The African Journal of Information Systems

Volume 15 | Issue 3 Article 2

October 2023

A Framework for Teaching Computational Thinking in Primary Schools: A Namibian Case Study

Maria M. Ausiku
University of Pretoria, mmausiku@gmail.com

Machdel C. Matthee
University of Pretoria, machdel.matthee@up.ac.za

Recommended Citation
Ausiku, Maria M. and Matthee, Machdel C. (2023) "A Framework for Teaching Computational Thinking in Primary Schools: A Namibian Case Study," *The African Journal of Information Systems*: Vol. 15: Iss. 3, Article 2.
Available at: https://digitalcommons.kennesaw.edu/ajis/vol15/iss3/2

The African Journal of Information Systems

A Framework for Teaching Computational Thinking in Primary Schools: A Namibian Case Study

Research Paper

Volume 15, Issue 3, October 2023, ISSN 1936-0282

Maria M. Ausiku
Department of Informatics
University of Pretoria
mmausiku@gmail.com

Machdel C. Matthee
Department of Informatics
University of Pretoria
machdel.matthee@up.ac.za

(*Received February 2023, accepted August 2023*)

ABSTRACT

Several professional development programs have been designed to train in-service teachers on a computational thinking (CT) curriculum, but few researchers have examined how these affect primary school teachers' self-efficacy and knowledge of CT in emerging economies. This study's objective was to create a framework for the professional development of primary school in-service teachers for the teaching of CT (referred to as professional development for primary computational thinking - PD4PCT) to be integrated into teachers' professional development programs. An initial framework was refined after implementing it at a Namibian school with a group of 14 teachers from five different disciplines (social studies, English, natural science, mathematics, and Afrikaans). Literature reviews, pre- and post-intervention questionnaires, semi-structured interviews, and self-reporting diaries were used to collect data. The framework was evaluated by experts via an online questionnaire. The findings show that teachers who participated in the professional development program improved their perceived CT knowledge, beliefs, and confidence to teach CT.

Keywords: Computational thinking, professional development, primary school teachers, unplugged, programming, participatory design, constructionism.

INTRODUCTION

Computational thinking (CT) is considered an important skill for every student in the 21st century (Karakasis & Xinogalos, 2020; Li et al., 2020; Wing, 2006). It can be understood as an approach to problem-solving that applies to many disciplines and draws on constructs fundamental to Computer Science such as decomposition, abstraction, pattern matching, and algorithmic design (including logical thinking) (Karakasis & Xinogalos, 2020). Some countries have included CT as a school subject, while others have embedded it in existing subject choices such as mathematics and science. A key question is how best to prepare and support teachers to include CT into their teaching practices (Bocconi et al., 2016). Several professional development efforts have been conducted to help teachers implement CT (see for example Gadanidis et al. (2018) and Weintrop et al. (2020)) and studies have confirmed that there is a need to upskill teachers and give specific guidance. However, research on how in-service teachers understand and adapt CT in their teaching practice is limited especially in developing economies' contexts like Namibia.

Therefore, the issue of how to design learning and teaching processes gains importance in the development of CT skills. For students to acquire this skill, teachers should improve their own competencies in designing and applying learning and teaching processes. Unfortunately, it appears that the competencies of in-service and pre-service teachers are not at the desired level. CT has the lowest average competency score among Spanish pre-service teachers' digital skills (Esteve-Mon et al., 2020). Alfayez and Lambert (2019) found that of 55 information technology teachers most have a low comprehension of CT and some have wrong ideas about the nature of CT. The pre-service teachers stated that they had an interest in CT and they wanted to participate in professional development programs related to CT, but they had negative attitudes towards integration of CT into lessons because they had difficulty in understanding this skill (Fessakis & Prantsoudi, 2019). Many teachers lack confidence in their abilities to teach CT (Sentance & Csizmadia, 2017), and many do not have a background in computer science or they feel that teacher development resources are insufficient (Yadav et al., 2016). Some teachers may think that CT is simply about the use of digital technology (Sands et al., 2018). Also, integration of CT into the subject area is challenging, as teachers must not only know the subject they teach but also understand Computer Science concepts (Kale et al., 2018).

Thus, providing effective professional development to teachers is essential if CT is to be implemented successfully in K-12 education (Kong et al., 2020; Mason & Rich, 2019; Mouza et al., 2017). To become a capable CT teacher, one needs to know what CT is, its educational goals, and how to teach it (Saeli et al., 2011).

This study sets out to answer the following research question: *What are the components of a framework for the professional development of primary school in-service teachers for the teaching of CT?* This paper shows how this question was answered by describing the iterative development, implementation, and validation of a framework for professional development for primary computational thinking (PD4PCT). What follows is a review of the existing literature and the methodology employed in this study. A detailed description of the development of the framework, the design of a teacher development program, its implementation based on the proposed framework, and the validation of the framework follow. Then, we present the evaluation results based on the feedback from the teachers and experts who participated in the program and discuss the implications of the results.

LITERATURE REVIEW

Definition of CT

Wing (2006) described CT as, "Solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental of Computer Science" (Wing, 2006, p.33). Different definitions of CT exist but the one of Selby and Woollard (2014) is particularly useful. They describe CT as a problem-solving method that incorporates thinking processes and uses decomposition, algorithms, abstractions, evaluation, and pattern recognition. The CT practices used in acquiring CT skills have been identified as tinkering, creating, persevering and collaboration (Csizmadia et al., 2015).

While there is no systematic description in the literature, researchers generally agree that CT skills include algorithmic reasoning, exploring different levels of abstraction, breaking problems down into small parts, and presenting knowledge through models (Barr & Stephenson, 2011; Grover & Pea, 2013).

Preparing Teachers for Teaching CT

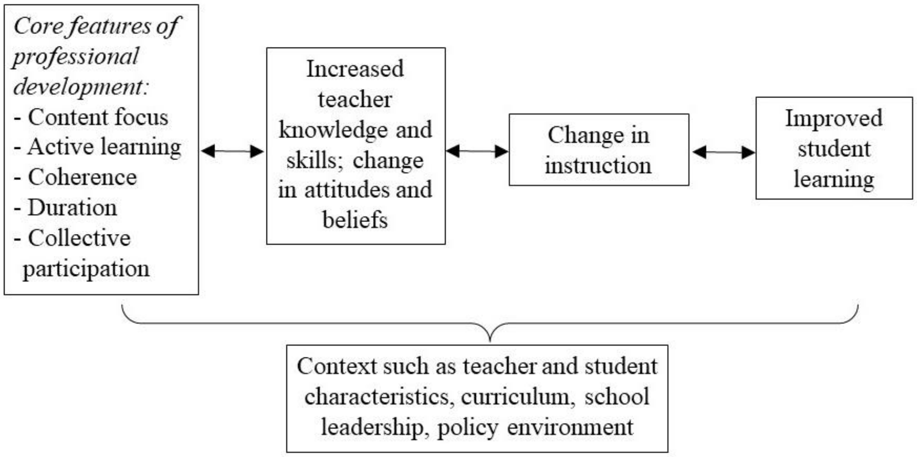
Some research exists which examines how to teach CT concepts to pre-service teachers (Alqahtani et al., 2021; Bower et al., 2017; Sands et al., 2018; Yadav et al., 2016). It is clear that in-service teacher professional development plans must incorporate CT, while educators of teachers must find opportunities to incorporate CT instruction into pre-service classes in order to best prepare pre-service teachers (Angeli & Giannakos, 2020). To further improve CT teaching and learning, teachers must be trained in far-reaching ways as to how to design CT learning exercises, how to teach CT, how to assess CT, and how to use technology to teach CT concepts (Boulden et al., 2021).

Ausiku and Matthee (2021) conducted a systematic review and show that common teaching strategies used to upskill teachers in CT, are unplugged activities, programming, robotics, game-based learning and project-based learning.

The focus of this paper is to present a professional development (PD) framework for an emerging economy context. Various PD frameworks exist which take context into account (see for example Desimone (2009) and Koehler and Mishra (2009)). The one that will be discussed here is the framework developed by Desimone (2009) (see Figure 1) According to this framework, to succeed, teacher PD should exhibit the following essential characteristics: subject emphasis, active learning, coherence, suitable duration, and collective involvement (Desimone, 2009). PD with a strong curriculum emphasis has been shown to positively affect teacher performance compared to PD that lacks a strong subject matter focus, while active learning is more effective than passive learning for teacher career development (Desimone, 2009). Another critical feature is coherence and Desimone (2009) identified two critical components of coherence: first, instruction must align with teachers' experience and beliefs; and second, PD material must align with classroom, regional, and national policy. According to research, intellectual and pedagogical reform requires PD programmes to be sufficiently lengthy, both in terms of the time period covered by the practice and the number of hours expended on the activity (Desimone, 2009). Collective participation is important because PD is successful when teachers from the same school, grade, or department collaborate. Another feature is the importance of context and the need for teacher educators to make PD compatible with the environments in which teachers work (Desimone, 2009).

