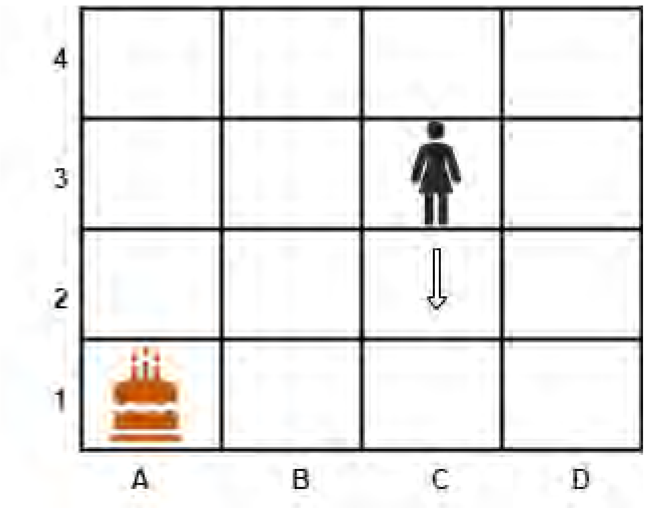
Computational thinking concept:

- Conditionals

Section E: CT Concepts in Visual Programming

Activity 1

1. Jane () is starting to walk at position **C3** facing the direction of the arrow on the grid.



Which instructions will take Jane to the cake () located at position **A1**?

(a)

(b)

End of Activity 1

Desired result:

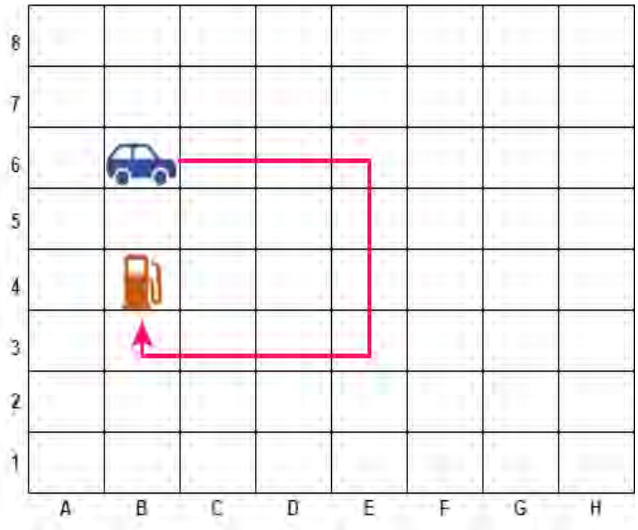
(d)

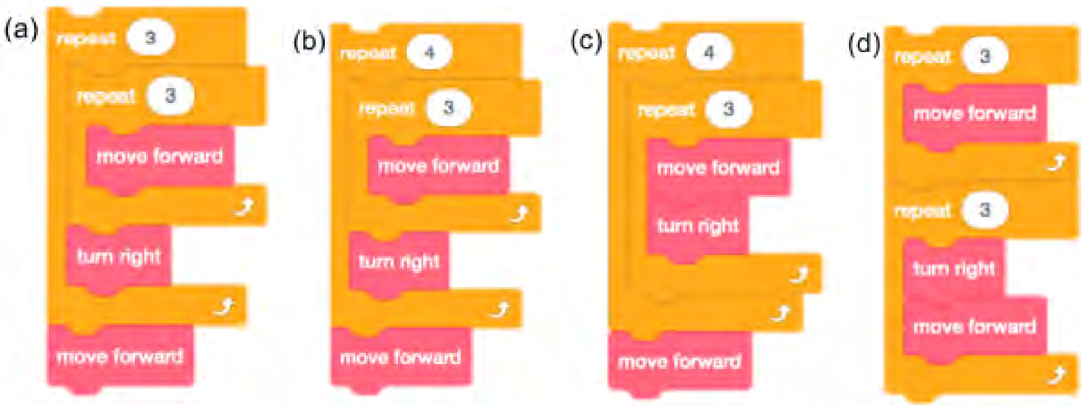
Computational thinking concept:

- Sequences

Activity 2

Which instructions take the car (**B6**) to the filling station (**B4**) using the path marked out?





