Caribbean Curriculum
Volume 29, 2022

COMPUTATIONAL THINKING FOR ALL: A New Skill for the Digital Age

Vimala Kamalodeen

Computational thinking has been associated with Computer Science and Mathematics to foster higher order thinking for solving problems. However, the need for problem solving skills that can address the requirements of an ever-increasing digital world has heralded the promotion of computational thinking across all disciplines. The Caribbean, like its global counterparts, have accepted that problem solving is a 21st-century skill that must be taught to this generation of students. This is especially critical today when the main tools of communication, collaboration, teaching, and learning require a computer. This paper discusses computational thinking and argues for its promotion at all school levels. The paper explores the debates around the definition of computational thinking and describes its growth locally and internationally. Two examples are presented to illustrate how teachers in Trinidad and Tobago have taught computational thinking in their classrooms using a strategy of game-based learning. These empirical examples highlight ways that computational thinking can be promoted across different school types to support efforts for a more equitable curriculum that is relevant and meaningful to diverse students. Implications are discussed within the framework of science, technology, engineering, the arts, and mathematics (STEAM) education and recommendations made for future research.

Introduction

It is well established that computers dominate every aspect of our daily lives. From teaching and learning to rudimentary tasks of booking a restaurant table require the use of a computer; thus, the ubiquity of digital technologies has become apparent. The computer evolved from a large cabinet in a room at the National Aeronautics and Space Administration (NASA) to a sleek palm-held device, available to all peoples across the globe. To succeed in this complex digital world, persons need skills and competencies not previously identified. Managing information, effective communication and frequent collaboration call for an understanding of how to perform daily, routine tasks, efficiently. Like reading and writing, computation, or reckoning is necessary to live and function well. Even further, harnessing the power of the computer to perform routine tasks and solve complex problems is key to living in a digitised world. Thus, computational thinking (CT) is an essential skill for the digital age for efficient information management (Shute et al., 2017) and focused problem solving (Korkmaz et al., 2017; Selby & Woollard, 2013; Wing, 2006).

Since Jeannette Wing made the clarion call for computational thinking for all in 2006, stating that "to reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability" (p. 33), research and debate have exploded on the meaning of computational thinking and its role in school curricula. A simple Google search on computational thinking will reveal millions of results across all media types. Many of the entries focus on definition, instructional strategies and their central role in problem solving. According to the International Society for Technology in Education (ISTE), computational thinking is the highest order of problem solving and is considered as a cross-curricular skill (Valenzuela, 2020). However, computational thinking has been popularly located within the discipline of Computer Science (CS) related to coding and programming, often leading students and teachers alike to consider it elitist and inaccessible. In Trinidad and Tobago, the inclusion of CT in the revised national Information and Communications Technology (ICT) curriculum for lower secondary placed greater focus on learning with, through, and about ICT. A stated ICT curriculum goal is to develop problem solving and computational thinking skills in younger secondary students (Ministry of Education of Trinidad and Tobago, 2014). As such, computational thinking is identified as a transversal skill to master routine problem-solving tasks and for building capacity to solve complex non-routine challenges.

Going forward, the paper discusses varying definitions of computational thinking, describes its core components as is commonly taught, and locates CT within Science, Technology, Engineering and Mathematics (STEM) education for broadened access. Two relevant examples of studies in different Trinidad classrooms are presented, followed by a discussion of implications for educators, curriculum specialists, and researchers. The paper serves to bring the topic of computational thinking to the fore in educational debates in the Caribbean, where literature is almost non-existent.

Development of Computational Thinking

What is CT?

Programming skills have been known to be a predictor of success in Computer Science, a subject area increasingly considered for promotion in a technological and scientific society. The field of Computer Science offers a way of conceptualising and solving complex problems from a new perspective using Computational Thinking, an essential skill needed for programming. Following on the early work of Seymour Papert, Jeanette Wing (2006) asserted that Computational Thinking was about "problem-solving, designing systems and understanding human behavior using concepts that are fundamental in computer science" (p. 33).