Figure 1

Core Conceptual Framework for Studying the Effects of Professional Development of Teachers



Note. Adapted from Desimone (2009)

Existing PD Frameworks for CT

Apart from general PD frameworks (as discussed above), a small number of PD frameworks exist with a focus on the training of teachers in developed countries to integrate CT into their classrooms. Three frameworks namely, activities, demonstrations, application, pre-activation, transparency, theory, exemplification, and reflection (ADAPTTER), computational thinking teacher development (CTTD) and code, connect, and create (3C) are discussed below.

ADAPTTER Framework

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

CTTD Framework

In another recent study, a framework for primary teacher development for CT was developed by Kong and Lai (2021). This framework was created and tested in Hong Kong at primary schools and emphasizes the four content-related aspects of Mishra and Koehler (2006)'s technological pedagogical content knowledge model. Technological content knowledge focuses on learning to program using a block-based programming environment. Knowledge of CT principles, practices, and views is the subject of content knowledge. Pedagogical content knowledge focuses on CT pedagogies that do not depend on the use of programming environments. The technological pedagogical content knowledge model emphasizes the integration of technology, pedagogy, and the content of CT in context. Based on these factors, a seven-step lesson structure for learning to teach a unit of curricular material was suggested (Kong & Lai, 2021).

3C PD Framework

Jocius et al. (2020) created the 3C PD framework to aid American middle (ages 11-13) and high (ages 14-18) school content-area teachers in introducing CT into their classes. Three main elements make up the suggested 3C PD model: code (bootcamp), connect (tying discipline content and pedagogy to CT), and create (the development of CT-infused learning segments). The 3C model is an integral part of a three-year research project, Infusing Computing, which was intended to describe how middle and high school teachers construct and deliver interdisciplinary, CT-infused curricula (Jocius et al., 2020). The research and facilitation team, which comprised computer scientists, education faculty members who have taught in the classroom, in-service computer science (CS) teachers, and in-service topic area teachers, drew on their knowledge and expertise to establish the 3C design and development process (Jocius et al., 2020).

Although PD frameworks for CT already exist, they have been tailored and implemented only in the context of developed nations. Therefore, the research reported on here endeavors to address this by creating a new framework specifically designed to suit the context of developing countries.

Learning Theories in the PD of Teachers for CT

The situated learning paradigm is based on cognitive theories and social psychology, which emphasize the importance of context-specific social engagement, learning communities, and authentic learning (Brown et al., 1989). According to the situated learning paradigm, learning occurs because of behavior in social interactions within an engaging and shared environment (Brown et al., 1989; Lave & Wenger, 1991; Takahashi, 2011). Closely related is constructionism, which shares constructivism’s connotation of learning as constructing information but emphasizes that learning is enabled by the creation of observable artefacts or items that can then be exchanged and explored with others (Papert & Harel, 1991). Constructionism views learners as active constructors of their own experience and believes that people learn more effectively while constructing individually significant artefacts, “whether it’s a sandcastle on the beach or a theory of the universe” (Papert & Harel, 1991, p. 1). Both these learning theories have been used with success in teacher CT training (see Ozturk et al. (2018) for an application of situated learning paradigm, and the use of constructionism in teacher CT training by Cetin (2016) and Marcelino et al. (2018)).

Participatory Design

The essence of participatory design is to empower users and foster communication and collaboration between designers and users. It makes a concerted effort to involve all stakeholders actively in the design process to ensure that the product/outcome satisfies all the stakeholders' needs and expectations. It is more concerned with the design process and procedures than with the design’s appropriateness and perfection (Schuler & Namioka, 1993). The participatory design approach in teacher development recognizes teachers as critical agents of educational transformation, repositioning them from information transmitters to creators of students’ learning (Mor et al., 2012). Participatory design enables teachers as design partners to create instructional content that is compatible with their students’ and their own teaching needs (Tuhkala, 2019).

METHODOLOGY

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

Phase 1

A case study is an empirical investigation that examines a contemporary phenomenon in its real-world setting, particularly when the distinction between phenomenon and environment is not readily apparent (Yin, 2009). One strength of case studies is their ability to investigate a phenomenon in its context; hence, case studies are a helpful method for examining the world (Rowley, 2002).

The first phase of the case study was conducted during November 2021 and involved 14 teachers from one primary school in Namibia who taught English, Afrikaans, mathematics, natural science and social studies at senior primary phase (Grades 4-7). The researcher was aided in identifying potential participants for the study by the school principal and heads of departments. In this study, the researcher mainly used qualitative methods to collect primary data through self-reporting journals and semi-structured interviews, but also used quantitative methods to collect some data through pre- and post-intervention questionnaires. Secondary data was collected through a literature review to inform the initial development of the framework.

Participatory design was used as a methodological approach for the intervention, which was the PD program. For the participatory design workshop, teachers were divided into five groups based on the subject taught. This resulted in a group for social studies, natural science, mathematics, English and Afrikaans language. Some groups consisted of fewer teachers than others.

Qualitative data from interviews were voice-recorded and transcribed. The data collected for the qualitative element's analysis were entered into ATLAS.ti software, and then coded and analyzed thematically. When attempting to comprehend a group of experiences, thoughts, or behaviors evident across a collection of data, thematic analysis is a suitable and effective technique (Braun & Clarke, 2012).

The quantitative data of pre- and post-intervention questionnaires were analyzed using SPSS software. Because the questionnaires' response data did not fulfil the assumptions of normality of popular parametric tests, such as a t-test, it was appropriate to employ non-parametric tests for hypothesis testing (Field et al., 2012). The Wilcoxon signed-rank test is a dependent non-parametric test (Field et al., 2012). For the quantitative analysis of questionnaires' responses, a non-parametric test, the Wilcoxon signed-rank test, was used to compare teachers' responses before and after participating in the training workshop.

To acquire a more comprehensive understanding of the data, the findings from the quantitative and qualitative analyses were combined.

Phase 2

The validation phase of the framework used the survey method to collect responses from expert reviewers. Purposive sampling was used in the study's second phase to pick study participants for the framework validation. Five experts were carefully chosen to obtain a variety of viewpoints and were persons who work in the industry or education sector and have important roles in the concerned organizations as recommended by Bogner et al. (2009).

Table 1 shows the details of the reviewers and the rationale behind choosing them.

Table 1

Validation Expert Reviewers

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

Note. CT = computational thinking; ICT = information and communication technology.

The validation process for the framework involved presenting it to five experts from the education sector and information and communication technology (ICT) industry for assessment. These reviewers were not involved in the study's initial data gathering. An online questionnaire (with four open-ended questions) was used to collect qualitative data from these experts using Google Forms. A PowerPoint presentation was prepared for the reviewers to give them the background of the study, an overview of CT, and to explain the framework, so they could answer the open-ended questions in the questionnaire. The collected data was thematically analyzed as per Braun and Clarke (2012), using ATLAS.ti. Findings from these themes led to the refinement of the initial PD4PCT framework.

DEVELOPMENT OF THE PROPOSED FRAMEWORK

The proposed framework is mostly based on Desimone (2009)'s conceptual perspective. Desimone’s focus is on measuring the impact of PD and she emphasizes the core features of effective PD, change in classroom practice, and student outcomes. The proposed PD4PCT focuses only on a part of her framework, namely content focus, active learning, collective participation, coherence, duration and context. In addition, elements were adapted from the 3C, ADAPTTER and CTTD frameworks, mentioned before.

This study assumes that understanding CT as a problem-solving skill will enhance teachers’ knowledge of the CT teaching and learning repertoire irrespective of their teaching specializations. The next section discusses the proposed framework’s components.

Pedagogical Content knowledge

Teachers should demonstrate an understanding of the CT definition, concepts and practices, as well as have the ability to employ and modify instructional strategies that facilitate student learning and application of CT across multiple subjects. These should all be based on feedback from PD programs and classroom observations. Additionally, teachers should be able to utilize a variety of materials, from paper to digital devices (where available), to aid students in their understanding of CT and the topics incorporated in it.

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

Incorporate Active Learning

Constructionism underpins this study. Active learner participation is crucial in constructionist instructional design, and task-based learning strategies are an excellent way to keep learners engaged throughout the course (Loi, 2004). Active learning, as proposed in this framework, should involve teachers in the process of designing and experimenting with instructional methods, allowing them to make use of authentic artefacts, immersive experiences, and other techniques to provide profoundly rooted and contextualized professional learning (Desimone, 2009). Professional development activities should actively engage teachers in meaningful discussion with other teachers or training specialists about the goal of a lesson, tasks, teaching strategies and practice. Teachers should apply a variety of CT practices, such as tinkering with existing artefacts to make changes or creating new ones by collaborating with each other.

The following sub-sections discuss the suggested teaching strategies for active learning that can be used in teaching teachers about CT skills.

A List of Active Learning Strategies

Unplugged computing is one strategy that is suggested to be used for the development of teachers' CT skills without the use of computing devices. This will enable teachers to engage in CT activities regardless of their school's computing infrastructure. It is also an easy method to use for teachers who do not have a computing background. Without the distraction of computer devices, the unplugged method broadens the reach of CT in different contexts (Huang & Looi, 2020).