End of Activity 2

Desired result

(a)

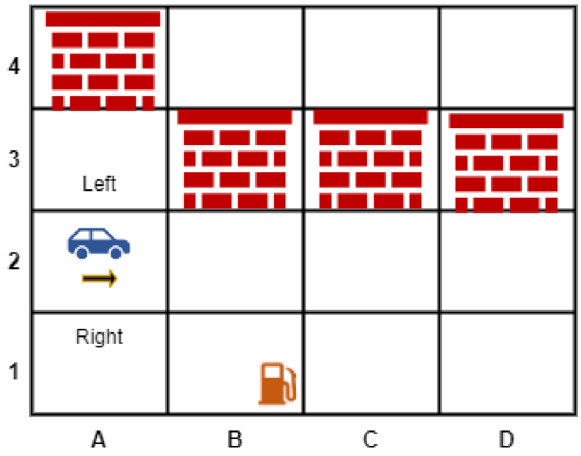
Computational thinking concept

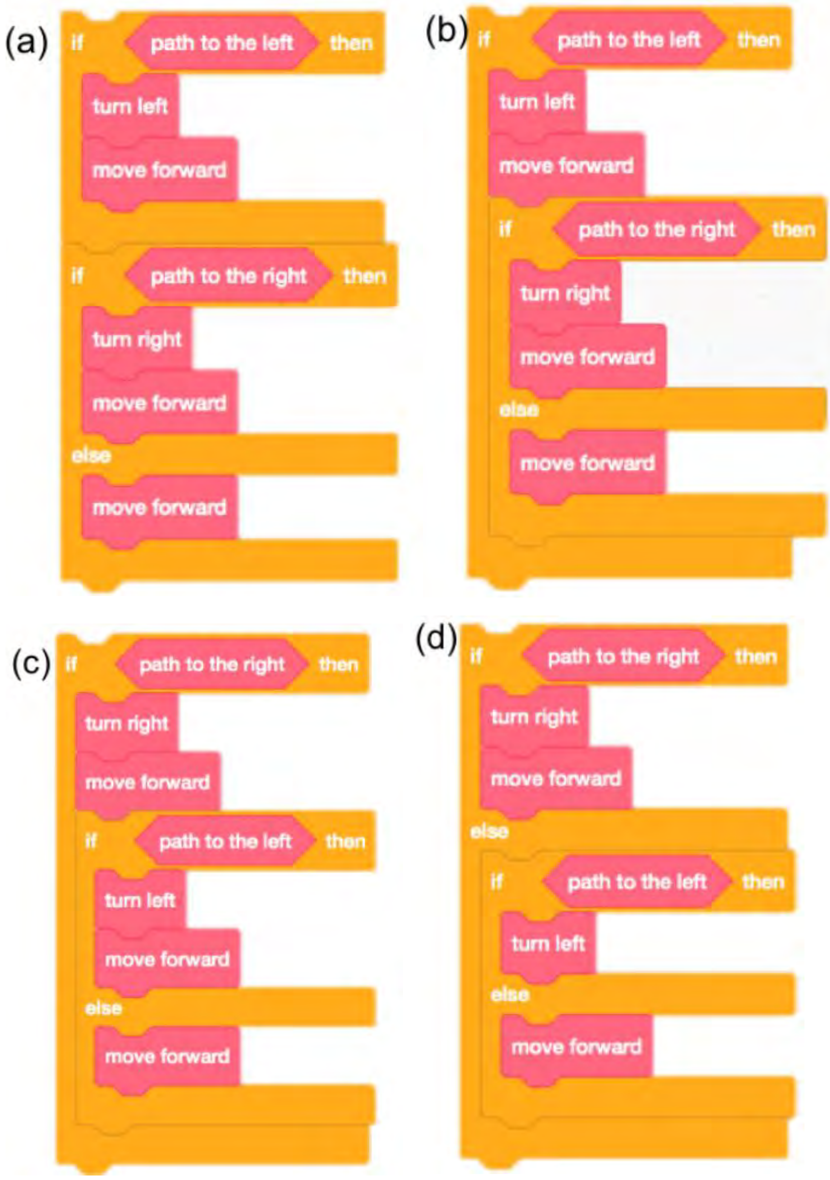
- Loops

Activity 3

A car starts at position **A2** and faces in the right direction as shown by the arrow below the car. The car can never pass through a brick wall or share the same position as a wall.

Which instructions correctly take the car to the filling station (**B1**)?