Based on Wing's early definition, the literature has been rife with debate over an operational definition, especially in relation to education (Aho, 2012; Grover & Pea, 2013; Hu, 2011, Shute et al., 2017). After all, computation is a core concept in mathematics. But Hu (2011) debated the nuances of the differences between computation as a process to produce output from input and computational *thinking* as an approach to thinking about solving problems that relies on algorithmically constructed models. Grover and Pea (2013) suggested a useful description of CT based on the seven "big" ideas of computing from the National Science Foundation (of the USA). They are:

> 1. Computing is a creative human activity
> 2. Abstraction reduces information and detail to focus on concepts relevant to understanding and solving problems
> 3. Data and information facilitate the creation of knowledge
> 4. Algorithms are tools for developing and expressing solutions to computational problems
> 5. Programming is a creative process that produces computational artifacts
> 6. Digital devices, systems, and the networks that interconnect them enable and foster computational approaches to solving problems
> 7. Computing enables innovation in other fields, including science, social science, humanities, arts, medicine, engineering, and business. (p. 39)

Various organizations in the United States of America (USA) such as the Computer Science Teachers Association (CSTA) with the ISTE have further clarified the concept of computational thinking as having the following components: problem decomposition, data collection, analysis and visualisation, abstraction, algorithms and procedures, automation, simulation and modelling and the simultaneous execution of tasks. Barr and Stephenson (2011) developed a checklist of core concepts and competencies in CT relevant for K-12 teachers, while others, such as

Grover and Pea (2013) and Selby and Woollard (2013), added other components. The elements of CT provided by Grover and Pea (2013) and Korkmaz et al., (2017) are useful for this paper. These elements are abstraction (reduction of unnecessary detail), generalisation (finding patterns/ pattern recognition), decomposition (structured problem modularising and break down), debugging (systematic error correction), and algorithmic thinking (a way of finding a solution using a sequence of steps). Table 1 summarises the various elements with a simple explanation of each component.

It is also important to highlight that CT encompasses attitudes such as persistence, perseverance, confidence, and communicative ability (Barr & Stephenson, 2011; Weintrop et al., 2016) as well as tinkering—especially for younger children (Bers et al., 2014). These attitudes have been developed in programming aspects of CS for many decades. However, it must be highlighted here that while programming lies at the heart of CS, CT is mostly about problem solving (Korkmaz et al., 2017) and can be considered more universally in education, much like literacy and numeracy (Yadav et al., 2014).

Table 1. Core Elements of Computational Thinking

Element	Definition
Abstraction	Filtering out extraneous information and getting to the core of the problem
Algorithmic Thinking	Developing a series of steps to produce a desired solution
Decomposition	Breaking down of larger or complex tasks into smaller manageable parts
Generalisation	Finding a broader classification for a group of problems or concepts
Debugging	Identification and fixing of errors

CT in STEM Education

Many of the efforts to push CS in schools began with coding, but more recently these have shifted to computational thinking (Garvin et al., 2019; Yadav et al., 2018). The goal of CT is to find solutions to varied complex problems that are based on the core elements of Computer Science. However, there are distinctions and connections between the terms computational thinking and computer science that need clarifying. Broadly speaking, computational thinking is a part of computer science and leads to the development of program design and programming; however, it is essentially a skill set and a pathway to higher order thinking

(Burke et al., 2020) and problem solving (Korkmaz et al., 2017), which are essential goals of STEM/STEAM.

It has been argued that Computer Science is a science as it draws on fundamental principles of systematic design and experimentation in the field of computers (Denning, 2005). Computer Science follows the Francis Bacon scientific paradigm of hypothesis formation and testing and using models for prediction. Computer Science as a discipline relies on Science, Engineering and Mathematics (SEM). Computer Science as a subject discipline has been offered at secondary and tertiary levels as General Certificate of Education (GCE) Advanced and Ordinary levels since 1992 in the Caribbean. The Caribbean Examinations Council (CXC) introduced Information Technology (IT) at the Caribbean Secondary Examination Certificate (CSEC) level in 1994 and Information Technology, Computer Science and Digital Media at Caribbean Advanced Proficiency Examinations (CAPE) levels later. Computer Science is the science of information processes and spawns many careers in specialised fields such as programmers, database administrators, software engineers, data analysts, and computer scientists. It is suggested that jobs in this sector will grow by 13% by the year 2030 in the USA by the US Bureau of Labor Statistics (2021), with the fastest growth in all occupations.