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

Robotics is the second most often researched method for teaching CT skills to primary school teachers (Ausiku & Matthee, 2021). Training teachers to integrate educational robotics into their classrooms should increase their confidence and CT skills as it encourages them to construct and program robotic kits such as LEGO. Hence, learning to teach CT concepts through robotics can be seen as a constructionist endeavor and teachers should know how to design such learning environments to be able to support learners.

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

Game-based activities enable teachers to extend common ideas from their subjects to a variety of dynamic activities using CT skills to generate models and representations using games. In constructionist teaching approaches, technology, and games (which can range from digital cameras to complex simulations) are used to assist students to complete projects. This helps them to recognise and solve challenges, to grasp new phenomena, to create mental models of these phenomena, as well as learning how to set objectives and to take control of their own learning.

Coherence

In designing for teachers’ development, coherence implies striving to promote a culture of learning during PD events and encouraging teachers to understand their learning as being connected to a linked set of ideas about schools, students, teaching, and learning. Coherence helps teachers to develop a consistent set of thoughts regarding their teaching practice over time and influences the transformation of instruction. Coherence also includes delivering PD in a way which is consistent with the teachers' own skills and beliefs (Desimone, 2009).

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

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

The teachers’ skills and beliefs relating to CT should be considered and determined before and after the PD workshops and this can be done through pre- and post-intervention questionnaires (Bean et al., 2015; Korkmaz et al., 2017; Rich et al., 2020; Weese & Feldhausen, 2017). These skills and beliefs will have an influence on the coherence of the PD program. For example, if the teachers’ skill and knowledge is not considered and if the training is not aligned to their current subjects, then the PD workshop may not succeed in improving their content knowledge of CT. Likewise if the teachers’ beliefs are unknown, it will be difficult to change their attitudes towards CT, and hence will not improve their teaching practice to incorporate CT, which is the objective of the workshop (Desimone, 2009).

Duration

According to research, intellectual and pedagogical reform require PD programmes to be sufficiently lengthy, both in terms of the time covered by the practice (e.g., one day or one semester) and the number of hours expended on the activity (Desimone, 2009). In anticipation of limited time available, the workshops can be made up of blocks taking place as after-school sessions, each about 2 or 3 hours long, for 2 or 3 days per week. These workshops should continue for at least 4 weeks. Another option is to conduct the workshops on Saturdays e.g., two Saturdays per month extending over a semester. Another alternative is to train the teachers during school holidays through intensive sessions e.g., 5 days covering at least 4 hours per day. At least 20 contact hours is ideal and training needs to be continuous, not just a once off activity. Hence, it can be spread over an entire semester or school year.

Collective Participation

Desimone (2009) refers to collective participation and mentions that this can be best achieved by including teachers from the same school, grade, or department in sessions. By collaborating, teachers can build communities that have a positive impact on the culture and instruction of their entire grade level, department, classroom, and area (Desimone, 2009).

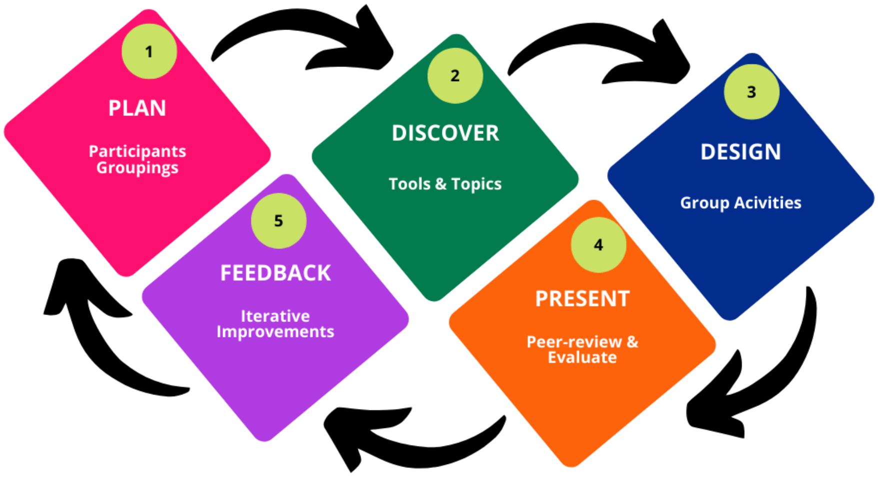
We enhance Desimone’s framework by suggesting an explicit method to implement collective participation, namely Participatory Design techniques. During the training workshops, teachers are provided with opportunities to collaborate in grade-level or subject-specific groups and to integrate CT skills while building new lesson plans and activities or enhancing existing ones. This allows them to have control over their creative process and to develop authentic artefacts that they can use in their classrooms. As Spinuzzi (2005) explained, participatory design enables teachers to redesign their working tools. The participatory design process is in line with constructionism, which is predicated on the notion that the most beneficial educational experiences involve the active construction of a variety of different things. These are particularly those that are significant on a personal or social level, are produced through interactions with other people as an audience, collaborators, and coaches, and encourage reflection on one's own way of thinking (Brennan, 2015; Papert, 1980).

The design process for the workshop follows the participatory design methodology and stages described by Agbo et al. (2021) and Spinuzzi (2005). The design methodology is the iterative construction of an artefact, and the design stages are plan and engage, discover, design, evaluate and feedback.

To better present these stages, Figure 2 below shows the process flow and connects it to participatory design.

Figure 2

Participatory Design Process Flow Diagram



Note. Adapted from Agbo et al. (2021) and Spinuzzi (2005).

Throughout the collaboration period, facilitators, and researchers 'step down' into direct working connections with teachers. This enables groups to ask facilitators questions throughout the preparation of lesson plans and activities, thereby acquiring just-in-time CT knowledge. Additionally, unlike traditional expert-to-novice PD approaches, participatory design presents teachers as knowledgeable collaborators with the agency to share their experiences gained from interactions with children and in a classroom environment. Hence, they make valuable contributions to the learning of others. At the conclusion of the workshop, teachers share, discuss, and reflect on their lesson plans and activities (Agbo et al., 2021).

Different teams can develop their own techniques in accordance with the constructionist principle of methodological and epistemological diversity, despite some characterizations of co-design as involving strong facilitation and well-defined duties (Kelter et al., 2021). According to the constructionist perspective, the sharing of a constructed artefact within a community is as critical as its construction, as it impacts a learner's comprehension and generation of meaning (Kurt, 2021).

In addition to lesson plans and activities, different teaching strategies should be shared and discussed amongst teachers. Teachers should choose the strategies that fit best with their school and classroom context.

Context

The effectiveness of the PD program depends on the context of the school and includes the school leadership and technological infrastructure.

Technological Infrastructure

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

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

Table 2 proposes a technology infrastructure and activities that can be done using programming, robotics or game-based learning strategies (Esteve-Mon et al., 2019; Gleasman & Kim, 2020; Jaipal-Jamani & Angeli, 2017; Leonard et al., 2016). It also shows how teachers can use the unplugged teaching strategy at schools where there are no computer labs or power connections (Brackmann et al., 2017; Rich et al., 2020).

Table 2

Technological Infrastructure Needs for Integrating CT into the Classroom

IT infrastructure	Computational thinking activities
● power connection ● functional and easily available computers or tablets with the necessary software,	● Automate algorithms and create programmes using Scratch and applying the following fundamental programming concepts:

IT infrastructure	Computational thinking activities
e.g., Scratch ● Internet connectivity ● a school-based IT team ● educational robotic kits, e.g., mBots, LEGO, etc.	○ Sequencing ○ Repetition ○ Variables ○ Conditionals ● Create interactive stories, animations and games. ● Programme the robotic kits to perform tasks.
No computers nor Internet connectivity	● Create algorithms (step-by-step instructions on paper). ● Playdough programming - one person (the programmer) instructs the other (the human computer) to create a playdough model based on verbal instructions alone. ● Computational word games - converse with your partner without breaking the flow. ● Paint by pixels - create pixelated graphics using a spreadsheet or piece of squared paper.

Note. Adapted from Brackmann et al. (2017), Jaipal-Jamani & Angeli (2017) and Rich et al. (2020).
IT = information technology; LEGO = leg godt.

School Leadership

The school culture and leadership play a vital role in ensuring that CT integration happens in the classroom. School leadership is viewed as an element in the stakeholder's relationships through the dimension of context (Brandt et al., 2012). The leaders should be aware of and involved in the integration plan, as a lack of cooperation from school management could impede CT integration (as shown in the study by Rich et al. (2017)). School administrators assist in articulating and communicating the school's vision and priorities, manage teaching and learning expectations, and create and authorize the instructional schedules that allow for CT integration efforts, thereby actively supporting their staff's PD (Sherwood et al., 2020; Boulden, 2021).

The Proposed PD4PCT Framework

Figure 3 and Table 3 provide a graphical presentation of the framework and the components and instruments developed for implementing the framework.