End of Activity 3

Desired result

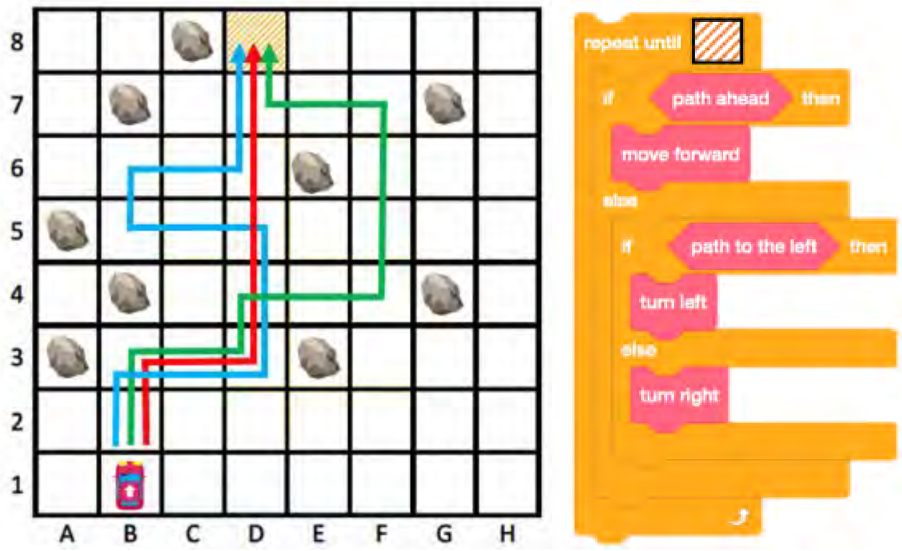
(c)

Computational thinking concept:

- Conditionals (if-else statements)

Activity 4

The following instructions move the car (**B1**) to the orange patterned tile (**D8**) using one of the paths marked out.



According to the instructions, which coloured path will the car take?

A: The red path

B: The blue path

C: The green path

D: None of the above

End of Activity 4

Desired result

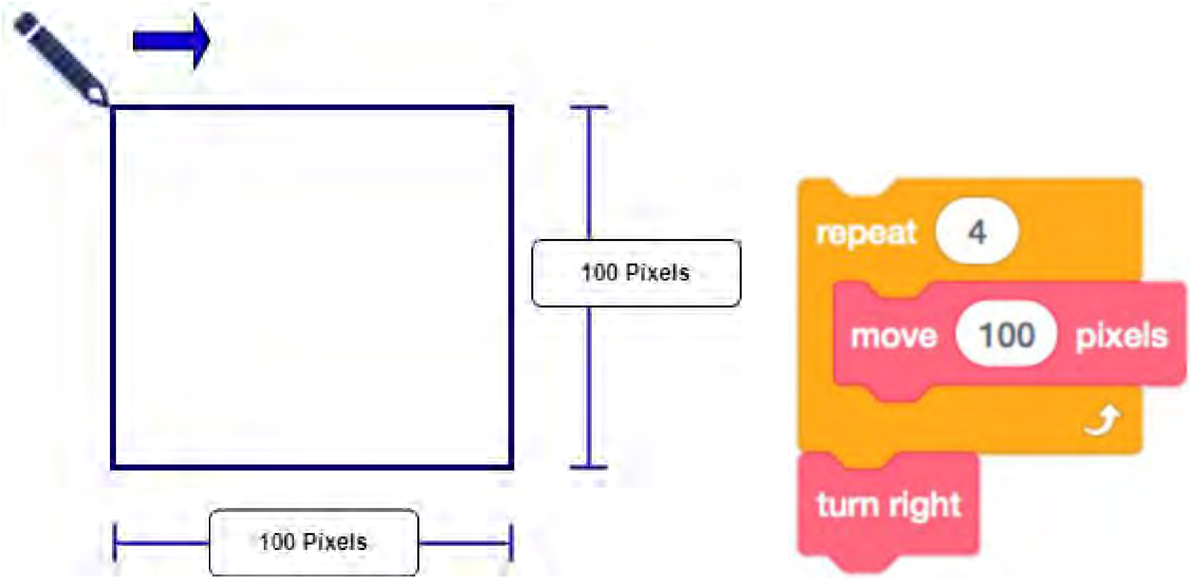
A

Computational thinking concept

- Loops and conditionals

Activity 5

The instructions below should make the pencil draw the square **once**. The pencil starts at the upper left corner position and moves in the direction of the arrow.



What is the **error** in the instructions?