However, CS suffered in the K-12 system as it was not offered in US schools until recently. In 2010, Wilson et al. made a searing attack on the US K-12 education system with a report entitled *Running on empty: The failure to teach CS in the digital age.* The report revealed very low numbers of females in the discipline and in industry, and the low levels of knowledge of computers in the US. The gender gap in the USA is noted by several researchers (for example Espino & González, 2015; Roberts et al., 2002), but not necessarily in other countries like Malaysia where females in undergraduate CS programs outnumber males (Othman & Latih, 2006). The predicament in the US led to major importation of skilled persons from India and China. Concerns by policy makers and education stakeholders led to them finding a way to address the issue of Computer Science education. This aim at finding a solution was buoyed by Wing's (2006) influential article in the *Communications of the ACM* where she suggested that Computational Thinking (CT) allows students to think like computer scientists. This notion emphasises computational thinking as a way to process and problem solve and does not have programming as a requirement to engage with it. As such, CT is "a formative skill like reading, writing and arithmetic" (Lu & Fletcher, 2009, p. 260), and is needed as a literacy tool for the digital age. These authors emphasise that CT is "not about getting humans to think like computers, but rather about developing a set of mental tools necessary to use computers to solve complex human problems" (p. 260).

Due to the universal applicability of CT to many disciplines and its heavy focus on problem solving and analytical skills, CT has been applied in STEM fields but less so in STEM education (Swaid, 2015). Some of these applications are in mathematics, physics, and environmental science and even music. Computer Science has no monopoly on the CT skill set; rather, "computational thinking is a way humans solve problems" (Wing, 2006, p. 35). Research suggests that including CT into STEM education helps STEM goals to reach the widest possible audience, builds higher level thinking skills, and provides preparation for STEM careers (Jona et al., 2014; Romainor et al., 2018). In more recent literature, CT is conceptualised as a discipline thinking practice as well as a trans-disciplinary thinking practice (Li et al., 2020; Weintrop et al., 2016). The transferability of knowledge and skills allows the integration of CT into existing disciplines and across STEM/STEAM/STREAM (science, technology, reading and wRiting engineering, the arts, and mathematics) fields. This integration can facilitate greater access to what might be termed the "r" or 'rithms literacy' (short for algorithms) in the current digital era (Grover & Pea, 2013, p. 40). CT is arguably at the core of all STEM disciplines (Jona et al., 2014; Li, 2020; Swaid, 2015).

CT in the Curriculum

Many global educators and stakeholders have been persuaded to include CT in their curriculum as they foresee the usefulness of this skillset in the future (Gavin et al., 2019). In the United States, collaborative efforts with the computer science education community across the country were commenced to initiate talks and guidelines for computer science education (K–12 Computer Science Framework, 2016; Barr & Stephenson, 2011; Woollard, 2016). Similar reform is taking place in Nordic and other European countries (Berge, 2017). Here in Trinidad and Tobago, a twin island state in the Caribbean, the need to develop computational skills in students has been acknowledged by educational stakeholders. Some reasons put forth to support CT in schools are the ease of entry in teaching computational ideas and the transdisciplinary nature of the field (Yadav et al., 2018) and the emphasis on problem-solving (Butler & Leahy, 2020). Yet others emphasise its relationship in promoting competencies in other subject areas such as algebra and informatics (Mindetbay et al., 2019). What is less argued is its importance in 21st-century curricula and the preparation and professional development needs of teachers to support such drives (Garvin et al., 2019). Teacher knowledge of CT is fundamentally important for successful teaching (Yadav et al., 2017).