Figure 3

Initial Framework Graphical Presentation

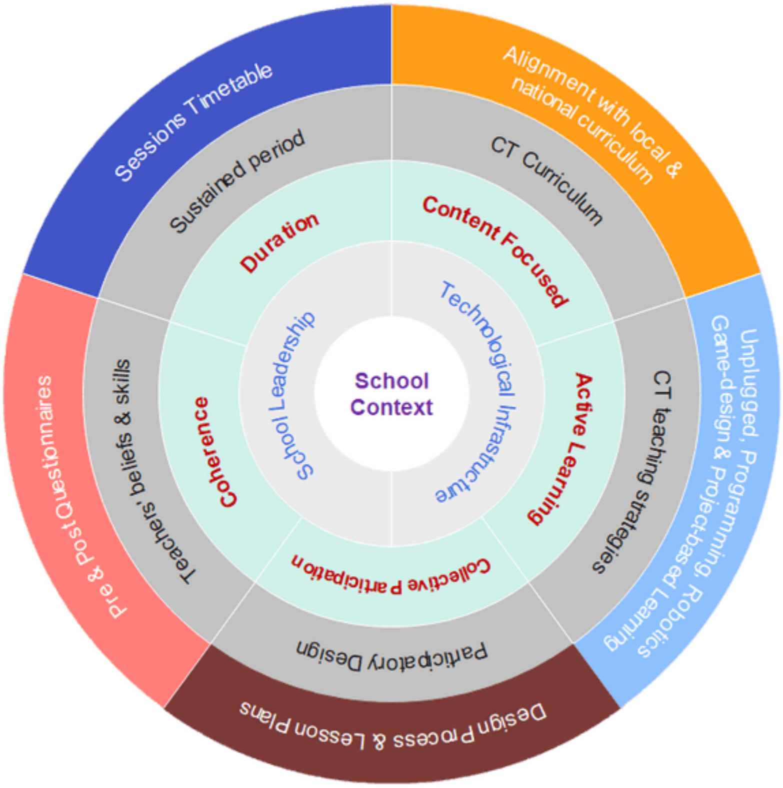


Table 3

Components and the Tools & Techniques Developed for the Implementation of the Framework.

Framework Component	Tools & techniques developed for the implementation of the framework.
Content Focused	CT Curriculum (Appendix)
Active Learning	List of active learning strategies
Coherence	Pre- and post- intervention questionnaires to measure teachers' skills, beliefs, and attitudes
Collective Participation	Participatory design stages (plan, discover, design, present, feedback)
Duration	Workshop sessions timetable spanning 4 weeks
Context	Mapping between technological infrastructure and CT strategies and strategies schools can follow to integrate CT

Note. CT = computational thinking.

PHASE 1: IMPLEMENTATION OF THE FRAMEWORK

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

School Context

The school which was the case is a primary school located in an informal settlement in Windhoek in Namibia. The school leadership had a positive outlook towards CT and supported the idea of being part of the study. The school principal and head of departments encouraged teachers to avail themselves to be participants, and this made it relatively easy for teachers to be active in the study. The school does not have a computer lab for learners currently but has few desktops and laptops for use by teachers.

Duration of the Participatory Training Workshop

A detailed workshop timetable was developed to guide the sessions. The workshop was 4 weeks long (from 11 October 2021 to 3 November 2021). The training sessions were 2 hours long and conducted after school for 3 days per week. Hence, the sessions allowed for a total of 24 hours of contact time. Teachers had the option to choose suitable dates based on their availability and this established self-motivation and control (Desimone, 2009).

Collective Participation Through Participatory Design

During the participatory design workshop, so as to encourage collective participation, teachers were grouped according to subject taught while working collaboratively as planned in Stage 1 *(Planning)* of the design process (Table 4). This resulted in five groups: Social studies, natural science, mathematics, English and Afrikaans. The social studies and English groups consisted of three teachers each, the natural science group had four teachers, while the mathematics and Afrikaans groups had two teachers each. Each group was provided with pens, paper, a CT Poster, a computer with Scratch software installed, the sessions timetable, etc.

Table 4

Summary of Participatory Design Process Implementation

Stage	Activities
Plan	● Group teachers in small teams per subject. ● Provide timetable and tools needed. ● CT poster, pens, paper, computers with Scratch program and lesson plan templates.
Discover	● Teachers select topics from the local/national syllabus that they feel are suitable for training (Trade, Sequences, Ecosystem, Grammar, Action Words). ● Provide advice on a topic selection with CT linkage.
Design	● Practical lesson development considering classroom context. ● Brainstorming on lesson plan components, paper and mock-up drawings, and concept presentations ● Work with teachers to define key ideas of subject matter (CT).

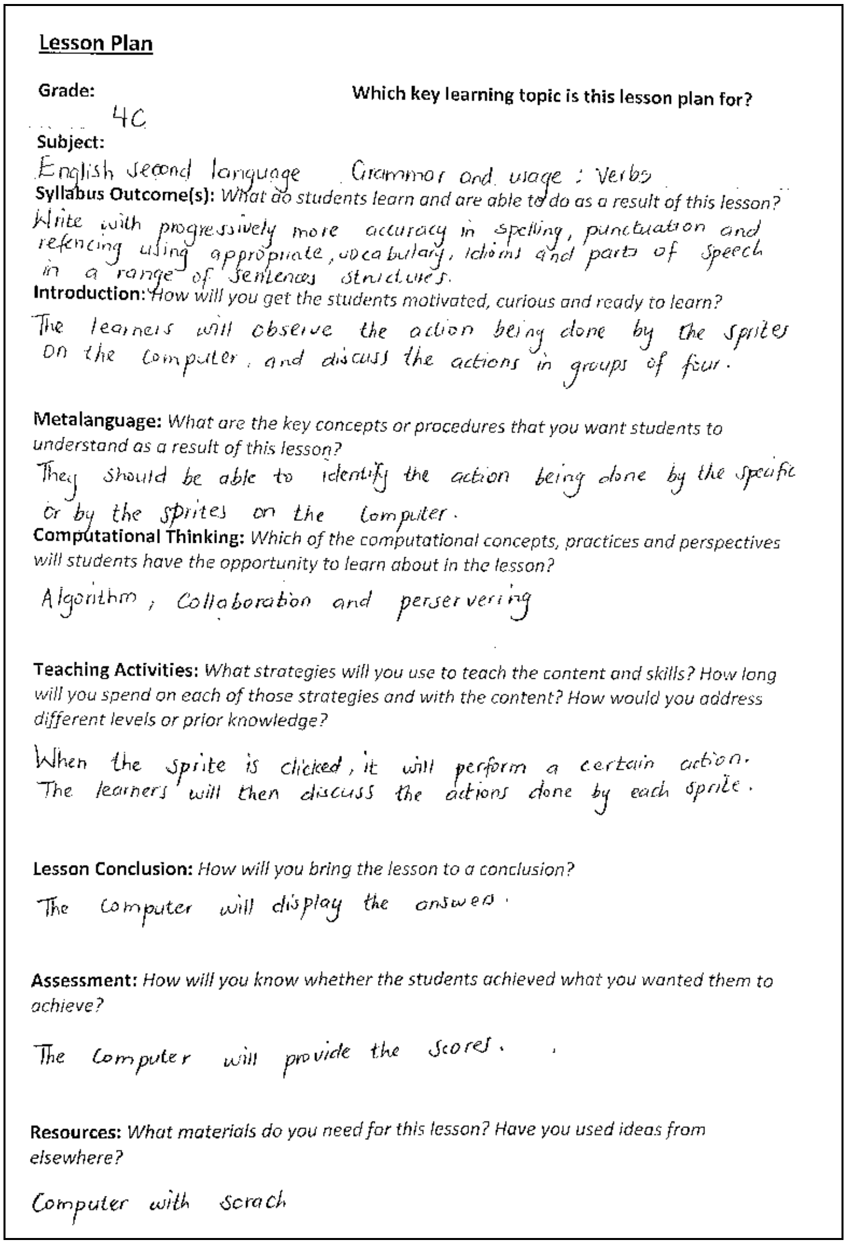
Stage	Activities
Present	● Modelling and practicing activities and teaching methods. ● Groups present their artefacts and peer-review each other. ● Work through CT activities with teachers in the role of learners.
Feedback	● Give other teams feedback on their artefacts - this leads to iterative improvements of the lesson plans and activities.

Note. CT = computational thinking.

Figure 4 shows a lesson plan designed by English teachers during the participatory design workshop.

Figure 4

A CT-Infused Lesson Plan for English Prepared by Teachers



Lesson Plan

Grade: 4C

Which key learning topic is this lesson plan for?

Subject: English second language Grammar and usage : Verbs

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

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

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

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

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

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

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

Algorithm, Collaboration and perservering

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

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

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

The computer will display the answer.

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

The computer will provide the scores.