A: The "*repeat" loop* should repeat 2 times.

B. The "*move"* block should be outside the loop.

C. The "*turn right"* block should be inside the loop, after the "*move"* block.

D. The "*turn right"* block should be inside the loop, before the "*move"* block.

End of Activity 5

Desired result

C

Computational practice:

- Debugging

Section F: CT Skills/Practices

Activity 1

The RU football team has been playing a soccer game against the UCT football team.

Here are the goal scorers:

minute 1: Thandi

minute 10: Praise

minute 35: Bernard

minute 47: Mandla

minute 73: Ayanda

minute 89: Richard.

Question 1:

Explain the steps you would undertake to reach the *desired result* as given below for the question:

- Which of the following final scores is not possible, considering that only one team manages to score two goals in a row?

  Options:

  A: 3-3, B: 5-1, C: 2-4, D: 4-2

  Desired result:

  B: 5-1

Question 2:

What information provided in the activity is not necessary for solving the problem in Q1?

End of Activity 1

Proposed Steps:

There are three possible final standings in this game.

- If one team scores first and the other team scores 2 in a row the final standing will be 3-3.
- If the first team scores first and also scores 2 in a row then the final standing will be 4-2.
- If the second team scores first and also scores two in a row then the final standing will be 2-4.

Computational Thinking Skills:

- Algorithmic thinking (*for question 1*)

Question 1 evaluates and enhances participants' algorithmic thinking for problem-solving. To come up with the three possible final game standings, the participants need to follow an algorithm that adheres to the condition that only one team achieves consecutive scoring. The desired result in this activity, which is a '5-1' final game standing, violates this condition. In relation to programming, the solution to this problem is a nested if-statement: The first if-statement checks which team scores first. The second if-statement checks if a team scored 2 goals in a row.

- Decomposition/modularizing (*for question 1*)

Resolving the final game standings for each possible scenario requires the application of the skill of decomposition. It will involve isolating and concentrating on one possible scenario at a time to determine its final game outcome.

- Abstraction (*for question 2*)

The specific minute when the teams scored (e.g., minute 1, 10, etc.), and the names of the goal scorers are not relevant to the question. Instead, this information should be simplified to the knowledge that the teams scored a total of six goals by the end of the game, which must be reflected in the final game standings. The key information that will help determine the final game standings (solve the problem) in this activity is the given condition and the total number of goals scored.

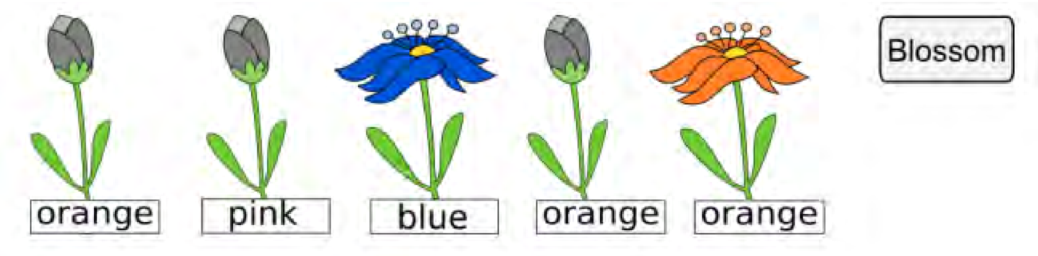
Activity 2

Thabisa is playing a computer game.

First, the computer secretly chooses colors for five buds. The available colors for each flower are **blue**, **orange**, and **pink**. Thabisa has to guess which flower has which color. She makes her first five guesses and presses the “Blossom” button.

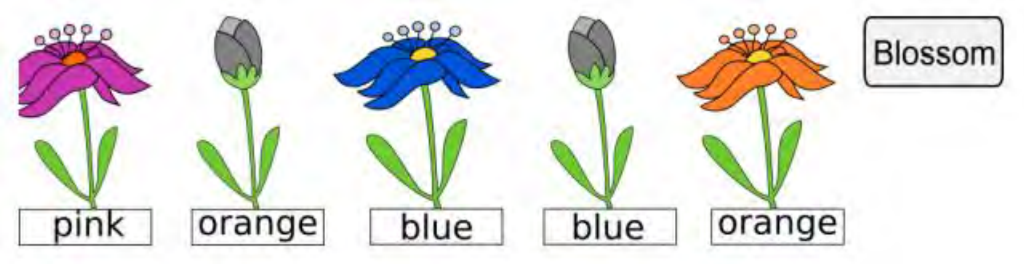
The buds, whose colors she guessed correctly, break into flowers. The others remain as buds.