The teaching of CT is a key feature of the revised National Certificate of Secondary Education (NCSE) Information and Communication Technology (ICT) curriculum (2014). This curriculum was designed for students at the Forms 1-3 levels (US grades 6-8). In it, one of the stated goals is "to develop computational thinking skills in students to assist them in being able to cope with daily challenges" (NCSE ICT, 2014, p. 29). The curriculum also specifies a Module on "Programming Concepts and Computational Thinking" (p. 31) as one of six modules of the curriculum. This curriculum change allows teachers to intentionally focus on building skills in CT; however, there is little or no research on teachers' pedagogical strategies nor secondary student experiences or achievements in CT, especially in a Caribbean context.

CT in Diverse School Settings

There is great debate about student performance in CS and why girls seem less confident to select courses therein as a major. Several challenges to success have been purported such as age, gender, experience, negative stereotypes, mathematical ability, and attitude (Guenaga et al., 2021; Margulieux et al., 2020; Rowntree et al., 2004; Vitores & Gil-Juarez, 2016, Sax et al., 2017). Of these, negative stereotypes and the classroom experience are significant determinants' of girls' academic choices (Beyer, 2014). Further, students often select programs and courses in middle school (North American settings) but need to feel confident in their choices. As such, experiences in school settings are crucial as well as being exposed to female teachers in the CS field (Beyer, 2014). As for women, it is also particularly true for underrepresented communities; science, technology, engineering, the arts, and mathematics students' experiences with success in CT are critical to their consideration of CS and related careers.

The dialogue among educators and stakeholders as to possible teaching strategies to develop CT in students lies in its early introductions—at primary and early secondary education levels (Butler & Leahy, 2020; Tsarava et al., 2017). Extant research in CS has mostly focused on female enrollment and attrition at tertiary level, but recently, work among younger learners has taken centerstage. This places the teacher and his/her pedagogy as critical to student learning experiences and success.

Many interventions have surfaced to teach computational thinking to students, each with its unique strengths and limitations. These include the use of programming tools, robotics, and game-based learning (Shute et al., 2017). The use of games has shown some promise in promoting CT in students (for example, Berland & Lee, 2011; Hooshyar et al., 2021; Kazimoglu et al., 2012; Sharma et al., 2019; Turchi et al., 2019). In particular, game-based learning is a popular instructional approach to improving CT skills (Shute et al., 2017). Plugged and unplugged resources in game-based learning (Kamalodeen et al., 2021) to promote CT (Caeli & Yadav, 2020) are useful and have become increasingly popular in recent times to help teachers pivot between online and offline resources in low technologically serviced school settings (Kalloo et al., 2019). According to Tsarava et al. (2017), "Playful plugged activities," such as board games, allow students a first positive experience with CT (p. 68), and address core STEM areas (Kalloo et al., 2019).

Examples of studies focused on developing CT in secondary schools

Two action research samples in schools in Trinidad and Tobago are selected to highlight key achievements and strategies teachers used to promote computational thinking in their classrooms. Action research is considered appropriate as a systematic inquiry (Zuber-Skerritt, 2001) of teachers into their practice. Specifically, educational action research allows for non-heterogenous perspectives across a range of school contexts (Keiny & Orland-Barak, 2009). Two projects are described, the first in a large rural government secondary school, and the other in a smaller urban all-girls government-assisted (denominational) school that were supervised by me. I regard the two cases as successful examples of curricula adaptation and innovation, one among low SES students, and the other focused on girls. These two cases highlight the ways that underserved populations can be included in higher level thinking curricula, through strategic planning and adaptable implementation.

Example 1: A Game-Based Learning Intervention With CT in a low SES School

Design. The classroom action research study took place in a seven-year large rural government school and is considered a low SES school. The selected class was first form with eight (8) students: three boys and five girls between the ages of 12 and 13 years with low SEA[1] scores. The classroom teacher was a female university graduate doing the PGDipEd in IT at the university. The CT component of the ICT syllabus of the Republic of Trinidad and Tobago was relied upon to build the unit of work.

---

[1] The SEA or Secondary Entrance Assessment is a national test for primary students to place them in secondary school, in Trinidad and Tobago

The quasi-experimental design of the study used a one-group intervention. The purpose of the intervention was to determine levels of student achievement of CT and observe their attitudes to engaging in CT activities.