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

Computer with scrach

Enhancing Teachers' Content Knowledge

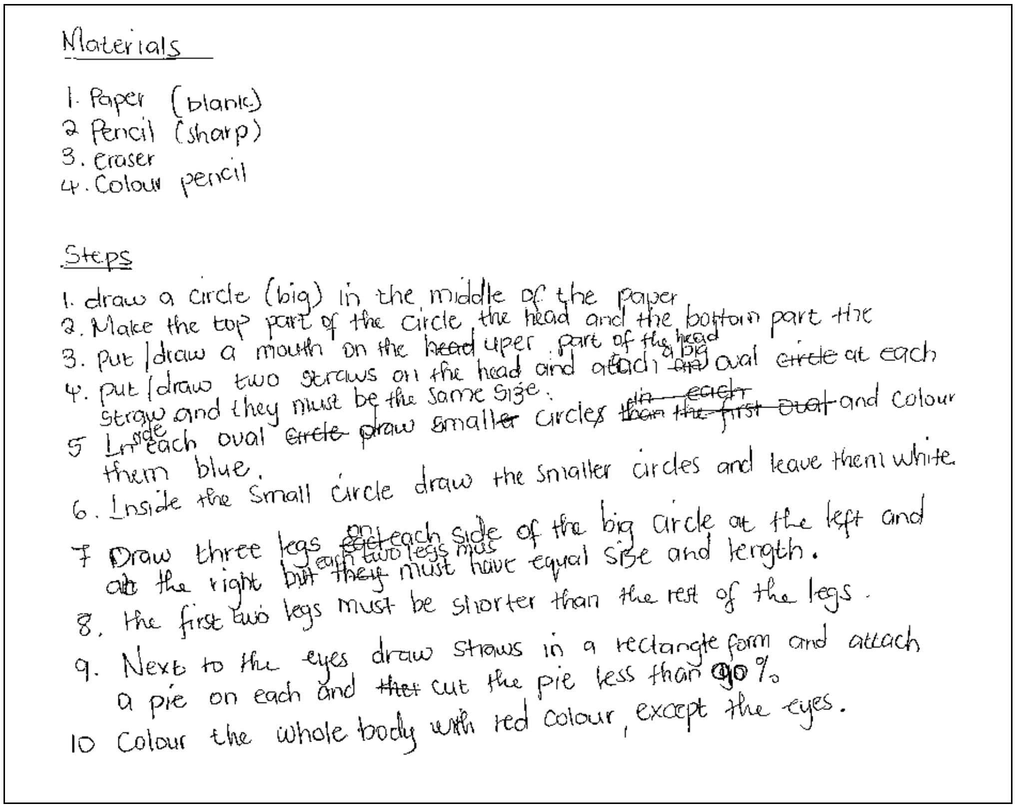
For the teachers to understand CT and its integration, the CT curriculum used in training focused on CT definition, CT concepts and approaches. The first day of the training was devoted to CT knowledge; teachers were introduced to the definition of CT, the CT concepts of algorithms, abstraction, decomposition, pattern recognition and evaluation.

Active Learning through Constructionism

During the training, teachers were introduced to a variety of teaching strategies for CT such as unplugged, programming, robotics, game-design and project-based. However, only unplugged and programming strategies were used during the workshop. The first programming session introduced the teachers to the concepts of sequences, loops, and conditionals ahead of the hands-on activities in the following session. As the study is rooted in constructionism, teachers were active participants during the sessions doing hands-on activities using both unplugged and programming strategies. For the unplugged strategy, teachers used pen and paper to design and execute tasks without computing devices. For example, in an activity called "Draw a crab," a group of teachers was shown a simple picture of a crab and had to write instructions for how to draw it. These instructions were then given to a different group to follow. This activity tested the skills of abstraction, algorithms, rules, cause, and effect. Figure 5 presents an output from the activity carried out by teachers.

Figure 5

Unplugged Algorithm Activity to Draw a Crab Created by Teachers



Materials

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

Steps

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

Meanwhile, for the programming strategy, the teachers used computers with Scratch software to create programs. This session taught teachers how they could teach CT concepts through the programming (coding) strategy using Scratch, a drag and drop visual programming language. During the hands-on session with Scratch, teachers worked in the groups formed during the planning stage (Table 4) to complete an activity called "Dancing cat." In this activity, they needed to apply the loop concept to make the cat talk and move. To complete the task successfully algorithmic thinking, collaboration and debugging was needed. Figure 6 is a screenshot of the code from the teachers' activity.

Figure 6

An Output from the "Dancing Cat" Activity Created in Scratch by the Teachers

```
when space key pressed
set number of movements to 0
ask How many times would you like me to do my dance? and wait
if answer < 1 then
    say join I can't dance for join answer times! for 2 seconds
else
    if answer > 100 then
        say I can't dance that many times! I'll get worn out! for 2 seconds
    else
        repeat answer
            repeat 2
                move 40 steps
                change number of movements by 1
                play drum (1) Snare Drum for 0.5 beats
                move -40 steps
                change number of movements by 1
                play drum (1) Snare Drum for 0.5 beats
            play drum (5) Open Hi-Hat for 1 beats

times to do dance
```

FINDINGS – PHASE 1

This section presents findings from the case study that were analyzed to test and refine the PD4PCT framework. The section is divided into sub-sections according to the relevant components of the framework, namely CT content knowledge, coherence, active learning, and collective participation.

CT Content Knowledge

This sub-section includes the analysis of items from the pre- and post-intervention questionnaires used to assess the teachers' perceived CT knowledge before and after the implementation of the framework (Bean et al., 2015; Korkmaz et al., 2017; Rich et al., 2020; Weese & Feldhausen, 2017).

When comparing the responses received before and after the training, it could be seen that there were substantial increases in the median values to all CT Knowledge statements, with large effect sizes. Hence, the teachers who attended the training event had gained a deeper knowledge of CT.

This comparison's statistical findings are summarised in Table 5. As described in the methodology section, the data distribution was not normal; hence, the Wilcoxon signed-rank test was used to compute the *p* value. The effect sizes (*r*) were determined using *Rosenthal's r* formula.

Table 5

Summary of the CT Knowledge Items Responses (N=14)

Item No	Item Description	Pre-Median	Post-Median	*P* value	*r* value
1	I can define what computation thinking is.	2.00	4.00	0.002	-0.59
2	I can describe fundamental computational thinking concepts (e.g., algorithms, abstraction, decomposition, pattern recognition & evaluation).	2.00	4.50	0.002	-0.58
3	I can describe fundamental coding/programming concepts (e.g., loops, variables, conditional logic).	0.00	4.00	0.001	-0.61
4	I can look at a process and figure out how to make it more efficient.	2.00	4.00	0.009	-0.49
5	I can suggest different solutions to solve problems.	2.50	4.00	0.003	-0.56
6	I can generalise solutions that can be applied to many problems.	2.50	4.00	0.002	-0.59
7	I am good at finding patterns in data.	4.00	5.00	<.001	-0.67
8	I am good at solving puzzles.	4.00	5.00	0.012	-0.48
9	I can read a formula (e.g., algorithm, equation, input/output process) and explain what it should do.	2.50	4.00	0.003	-0.56
10	When I'm presented with a problem, I can easily break it down into smaller steps.	2.00	4.00	0.003	-0.57
11	When solving a problem, I work with others to solve different parts of the problem simultaneously.	4.00	4.50	0.002	-0.58
12	When solving a problem, I look how information can be collected, stored, and analysed to help solve the problem.	2.00	4.00	0.002	-0.58
13	When solving a problem, I create a solution where steps can be	2.00	4.50	0.003	-0.56

Item No	Item Description	Pre-Median	Post-Median	*P* value	*r* value
	repeated.				
14	When solving a problem, I create a solution where some steps are done only in certain situations.	2.00	4.00	0.001	-0.61
15	When solving a problem, I try to simplify the problem by ignoring details not needed.	2.00	5.00	0.003	-0.56

Note. CT = computational thinking.

Coherence

This section includes the analysis of items from the pre- and post-intervention questionnaires conducted to measure the teachers' beliefs and attitudes towards CT (Bean et al., 2015; Korkmaz et al., 2017; Rich et al., 2020; Weese & Feldhausen, 2017).

The results show that the teachers' responses to the statements about their beliefs and attitudes towards CT were generally low before the training. Overall, teachers who participated in the training program had their views and attitudes towards CT altered for the better.

This comparison's statistical findings are summarised in Table 6.

Table 6

Summary of Teachers' Beliefs and Attitude Towards CT Responses

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

Note. CT = computational thinking.

When comparing the median responses before and after the training, there were significant increases in participants' replies to all CT teacher self-efficacy statements, with effect sizes greater than -0.47. Therefore, the total self-efficacy of the teachers who attended the training session increased with regard to CT. This comparison's statistical findings are summarised in Table 7.

Table 7

Summary of Teachers' Self-Efficacy on CT Responses

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

Note. CT = computational thinking.

Active Learning and Collective Participation

During the participatory training workshop, teachers were regarded as design partners and invited to select their subject topics and construct activities and lesson plans incorporating CT concepts and practise. The inductive qualitative analysis revealed two themes relating to the framework's active learning and collective participation components, described in the following sub-sections.