Thabisa’s first go:



Thabisa then has another go at guessing and presses the “Blossom” button again.

Thabisa’s second go:



Question:

Explain the steps you would undertake to get to the *desired result* as given below for the following question:

- What colours did the computer choose for the flowers?

  Options:

  A: blue pink blue orange orange

  B: pink blue blue blue orange

  C: pink blue blue pink orange

  D: pink pink blue pink orange

  Desired result:

  C: pink blue blue pink orange

End of Activity 2

Proposed steps:

- After 2 guesses, the colors for the first, third, and fifth flowers are known as the flowers blossom. (Rules out answer A)
- The second flower must be blue as there are only 3 colors and both guesses of pink and orange did not make the flower blossom. (Rules out answer D)
- Similarly, the guesses of orange and blue for the fourth flower did not make the flower blossom, so it must be pink. (Rules out answer B)

Computational Thinking Skill:

- Pattern recognition/knowledge transfer

Leveraging prior experience and knowledge (e.g., events that happened and did not happen) is an important skill for solving many kinds of problems. After each guess the player gets complete information about all the flowers. If in each guess the player chooses a different color for the not-yet-blossomed flowers, then in the third guess he/she can always correctly pick the colors of all the flowers.

Activity 3

1. A rat is at the entrance of a tube system. It wants to reach the cheese at the end of the tube. The rat always follows these commands:

   1. Go downwards until a crossing
   2. At the crossing, move through to the next vertical tube
   3. Go to command 1

Question:

Explain the steps you would undertake to determine the *desired result* as given below for the question:

- In which tube should the rat start so that it reaches the cheese?

  Options:

  A: 1, B: 2, C: 3, D: 4, E: 5

  Desired result

  C: 3

End of Activity 3

Proposed steps:

Following the given commands,

- From tube 1, the rat always reaches tube 3.
- From tube 2 it reaches tube 1.
- From tube 4 it reaches tube 2.
- From tube 5 it gets to tube 4.

Computational thinking skill:

- Algorithmic thinking

Many computer programs are programmed so that they have to follow the exact commands. This rat does the same thing: It follows the commands 'go downwards' and 'change directions at the next crossing' over and over again. These kinds of commands depend on the tube entrance as to which way the rat runs in the tube system. Most computer programs are deterministic: if you always input the same data, the program performs the same calculations and delivers the same output.

Activity 4

To arrange a dinner party Nomusa needs to talk to five friends:

Alicia, Beat, Caroline, David, and Emil

Nomusa can talk to Emil right away. However, to talk to her other friends, there are a few points to consider:

1 – Before she talks to David, she must talk to Alicia.

2 – Before she talks to Beat, she must first talk to Emil.

3 – Before she talks to Caroline, she must first talk to Beat and David.

4 – Before she talks to Alicia, she must first talk to Beat and Emil.

Question:

Explain the steps you would take to determine the desired sequence as shown below for the following question:

In what order should Nomusa talk to all her friends if she wants to talk to all of them?

Options:

A: Emil ⟶ Beat ⟶ Alicia ⟶ David ⟶ Caroline

B: Emil ⟶ Alicia ⟶ Beat ⟶ David ⟶ Caroline

C: Emil ⟶ Alicia ⟶ Beat ⟶ Caroline ⟶ David

D: Emil ⟶ Beat ⟶ David ⟶ Alicia ⟶ Caroline

Desired sequence:

A: Emil ⟶ Beat ⟶ Alicia ⟶ David ⟶ Caroline

End of Activity 4

Proposed steps:

This task has to do with dependencies (sequences).

- Emil is the only friend with no dependencies so he must go first.
- Beat only depends on Emil so he can go second.
- Alicia depends on Beat and Emil, so she can go third.
- David depends on Alicia and David so she can go last.
- Finally, Caroline depends on Alicia and David so she can go last.

Computational Thinking Skill:

- Sequences (CT Concept)
- Decomposition and Algorithmic thinking

Appendix A2: Semi-Structured Interviews and Focus Group Discussion Guide

1. What do you think Computational Thinking might mean?

   …………………………………………………………………………………………

   …………………………………………………………………………………………

   …………………………………………………………………………………………

2. Do you think visual programming contributed to your understanding of computational thinking? If yes, In what ways? Give examples.