The pedagogical strategy utilised a gamified environment in addition to digital game-based learning. A gamified course was developed to motivate and engage students through nine lessons that focused on the key CT components in Table 1 above. The course used quizzes to assess student understanding of CT. The unit of work ended with a 3-day independent study marathon on course 2 of *code.org*[2] focused on sequence, loops, conditionals, and creativity. Games in the unit were selected from *mindgames.com* and *code.org* as they promoted problem solving skills in a game format that was time sensitive and allowed for computational thinking. The teacher used these games as assessments for learning.

The teacher engaged in personal professional development in order to enact this action research. The teacher made a comprehensive plan for teaching CT to first formers (Yadav et al., 2018). The quasi-experimental study collected data using diagnostic assessments, quizzes, polls and results from *code.org* and from teacher blog reflections of the action research.

Findings. Findings are presented based on student performance in the course, then game achievement, followed by the teacher's own perceptions of students' attitudes to the CT lessons. First form students demonstrated substantial knowledge of CT in the five components as seen in Table 2. Mean scores above 75% were seen in two components: decomposition and abstraction while mean scores were above 50% in all components. The teacher felt that the quiz scores "could have been better… but [I felt] that at the end of the sessions, I am saying that my students are really good at this" (Wight, 2019, p. 38).

Table 2. Summary results for CT across selected components among first formers

---

[2] Code.org provides curriculum for K-12 computer science in the USA and is used globally