Integrating CT as Design Partners

The results demonstrated that teachers used algorithms most frequently, followed by decomposition and pattern recognition. Most teachers who designed activities utilised the collaboration and perseverance approaches because they believed that activities were more enjoyable and productive when completed in pairs or groups. As design partners, teachers indicated that they learnt how to design their lesson plans and activities incorporating CT concepts. The data also showed that teachers appreciated the feedback received from peers as part of a collaboration with others who were teaching different subjects. When analyzing the interview data, it shows that social studies and mathematics teachers incorporated at least two CT concepts and practices into their lessons and activities compared to other subjects 'teachers as shown in Table 8.

Table 8

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

Group	Subject	CT concepts	CT practices	Active learning strategy
Group 1	Social Studies	Algorithms Decomposition	Collaboration Persevering	Unplugged, Programming
Group 2	English	Algorithms	Collaboration Persevering	Programming
Group 3	Natural Science	Algorithms, Pattern recognition	Collaboration	Unplugged, Programming
Group 4	Mathematics	Algorithms, Decomposition, Pattern recognition	Collaboration Persevering Debugging	Unplugged
Group 5	Afrikaans	Debugging	Creating Tinkering Collaboration	Programming

Note. CT = computational thinking.

Future Plans to Integrate CT

After training, all teachers agreed that given their school context, they would use the unplugged strategy in their classroom to incorporate CT skills by means of posters, workbooks, and flashcards:

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

Teachers discussed integrating CT approaches into their activities and lessons in the qualitative data. A common approach used for integrating CT was collaboration, so learners work in groups and experience perseverance as part of the process. One teacher indicated that they would also use Scratch for creating quizzes, and learners would be able to use the few available computers for group work. Overall, the results from the interviews indicated that most teachers could design lesson plans and activities that incorporated algorithms, decomposition, and pattern recognition concepts. They also preferred to apply collaboration to classroom activities when integrating CT skills. Looking at the data, teachers were uncomfortable integrating concepts such as abstraction and debugging into their lessons, suggesting that they did not grasp the concepts and could not align them to their topics. Regarding their future plans for integrating CT, they all agreed that the best-suited teaching strategy for their school was unplugged due to the lack of computers at the school.

Context

When doing a PD program for teachers, one should consider the context of their school. This study's findings showed that the teachers were aware of their context when choosing the material and teaching strategies for their CT-infused lesson plans. All the teachers agreed that they would prefer to use unplugged strategies because of the lack of computers at their school. This agrees with the findings of Sherwood et al. (2020) that CT integration can take different forms depending on the needs of the teachers or school. In addition, schools where there is no electricity can also do unplugged activities (Brackmann et al., 2017; Rich et al., 2020). Another contextual aspect that appeared in the findings was that of school leadership. The teachers who mentioned this indicated that their school leadership had a positive attitude towards technology and CT. They believed that when money became available, they would not hesitate to buy more computers because they were already busy building a computer lab. This finding is supported by Leonard et al. (2017), Rich et al. (2017) and Sherwood et al. (2020), namely that school leaders need to be aware of and involved in CT integration.

Other Findings - Subject Matter Knowledge

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

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

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

These teachers' remarks highlight the importance of subject matter expertise as a precondition for effective integration of CT with the specific subject's topics (Kirwan et al., 2022; Jocius et al., 2020).

PHASE 2: VALIDATION OF THE FRAMEWORK

The validation approach followed aimed to assess the validity and applicability of the PD4PCT framework components and to verify the dependability of the recommendations.

Expert Feedback

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

Applicability of the Framework to the Namibian Context

The reviewers' responses indicated that they all believed that the framework was suitable for the existing curriculum in Namibia. They also believed that the study was pertinent and topical and that if the findings were shared with the country's education ministry, they could affect change in the education system.

The Role of Technology in Society

The experts also emphasized the importance of technology today and urged the incorporation of CT into the teaching of existing topics at primary school level. Learner-centered techniques were also recommended since they help increase the efficiency of learning.

Application of the Framework to PD

One possible challenge to implementing this framework was thought to be gaining the support of teachers and school authorities. To promote buy-in, one expert expressed the view that stakeholders needed to be persuaded by helping them to comprehend the democratic importance of CT and to see the need to incorporate CT into teacher training programmes. Another point made by the expert regarding the teachers' PD was the shortage of trained teacher educators with the ability to integrate CT. The experts advocated for collaboration between schools and industry experts to address this difficulty. The responses also indicated that limited access to digital tools and curricular resources was a barrier for teachers and PD providers. One expert emphasized that teachers should be offered ICT literacy training before an intervention that requires computer use.

Relevance of the Framework

One expert, who is the head of the PD division in the region, referred to poor PD programmes conducted in the past and how this framework could be used to design more effective training programs for teachers. The experts did not find any gaps in the framework. Some experts remarked that the framework would be useful and would be applicable to training programs.

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

Revised PD4PCT Framework

The framework was revised after taking into consideration the implementation experience mentioned earlier and the experts' evaluation results. This led to the addition of a new component "Integration levels". Following the informal discussions with teachers that transpired during training it became clear that a lack of subject expertise could hamper the integration of CT into some topics. This research identifies a variety of likely reasons why teachers might steer clear of certain topics. Teachers typically lack the confidence to give instruction in something they are unfamiliar with or when they have little topic expertise (Koehler et al., 2013; Shulman, 1986). The teachers in this research were similar to other primary school teachers worldwide in that they were generalists and lacked specialized training in particular areas. The implementation experience and expert comments from the validation questionnaire also alluded to expert involvement.

The contribution of subject experts is noted in Jocius et al. (2020)'s 3C framework where they had Computer Science and core subject experts as facilitators during the intervention. This element was initially covered in our framework by assuming that the collective expertise per group would suffice. However, following the informal discussions with the teachers and the feedback from reviewers, it is clear that there is a need to involve subject experts in the interventions. This resulted in its inclusion in the revised framework. The tasks would include understanding teachers' subject knowledge and identifying topics that are suitable for CT integration ahead of time. Other recommendations are:

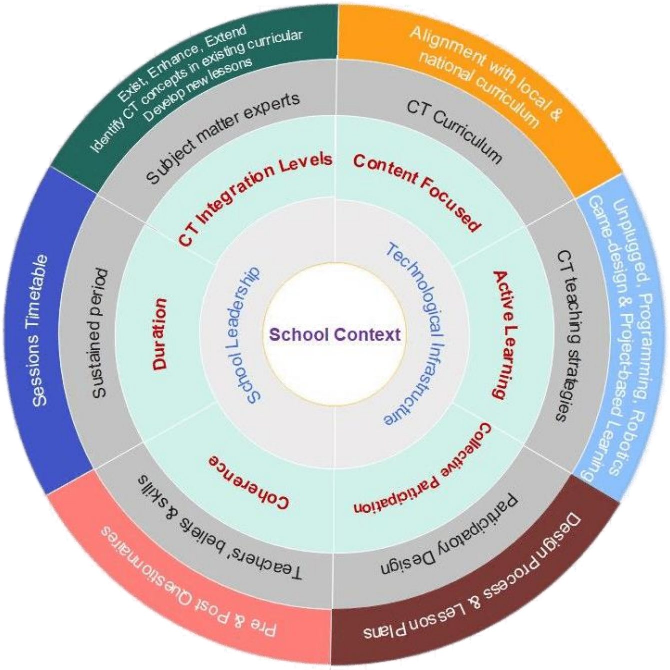
1. The facilitator should explain fundamental CT concepts, linking them to specific subjects with examples.
2. Increase the duration of the intervention to over 24 hours.
3. Provide opportunities for further training after the intervention.
4. Offer an initial computer literacy course to teachers before or at the start of the intervention. (It was suggested during the validation phase by experts that computer literacy training be offered to teachers before a CT intervention. This is because not all teachers are computer literate and eventually they would need to use computing devices to enhance their CT lessons.)

Revised Graphic Representation

The revised framework is provided in Figure 7. The additional component of Integration Levels deals with identifying CT concepts in existing curricular and then enhancing and extending them by developing lessons with the help of subject matter experts. The development of lessons needs to be at three different levels: Exist – identify existing CT practices, concepts; Enhance - add lessons or activities to deepen the connections between subjects and CT concepts and Extend - integrate activities connected to CT to enhance subject knowledge - this is likely to involve programming (Waterman et al., 2019).

Figure 7

Revised PDF4CT Framework



Note. CT = computational thinking.

DISCUSSION

This section explores the connection between the case study's findings and existing literature, and it’s structured according to the components of the framework.

CT content knowledge

Participation in the training significantly improved teachers' grasp of computational concepts, in line with Hickmott's (2020) perspective on the positive impact of PD programs for CT. The training, incorporating real-life contexts, benefited teachers with limited CT knowledge, as seen in similar studies (Çakır et al., 2021; Corradini et al., 2017; Mason & Rich, 2019; Uzumcu & Bay, 2020), enhancing their ability to integrate coding and CT into curricula. Mathematics teachers found it easier to integrate CT concepts using the unplugged technique, supported by research like Ausiku and Matthee (2020) and Huang and Looi (2020). This method provided a strong theoretical basis, enabling seamless integration of CT concepts, corroborated by studies emphasizing the link between CT and mathematics (Rich, Yadav et al., 2019; Nordby et al., 2022). Direct teacher preparation through unplugged methods and promoting CT as a general problem-solving approach improved teachers' instructional abilities, as evident in the study's results.