   …………………………………………………………………………………………

   …………………………………………………………………………………………

   …………………………………………………………………………………………

3. How would you describe your experience of learning and applying computational thinking skills during the intervention? Give examples.

   …………………………………………………………………………………………

   …………………………………………………………………………………………

   …………………………………………………………………………………………

4. If yes to Q2; What components of the visual programming module do you feel contributed most to your development of computational thinking skills?

   …………………………………………………………………………………………

   …………………………………………………………………………………………

5. Were there any aspects of the coding in visual programming that you found difficult or frustrating in relation to learning and developing computational thinking skills?

   …………………………………………………………………………………………

   …………………………………………………………………………………………

6. If yes to Q2; Can you identify any factors that facilitated your understanding and application of computational thinking during this intervention? Give examples.

   …………………………………………………………………………………………

   …………………………………………………………………………………………

   …………………………………………………………………………………………

7. Can you describe any obstacles or challenges you faced in developing your computational thinking skills during this intervention?

   …………………………………………………………………………………………

   …………………………………………………………………………………………

   …………………………………………………………………………………………

8. What are your views and experiences about using a visual programming tool like Scratch to learn and develop computational thinking skills?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

9. Were there any specific instructional strategies or elements within the intervention that you found particularly engaging or effective?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

10. How do you feel your own views or perceptions of computational thinking developed during the intervention?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

11. Given your experiences, what recommendations would you offer for enhancing the development of computational thinking skills in pre-service teacher training using visual programming?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

12. Given your experiences, what recommendations would you offer for enhancing the development of computational thinking skills in pre-service teacher training?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

13. How do you see yourself transferring the CT skills learned during the training into real-life situations i.e teaching?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

14. Can you suggest any improvements or modifications to the training that you believe would enhance the development of CT skills?

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Appendix A3: Participants' reflective journal questions

1. Describe your experience during today's session. What stood out the most for you (if there is any)?

   …………………………………………………………………………………………
   …………………………………………………………………………………………
   …………………………………………………………………………………………
   …………………………………………………………………………………………
2. What CT skills or knowledge did you gain in the session?

   …………………………………………………………………………………………
   …………………………………………………………………………………………
   ………………………………………………………………………………………..
3. Were there any difficulties or challenges you faced during the session? If so, describe them.

   …………………………………………………………………………………………
   …………………………………………………………………………………………
4. If yes to Q3; How did you overcome these challenges, or how do you plan to in the future?

   …………………………………………………………………………………………
   …………………………………………………………………………………………
   …………………………………………………………………………………………
5. How have you seen your understanding of CT and visual programming evolve throughout today's session?

   …………………………………………………………………………………………
   …………………………………………………………………………………………
   …………………………………………………………………………………………
6. What were the enabling factors that helped you learn towards developing CT in today's session?

   …………………………………………………………………………………………
   …………………………………………………………………………………………
   …………………………………………………………………………………………
7. How did the interactions between peers and the lecturer impact your learning experience?

………………………………………………………………………………………
………………………………………………………………………………………
………………………………………………………………………………………

8. How do you feel the session's activities can improve your CT skills and be applied to real-life situations?

………………………………………………………………………………………
………………………………………………………………………………………
………………………………………………………………………………………

9. In what ways do you think you can apply what you've learned today, especially in a classroom setting?

………………………………………………………………………………………
………………………………………………………………………………………
………………………………………………………………………………………

10. Share any other thoughts, feelings, or reflections related to this session that have not been covered by the previous questions.

………………………………………………………………………………………
………………………………………………………………………………………
………………………………………………………………………………………
………………………………………………………………………………………

Appendix A4: Computer Literacy Skill Set Questionnaire

1. Did you do computer science/studies in high/secondary school?
   - Yes
   - No
2. Do you have a smartphone?
   - Yes
   - No
3. Do you have a computer?
   - Yes
   - No
4. Which social media account do you have?
   - Facebook
   - WhatsApp
   - Twitter
   - Instagram
   - None
   - Other, specify
5. Do you know how to? (circle the option)
   - Turn on/off a computer
   - Type on a computer keyboard
   - Open applications/programs on a computer
   - Play music or videos on a computer
   - Use Microsoft Word
   - Use Microsoft Excel
   - Use Microsoft PowerPoint
6. Please rate yourself on each of the skills and knowledge below on a scale of 1 to 5. (1 = Not at all 2 = Somewhat 3 = Moderate 4= Enough 5 = Excellent)
   - Typing on computer keyboard
   - Logging in
   - Creating files/folders
   - Working with databases
   - Software installation
   - Network design and installation
   - Computer programming