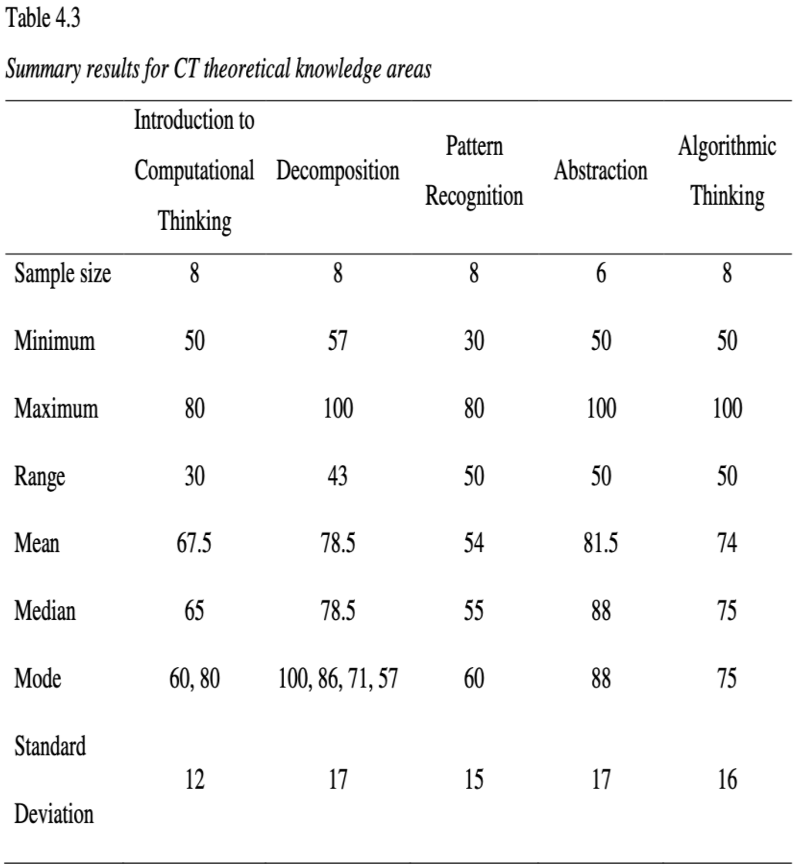


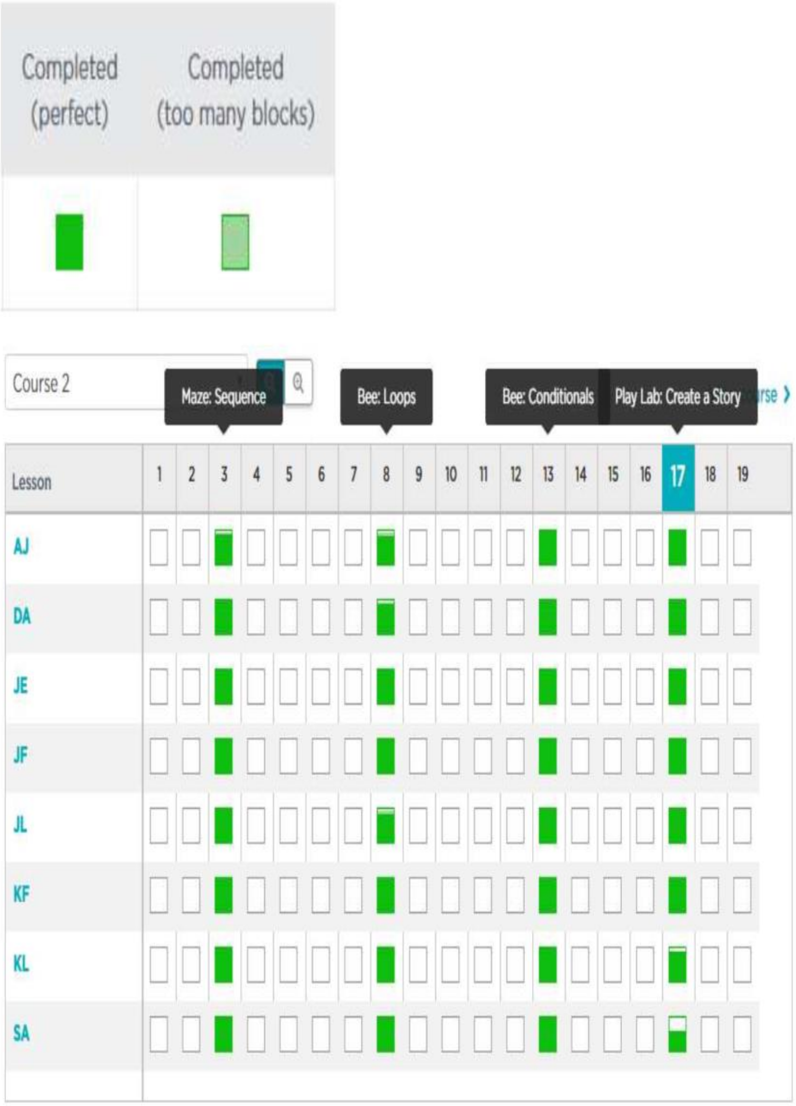
Table 4.3

Summary results for CT theoretical knowledge areas

	Introduction to Computational Thinking	Decomposition	Pattern Recognition	Abstraction	Algorithmic Thinking
Sample size	8	8	8	6	8
Minimum	50	57	30	50	50
Maximum	80	100	80	100	100
Range	30	43	50	50	50
Mean	67.5	78.5	54	81.5	74
Median	65	78.5	55	88	75
Mode	60, 80	100, 86, 71, 57	60	88	75
Standard Deviation	12	17	15	17	16

Note. "Developing computational thinking among form 1 Information and Communication Technology (ICT) students in a rural government secondary school in Trinidad," by B. Wight, 2019, p. 31, Academia.edu.

Students performed exceptionally well at the coding activities at the end of the unit from *code.org*. The games were *maze sequence, bee loops, bee conditionals* and *Playlab: Create a Story*. Figure 1 shows the completion status across the four games. All 8 students attempted the 4 games, 8 out of 8 completed the first 3 games and 6/8 completed the last game. Three students progressed to create an animated story using block-based coding (see Figure 2). Those who completed the games were awarded certificates through *code.org*.