Coherence

After training, teachers' attitudes towards CT improved significantly, as indicated by a questionnaire adapted from Bean et al. (2015). Initially sceptical, teachers became confident in teaching CT, aligning with effective CT teaching shown in prior studies (Bower et al., 2017; Curzon, McOwan et al., 2014; Falkner et al., 2018). The training enhanced their belief that CT concepts can be taught at the primary level in any subject, reflecting positive changes. However, teachers felt that the limited duration of the intervention hindered their confidence, suggesting a need for extended training with more hands-on activities to enhance their CT teaching skills. Additionally, lack of time remains a significant barrier to integrating CT into classrooms, aligning with challenges noted in prior studies (Bower et al., 2017; Kadirhan et al., 2018; McGinnis et al., 2019; Rich et al., 2017).

Collective Participation

According to the findings, some teachers prefer hands-on activities and longer training for comfortable CT teaching, aligning with Desimone's (2009) framework emphasizing active, sustained PD. Collaboration and co-designing, backed by Kelter et al. (2021) and Kurt (2021), enhance learning and meaningful comprehension in small groups, allowing meaningful contributions (Agbo et al., 2021). Limited computer access led teachers to opt for unplugged CT activities, especially in developing countries where context crucially influences teaching tactics, as seen in previous studies (Ausiku & Matthee, 2020; Espinal et al., 2021; Kong & Wong, 2017; Muñoz del Castillo et al., 2019).

Active Learning

In this study, teachers actively participated in designing instructional materials and collaborated with the researcher on lesson objectives and strategies, aligning with active learning principles (Darling-Hammond et al., 2017; Desimone, 2009). They engaged in constructive discussions, creating tailored teaching materials with facilitation from the researcher, following a constructivist approach noted by Kurt (2021) and Loi (2004). Teachers integrated CT concepts into their teaching materials using unplugged and programming methods. They played an equal role as decision-makers, reflecting the collaborative approach emphasized in the study by Iversen et al. (2017).

Duration

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

CONCLUSIONS

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

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

By comparing the PD4PCT framework to the three existing ones, the ADAPPTER framework was purely for secondary school teachers and focused more on the teaching process, while the 3C framework combined middle school and high school teachers. None of the three existing frameworks explicitly mentions the elements of context, collective participation, and coherence (aligning the intervention with teachers’ beliefs and attitudes towards CT). The three frameworks were created and tested in developed countries. The PD4PCT framework contributes towards the scientific body of knowledge on the PD of primary school teachers in a developing economy context to integrate CT into their teaching.

The results showed that while developing a PD program or intervention for teachers, the school context of the teachers should be considered first. This element will establish the tone for the intervention's tactics and strategies. It was also revealed that teachers’ beliefs and attitudes should be known before implementing training. It was also discovered that lack of subject matter knowledge can impede the integration of CT into certain topics, hence the importance of using subject matter experts during an intervention.

It is crucial to highlight the constraints that this study had. A key limitation is that the study was limited to 14 teachers from one school in Windhoek, focusing only on the senior primary phase.

The study lays the groundwork for future research and adds to the knowledge related to teacher skill development in CT. Understanding the CT curriculum is insufficient; there has to be a mapping to the subject areas where CT can be used, so more research is needed on how these mappings can be done effectively. One recommendation for further research based on this study is to retest the results in other developing economies in Africa or Asia.

REFERENCES

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Laine, T. H. (2021). Co-design of mini games for learning computational thinking in an online environment. In *Education and Information Technologies, 26*(5), 5815-5849. https://doi.org/10.1007/s10639-021-10515-1

Alfayez, A. A., & Lambert, J. (2019). Exploring Saudi Computer Science teachers’ conceptual mastery level of computational thinking skills. *Computers in the Schools*, *36*(3), 143–166. https://doi.org/10.1080/07380569.2019.1639593

Alqahtani, M. M., Hall, J. A., Leventhal, M., & Argila, A. N. (2021). Programming in mathematics classrooms: Changes in pre-service teachers' intentions to integrate robots in teaching. *Digital Experiences in Mathematics Education*, *8*(1), 70–98. https://doi.org/10.1007/S40751-021-00096-6.

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

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47–57.

Ausiku, M., & Matthee, M. (2021). Preparing primary school teachers for teaching computational thinking: A systematic review. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12511 LNCS* (pp. 202–213). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-66906-5_19

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, *2*(1), 48-52. https://doi.org/10.1145/1929887.1929905

Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015). Starting from scratch: Developing a pre-service teacher training program in computational thinking. In *Proceedings - Frontiers in Education Conference, FIE, 2014,* (pp. 1–8). https://doi.org/10.1109/FIE.2015.7344237

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking: Approaches and orientations in K-12 education. In *EdMedia+ Innovate Learning* (pp. 13-18). Association for the Advancement of Computing in Education (AACE).

Bogner, A., Littig, B., & Menz, W. (Eds.). (2009). *Interviewing Experts*. Springer. https://doi.org/10.1057/9780230244276

Boulden, D. C., Rachmatullah, A., Oliver, K. M., & Wiebe, E. (2021). Measuring in-service teacher self-efficacy for teaching computational thinking: development and validation of the T-STEM CT. *Education and Information Technologies, 26*(4), 4663–4689. https://doi.org/10.1007/S10639-021-10487-2

Bower, M., Wood, L.N., Lai, J.W., Highfield, K., Veal, J., Howe, C., Lister, R., & Mason, R., (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, *42*(3), 53–72. https://doi.org/10.14221/ajte.2017v42n3.4

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th Workshop On Primary and Secondary Computing Education* (pp. 65-72). https://doi.org/10.1145/3137065.3137069

Brandt, E., Binder, T., & Sanders, E. (2012). Tools and techniques: Ways to engage telling, making and enacting. In *Routledge International Handbook of Participatory Design* (pp. 165-201). Routledge. https://doi.org/10.4324/9780203108543-14

Braun, V., & Clarke, V. (2012). Thematic analysis. In APA Handbook of Research Methods in Psychology, Vol 2: Research Designs: Quantitative, Qualitative, Neuropsychological, and Biological. (pp. 57-71). American Psychological Association. https://doi.org/10.1037/13620-004

ComputerBrennan, K. (2015). Beyond technocentrism: Supporting constructionism in the classroom. *Constructivist Foundations*, *10*(3), 289-296. http://www.univie.ac.at/constructivism/journal/10/3/289.brennan

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, *18*(1), 32-42. https://doi.org/10.3102/0013189X018001032

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

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J (2015). *Computational Thinking - A Guide for Teachers*. Computing At School.

Department of Basic Education of South Africa. (2021). *Draft CAPS for Coding and Robotics*. https://www.education.gov.za/DraftCapsCodingRobotics.aspx

Desimone, L. M. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational Researcher, 38*(3), 181-199. https://doi.org/10.3102/0013189X08331140

Dolgopolovas, V., Dagienė, V., Jasutė, E., & Jevsikova, T. (2019). Design Science Research for computational thinking in constructionist education: A pragmatist perspective. *Problemos*, *95*, 144-159. https://doi.org/https://doi.org/10.15388/Problemos.95.12

Du, X., & Igwe, K. C. (2018). Computational thinking in teacher professional development programs. In V. Dagienė & E. Jasute (Eds.), *Constructionism 2018* (pp. 193-203).

Esteve-Mon, F., Adell-Segura, J., Nebot, M., Novella, G., & 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*(1), 139-152. https://doi.org/10.28945/4442

Esteve-Mon, F. M., Llopis, M. A., & Adell-Segura, J. (2020). Digital competence and computational thinking of student teachers. *International Journal of Emerging Technologies in Learning*, *15*(2), 29-41. https://doi.org/10.3991/ijet.v15i02.11588

Fessakis, G., & Prantsoudi, S. (2019). Computer Science teachers' perceptions, beliefs and attitudes on computational thinking in Greece. *Informatics in Education*, *18*(2), 227-258. https://doi.org/10.15388/infedu.2019.11

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R*. Sage publications.