Appendix B1: Letter to Pre-service Teacher Participants

Dear Sir/Madam

Re: Participation in research on working with pre-service teachers (PSTs) to develop computational thinking at a South African University

I kindly request your voluntary participation in my research project. The study will be conducted from September to October 2023 and aims to collaborate with you in understanding the development of computational thinking using visual programming approach in a pre-service teacher training program at a South African university. The study comprises four phases: Phase one involves working on computational thinking reflective activities for all consenting BEd students. Phase two involves writing reflective journals during the intervention. Phase three will involve semi-structured individual interviews with twelve selected students based on their willingness. Phase three will involve two focus group discussions with the twelve selected participants. The last phase will involve revisiting the computational thinking reflective activities for all the consenting BEd students.

Participation in this research study is entirely voluntary, and you have the right to withdraw your consent at any time. If you choose to participate and agree to be interviewed and take part in the focus group interview, I kindly request your permission to audio-record the interview and focus group discussion sessions for later data analysis. Please be assured that your identity and opinions will be treated with the utmost anonymity. The data collected will be used solely for the purpose of this study.

By participating in this study, you will have the opportunity to share your experiences and knowledge with other BEd students who have been selected. This collaboration will contribute to designing and developing an introduction to technology curriculum relevant to the BEd programme and responsive to the new primary school curriculum. The curriculum can then be implemented across various programs, including BEd and other teacher education programs.

The research activities in this study will also prepare you, as a pre-service teacher, to teach the newly introduced subject of coding and robotics in your future classrooms. Additionally, your involvement in this study will allow you, as an educator to reflect on computational thinking and its effective integration in teaching and learning.

It is important to note that this research project has obtained approval from both the Rhodes University Ethical Standards Committee and the Education Department Higher Degrees Committee. If you have any concerns regarding the research, you may address them to the Ethics Office at Rhodes University via ethics-committee@ru.ac.za or by calling +27 (0) 46 603 7727. If you have any questions about the research, please do not hesitate to contact me, Mr. Chikondi Sepula, at +27717368060 or csepula@unima.ac.mw.

Consent: I am aware that

- I will voluntarily participate in the research stated above.
- I am willing to be interviewed and make time for it.
- I am willing to participate in focus group discussions and make time for it.
- I am free to withdraw from this study at any time without facing any negative or undesirable consequences.
- I am also aware that the information provided by me will be strictly confidential and the findings will be reviewed to support the redesign and development of the Introduction to Technology curriculum.
- My identity in this study will be protected by the code of ethics stipulated by Rhodes University
- Having taken note of the above information, I freely volunteer to take part in the research process and acknowledge that I have not been forced to do so.

Declaration

I…………………………………………………. (Full name and surname of participant) hereby confirm that I understand the contents of this letter and the nature of the research project. I consent to participate in the research project.

Signature of participant …………………………………………. Date……………………….

Yours Sincerely

Mr. Chikondi Sepula

Appendix B2: Letter to the Registrar

RHODES UNIVERSITY
Where leaders learn

ACCESS LETTER REQUESTING PERMISSION TO CONDUCT RESEARCH

Department of Secondary and Post-School Education
Donkin Street
Grahamstown 6139
26-July-2023

The Registrar

Rhodes University

Dear Madam

REQUEST FOR PERMISSION TO CONDUCT RESEARCH WITH BACHELOR OF EDUCATION STUDENTS

This letter is a formal request to conduct research with first-year Bachelor of Education (BEd) students at Rhodes University. The research project aims to *work with pre-service teachers to develop their computational thinking using visual programming approach*. I, Mr. Chikondi Sepula, a Master of Education candidate and a parti-time lecturer from the Department of Primary and Early Childhood Education at Rhodes University will oversee and conduct the research project. The study is scheduled to take place from September to October, 2023.The research is being supervised by Dr Clement Simuja.

The research project will involve requesting voluntary participation from students to engage in computational thinking reflective activities and write reflective journals. Additionally, as part of the research activities, we will conduct individual interviews and two focus group discussions with twelve selected students. These interviews will enable the selected participants to share their experiences as pre-service teachers and their involvement in an introduction to technology course. While the participation of Pre-Service Teachers (PSTs) in the research activities related to this project, such as interviews, reflective journals, focus group discussions (FGDs), and CT assessment tools, is voluntary, it is not connected to the accumulation of credits or marks for the module. The research activities are separate from the module assignment that all BEd students must complete.