Completed (perfect)	Completed (too many blocks)
🟩	▨

Lesson	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
AJ	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
DA	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
JE	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
JF	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
JL	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
KF	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
KL	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐
SA	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	☐	🟩	☐	☐	☐	🟩	☐	☐

Figure 1. Completion Status of all 8 Students in the Form One Class Doing CT

Note. "*Developing computational thinking among form 1 Information and Communication Technology (ICT) students in a rural government secondary school in Trinidad*," by B. Wight, 2019, p. 32, Academia.edu.

```
set actor 3 ▾ to the bottom right ▾ position
set actor 3 ▾ to a unicorn ▾ image
play win point 2 ▾ sound
actor 1 ▾ say " wow! a unicorn. we can use it to get to the city faster princess. "
wait for 2 seconds ▾
set actor 2 ▾ to a happy ▾ mood
actor 2 ▾ say " good idea brave knight! "
```

Figure 2. Animated story created by 3 students who advanced beyond the 4 games from code.org.
Note. "*Developing computational thinking among form 1 Information and Communication Technology (ICT) students in a rural government secondary school in Trinidad*," by B. Wight, 2019, p. 36, Academia.edu.

Poll data indicate that all students enjoyed the challenges, felt that the activities were meaningful to them, and perceived their problem-solving skills to be improved. Students volunteered to come to classes during their lunch break and persevered in completing tasks. Students displayed multiple solutions to problems and often offered creative solutions.

The teacher reflected on students' willingness to participate despite possessing no prior knowledge of CT and the difficulties of engaging in higher-level thinking activities. The teacher faced several challenges in completing the unit. One of them was the reading and vocabulary levels of the students. For example, the teacher had to teach the vocabulary terms of 'algorithm' and 'abstraction'. Additionally, the school often closed abruptly, and lessons were curtailed, so the teacher had to quickly adapt her unit and lesson plans to complete in a shorter period. She used her lunch periods to accommodate students in completing tasks. Using a reflective stance, the teacher noted her challenges and successes. Wight (2019) wrote, *"Usually I will say, I'm getting better at this... at the end of these sessions, I am saying, 'My students are really good at this.'" (p. 124).* ***Discussion.*** The combined data from the lesson assessments, polls, teacher blog and game metrics seem to indicate that first form students made some gains in CT knowledge and skills. The knowledge test was aligned to key components of CT using a table of specifications and show at least a basic knowledge base (mean score >50% in all components). It was more difficult to assess the knowledge gains by component in the digital games. Measures of persistence and completion rates were used to signal students' attitudes to CT with 75% (6 out of 8) of students completing the course of 4 games. The component of CT—pattern recognition or generalization—seemed to have the lowest performance score. Students seemed to perform better at the digital game activities than the knowledge test during the course and perhaps that has to do with the motivational affordances of games and the excitement that they can bring to players (Bachu & Bernard, 2011; Papastergiou, 2009). However, in this study, motivation levels were not measured just observed.

There were no observable differences between boys and girls in engaging in the activities. These are similar findings to Papastergiou (2009) who used digital game-based learning for promoting CT among high school students. These findings are similar to those of primary children using game-based approaches (Author, 2019). The literature is replete with the connection between enjoyment and learning, and particularly so when children feel successful while doing coding activities (Sharma et al., 2019). Additionally, students accelerated through the challenge levels and moved from game to game achieving what Bachu and Bernard (2011) call fluency.

Implications. The study was limited as it was a small-scale classroom intervention with teacher designed assessments. Additionally, the study did not link the CT components taught in the course in the 9 lessons with the CT skills demonstrated through the games. As such, it is unclear how the games at the end of the unit from *code.org* were aligned to individual CT components or to student achievement in CT. This has been an ongoing challenge for teachers due to the nascent development of usable CT assessment frameworks (Shute et al., 2017).

Nonetheless, the pedagogical strategy used in this study seemed to have promoted enjoyment and excitement among 12–13-year-old girls and boys alike at the selected low SES school. While certain vocabulary terms were difficult for students to read and concepts challenging to grasp, the design of the CT plan kept students engaged. The lack of a gender differential is interesting, but the study is too small and brief to make any inferences about the gender gap, if any, in computational thinking. The study also gives insights into how instrumental the teacher is in making any curriculum innovation or adjustment work. The teacher engaged in professional development and joined an online community to gain a wealth of resources and expertise to teach this unit. The role of the teacher in designing units of instruction is deemed to be important as careful selection of games is needed to bring challenge and accomplishment in the classroom (Beyer, 2014).