Gadanidis, G., Clements, E., & Yiu, C. (2018). Group theory, computational thinking, and young mathematicians. *Mathematical Thinking and Learning*, *20*(1), 32-53. https://doi.org/10.1080/10986065.2018.1403542

Gleasman, C., & Kim, C. (2020). Pre-service teacher's use of block-based programming and computational thinking to teach elementary mathematics. *Digital Experiences in Mathematics Education, 6,* 52-90. https://doi.org/10.1007/s40751-019-00056-1

Grover, S., & Pea, R. (2013). Computational thinking in K-12. *Educational Researcher, 42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Huang, W., & Looi, C. K. (2020). A critical review of literature on "unplugged" pedagogies in K-12 Computer Science and computational thinking education. *Computer Science Education*, *31*(1), 1-29. https://doi.org/10.1080/08993408.2020.1789411

Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education & Technology*, *26*(2), 175-192. https://doi.org/10.1007/s10956-016-9663-z

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. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 971-977). https://doi.org/10.1145/3328778.3366797

Kadirhan, Z., Gül, A., & Battal, A. (2018). Self-efficacy to teach coding in K-12 education. In *Self-Efficacy in Instructional Technology Contexts* (pp. 205-226). Springer International Publishing. https://doi.org/10.1007/978-3-319-99858-9_12

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational what? Relating computational thinking to teaching. *TechTrends, 62*(6), 574-584. https://doi.org/10.1007/s11528-018-0290-9

Karakasis, C., & Xinogalos, S. (2020). BlocklyScript: Design and pilot evaluation of an RPG platform game for cultivating computational thinking skills to young students. *Informatics in Education*, *19*(4), 641-668. https://doi.org/10.15388/infedu.2020.28

Kelly, A. E. (2013). When is design research appropriate? In T. Plomp, & N. Nieveen Eds., *Educational Design Research*, (pp. 135-150). Netherlands Institute for Curriculum Development (SLO).

Kelter, J., Peel, A., Bain, C., Anton, G., Dabholkar, S., Horn, M. S., & Wilensky, U. (2021). Constructionist co-design: A dual approach to curriculum and professional development. *British Journal of Educational Technology*, *52*(3), 1043-1059. https://doi.org/10.1111/bjet.13084

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2019). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology*, *29*,174-188. https://doi.org/10.1007/s10956-019-09798-4

Kirwan, C., Costello, E., & Donlon, E. (2022). ADAPTTER: Developing a framework for teaching computational thinking in second-level schools by design research. *TechTrends*, *66*(3), 495-509. https://doi.org/10.1007/s11528-022-00735-8

Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)?. *Contemporary Issues in Technology and Teacher Education*, *9*(1), 60-70 https://doi.org/10.1177/002205741319300303

Koehler, M. J., Mishra, P., & Cain, M. W. (2013). What is technological pedagogical content knowledge (TPACK)? *Journal of Education*, *193*(3), 13-19. https://doi.org/10.1177/002205741319300303

Kong, S. C., & Lai, M. (2021). A proposed computational thinking teacher development framework for K-12 guided by the TPACK model. *Journal of Computers in Education*, *9*(3), 379-402. https://doi.org/10.1007/s40692-021-00207-7

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers and Education*, *151*(March), 103872. https://doi.org/10.1016/j.compedu.2020.103872

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558-569. https://doi.org/10.1016/j.chb.2017.01.005

Kurt, S. (2021). *Constructivist Learning Theory - Educational Technology*. Educational Technology. https://educationaltechnology.net/constructivist-learning-theory/

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate Peripheral Participation*. Cambridge university press. https://doi.org/10.1017/CBO9780511815355

Leonard, J., Barnes-Johnson, J., Mitchell, M., Unertl, A., Stubbe, C. R., & Ingraham, L. (2017). Developing teachers' computational thinking beliefs and engineering practices through game design and robotics. In *North American Chapter of the International Group for the Psychology of Mathematics Education*.

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, stem attitudes, and computational thinking skills. *Journal of Science Education and Technology*, *25*(6), 860-876. https://doi.org/10.1007/s10956-016-9628-2

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research*, 3, 1-18. Springer. https://doi.org/10.1007/s41979-020-00030-2

Loi, D. (2004). Teaching design theory and practice: A participatory journey. In *DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia* (pp. 659-664). https://www.designsociety.org/publication/19818/teaching+design+theory+and+practice%3a+a+participatory+journey

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

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education* 19(4), 790-824.

Massachusetts Department of Elementary and Secondary Education. (2016). *2016 Massachusetts Digital Literacy and Computer Science (DLCS) curriculum framework. June*. http://www.doe.mass.edu/frameworks/dlcs.pdf

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, *108*(6), 1017-1054. https://doi.org/10.1111/j.1467-9620.2006.00684.x

Mor, Y., Warburton, S., & Winters, N. (2012). Participatory pattern workshops: A methodology for open learning design inquiry. *Research in Learning Technology*, *20*. https://doi.org/10.3402/rlt.v20i0.19197

Mouza, C., Yang, H., Pan, Y. C., Yilmaz Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, *33*(3). https://doi.org/10.14742/ajet.3521

Ozturk, Z., Dooley, C. M. M., & Welch, M. (2018). Finding the hook: Computer Science education in elementary contexts. *Journal of Research on Technology in Education*, *50*(2), 149-163. https://doi.org/10.1080/15391523.2018.1431573

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books. https://dl.acm.org/citation.cfm?id=1095592

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism, 36*(2), 1-11.

Rich, K. M., Yadav, A., & Larimore, R. A. (2020). Teacher implementation profiles for integrating computational thinking into elementary Mathematics and Science instruction. *Education and Information Technologies*, *25*, 3161-3188. https://doi.org/10.1007/s10639-020-10115-5

Rich, P. J., Browning, S. F., Perkins, M. K., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2019). Coding in K-8: International trends in teaching elementary/primary computing. *TechTrends*, *63*(3), 311-329. https://doi.org/10.1007/s11528-018-0295-4

Rich, P. J., Jones, B., 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), 1-20. https://doi.org/10.21585/ijcses.v1i1.6

Rowley, J. (2002). Using case studies in research. *Management Research News*, *25*(1), 16-27. https://doi.org/10.1108/01409170210782990

Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, *10*(1), 73-88. https://doi.org/10.15388/infedu.2011.06

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In *Computational Thinking in the STEM Disciplines* (pp. 151-164). Springer International Publishing. https://doi.org/10.1007/978-3-319-93566-9_8

Schuler, D., & Namioka, A. (1993). Participatory Design: Principles and Practices. CRC Press.

Selby, C., & Woollard, J. (2014). Refining an understanding of computational thinking. http://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469-495. https://doi.org/10.1007/s10639-016-9482-0

Sherwood, H., Yan, W., Liu, R., Israel, M., Fancsali, C., & Pierce, M. (2020, February). School-wide integration of computational thinking into elementary schools: A cross-case study. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (p. 1325). https://doi.org/10.1145/3328778.3372623

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, *15*(2), 4-14. https://doi.org/10.3102/0013189X015002004

Spinuzzi, C. (2005). The methodology of participatory design. *Technical Communication*, *52*(2), 163-174.

Takahashi, S. (2011). Co-constructing efficacy: A "communities of practice" perspective on teachers' efficacy beliefs. *Teaching and Teacher Education*, *27*(4), 732-741. https://doi.org/10.1016/j.tate.2010.12.002

Tuhkala, A. (2019). Participatory design: An approach for involving teachers as design partners. In *JYU Dissertations*. http://urn.fi/URN:ISBN:978-951-39-7782-5

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2019). Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, *29*(1), 53-64. https://doi.org/10.1007/s10956-019-09801-y

Weese, J. L., & Feldhausen, R. (2017). STEM outreach: Assessing computational thinking and problem solving. In *ASEE Annual Conference and Exposition, Conference Proceedings*, *2017-June*. https://doi.org/10.18260/1-2--28845

Weintrop, D., Walton, M., Elby, A., Walkoe, J., & others. (2020). Mutually supportive mathematics and computational thinking in a fourth-grade classroom.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565-568. https://doi.org/10.1007/s11528-016-0087-7

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

APPENDIX: CT CURRICULUM

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

CT Skills	Grades 0-2	Grades 3-5	Grades 6-7
Decomposition	● Break a complex task into simpler subtasks (e.g., break a long path into a series of smaller paths that one can follow).	● Break a complex task into simpler subtasks. ● Develop a solution by assembling collections of smaller parts (e.g., organizing a school trip).	● Break a complex task into simpler subtasks. ● Develop a solution by assembling collections of smaller parts. ● Individually and collaboratively decompose a problem and create a sub-solution for each part (e.g., video game, robot obstacle course, making dinner).
Pattern Recognition	● Identifying patterns and commonalities in artefacts. ● Identify common patterns and similarities between older and newer problem-solving tasks and use sequences of instructions to solve a new problem.	● Identifying patterns and commonalities in artefacts. ● Remix and reuse (by extending if needed) resources previously created. ● Adapting solutions, or parts of solutions, so they apply to a whole class of similar problems. ● Transferring ideas and solutions from one problem area to another.	● Identifying patterns and commonalities in artefacts. ● Remix and reuse (by extending if needed) resources previously created. ● Adapting solutions, or parts of solutions, so they apply to a whole class of similar problems. ● Transferring ideas and solutions from one problem area to another.
Evaluation	● Recognize when instructions do not correspond to actions. ● Remove and fix errors. ● Assessing that an artefact or solution is fit for purpose.	● Recognize when instructions do not correspond to actions. ● Remove and fix errors. ● Assessing that an artefact is fit for purpose. ● Assessing whether the solution is effective and efficient. ● Shows awareness of tasks best completed by humans or computers.	● Recognize when instructions do not correspond to actions. ● Remove and fix errors. ● Assessing whether the solution is effective and efficient. ● Identifying ways to improve solutions or information quality. ● Selecting and justifying appropriateness, precision, or quality of "best" solutions and information sources.