The participants will benefit from this research as their involvement may prepare them to teach the newly introduced primary school subject of coding and robotics in their future classrooms. Furthermore, the pre-service teachers' contributions in this study will aid in the design and development of a relevant and responsive introduction to technology curriculum, which can be implemented across the curriculum in BEd and other teacher education programs. Additionally, as a teacher educator, the research findings will allow me and my colleagues to reflect on our BEd teaching practices and explore how coding and computational thinking can be effectively integrated across all BEd courses.

Upon the completion of the study, I commit to providing the Registrar's Office at Rhodes University with an electronic copy of the research publications. It is important to note that this research has obtained approval from both the Faculty of Education Ethics Committee and the Education Department Higher Degrees Committee. Should any concerns arise during the research process, they can be addressed to the Ethics Coordinator of the Education Faculty, Dr. Pamela, via email at p.vale@ru.ac.za.

If you have any further questions or require additional information regarding the research, please do not hesitate to contact me directly at 22S6551@campus.ru.ac.za

Yours sincerely,

Mr. Chikondi Sepula

Appendix C1: Research Proposal Approval Letter

RHODES UNIVERSITY
Where leaders learn

www.ru.ac.za

Education Higher Degrees Committee
Chair
Education Department, Grey Street, Grahamstown/Makhanda, 6139, South Africa
PO Box 94, Grahamstown/Makhanda, 6140, South Africa
t: +27 (0) 46 603 8315
e: dean.education@ru.ac.za

15/09/2023

C Sepula

Dr Clement Simuja

7431

Education Department

Dear C Sepula

Re: Exploring the Development of Computational Thinking Skills among Pre-Service Teachers through Visual Programming: An Interventionist Case Study -7431 Aug 2023

This letter confirms that the Higher Degree Committee has approved your research proposal, for the degree of Education low/med risk in Education in the Faculty of Education titled: Exploring the Development of Computational Thinking Skills among Pre-Service Teachers through Visual Programming: An Interventionist Case Study

You will need ethical clearance from the Education Faculty Ethics Committee before you can begin your research. The ethics review process is underway and you will receive a separate outcome notification from the Chair: Education Faculty Ethics Committee.

Sincerely,

MBlackie

Prof Margaret Blackie

Chair: Education Higher Degrees Committee

Associate Professor: Centre for Higher Education Research, Teaching and Learning

cc. Mr Chikondi Sepula

Appendix C2: Ethics Clearance Letter

RHODES UNIVERSITY
Where leaders learn

Rhodes University, Education Faculty
Research Ethics Committee
PO Box 94, Makhanda, 6140, South Africa
Tel: +27 (0) 46 603 8393
Fax: +27 (0) 46 603 8028
email: e.rosenberg@ru.ac.za

https://www.ru.ac.za/researchgateway/ethics/

19 September 2023

Dr Clement Simuja

Education Department

C.Simuja@ru.ac.za

Dear Dr Clement Simuja and Mr Chikondi Sepula

Re: Exploring the Development of Computational Thinking Skills among Pre-Service Teachers through Visual Programming: An Interventionist Case Study

APPLICATION NUMBER: 2023-7431-8062

This letter confirms that your research ethics application has been reviewed and **APPROVED** by the Education Faculty Research Ethics Committee (EF-REC). Your permission letter(s) where applicable have been received and you are free to proceed with your study.

Approval is granted for 1 year. An annual progress report is required in order to renew approval for an additional period. You will receive an email notifying you when the progress report is due.

Should any substantive change(s) be made during the research process, that may have ethical implications, you should notify the Education Faculty REC Chair via email. This includes changes in investigators. The REC Chair will advise as to whether a new application is necessary.

Do keep this clearance letter secure and accessible throughout your study and after its completion. It will be needed when a thesis is examined and when publications are submitted to journals.

Please also submit a brief report to the REC Chair on the completion of the research. This can be done via email. The purpose of this report is to indicate whether the research was conducted successfully and whether any ethics-related matters arose that the committee should be aware of, in order to guide future studies.

Sincerely,

Prof Eureta Rosenberg

Chair: Education Faculty Research Ethics Committee