A serious drawback to the use of digital game-based learning in the school was the severe disruption to classes due to the lack of electricity or internet connectivity. As such, students did not get to complete tasks and did not have the facilities at home either. The teacher offered lunch break supervised sessions for the affected students to complete. This break in classes is common in rural schools. This is a caution for using online activities in schools like these and suggests that the use of unplugged resources (Kalloo et al., 2019) can mitigate these challenges.

Example 2- A Game-Based Learning Intervention with CT in an All-Girls Urban School

Design. The classroom action research study took place in an all-girls denominational school. The selected class was one fourth form with 28 girls who had chosen CSEC IT as a subject choice. CT is not a component of the syllabus. The quasi-experimental design of the study used a one-group pre-test post-test intervention (see sample question from the test in Figure 4). The classroom teacher was a female graduate with PGDipEd in Mathematics. The students' computational thinking levels were measured before the intervention using a pre-test. The intervention involved using four purposefully selected games (Papadakis, 2020), plugged and unplugged (Hsu & Liang, 2021; Papastergeriou, 2009). The games met all the components of computational thinking—abstraction, decomposition, generalization, algorithmic thinking and debugging (see table 1). The games were *LightBot* (plugged) (see Figure 3), Cryptograms (unplugged), *Minesweeper* (plugged), and zebra puzzles (unplugged). Each game was then introduced to the students sequentially. After the three-week intervention, the students were tested again using a post-test, and the results were compared with the pre-test results. Overall, the study collected data from pre- and post-tests, questionnaires, journals, and game blogs. A two-tailed t-test was used to test performance change in the pre- and post-test intervention, and in-vivo and values coding were used to generate thematic statements from the journals and blogs. Subscales from the questionnaire guided the coding process. These datasets were integrated for triangulation and content validation.

Findings. Pre- and post-test data analyses showed the increase in levels of computational thinking was statistically significant. Therefore, there was a significant difference in the scores for the post-test (M=22.29, SD=6.25) and the scores for the pre-test (M=16.78, SD=4.91); t(27) = -4.023, p< .001. Overall, there was improvement in each of the five CT components (see Table 1) between the pre- and the post-test. However, individual students showed three different types of performance: they either improved, stayed the same, or got a reduced score. These results do not signal clear improvement in CT levels based on the intervention strategy. Based on performance in the post-test, students were ranked as high, medium and low achievers. To interrogate how students in each of these three categories felt about the intervention, selected student journals were interrogated. The teacher selected 6 journals, two from each category. Feedback from a high achiever suggested that she liked the "challenge in the game and enjoyment", whereas feedback from a low achiever reflected negative emotions of non-enjoyment, tedium, challenge, and frustration when playing *Lighbot*. Overall, the female students expressed a range of emotion from 'real fun' to 'frustrated'.

Figure 3. Screenshot of the Lighbot game showing visuals for commands.

Note. From "Using game-based learning to promote computational thinking levels in fourth form girls at a high-achieving secondary school in Trinidad," by S. Ramdass, 2019, p. 120. The University of the West Indies, St. Augustine

Aligned to each level of CT, a sample of students' thinking is presented as written in their journals. Table 2 shows that for each skill level, the high achievers were able to make their thinking visible through writing. Various techniques, strategies and procedures were elicited. The missing sections in Table 2 for low achievers reveal that students who had difficulty did not write down their thinking.

Table 2. Comparison of comments from high and low achievers while doing CT

CT skill	High achiever comment	Low achiever comment
AT	"... come up with creative ways to use the different commands to solve the given tasks step by step."	"My first steps in solving the puzzles was reading the hints and trying to put in the information that they give you and things that I was sure about."
Decomposition	"When the level became more complex I had to work each one out in parts."	"Start with boxes near the ones and twos to easily narrow it down and tackle smaller bombs before attempting the larger ones."

Abstraction	"I developed strategies in the game that helped me move from one level to the other."	
Generalisation	"However, the levels got more challenging as the game progressed. This forced me to utilise the concepts they learnt earlier on in different ways to complete the given tasks."	
Debugging	"To solve this problem, I ran the program to see where I went wrong and corrected my errors accordingly."	"I had to do some trial and error with some parts or even put in code as I ran it."

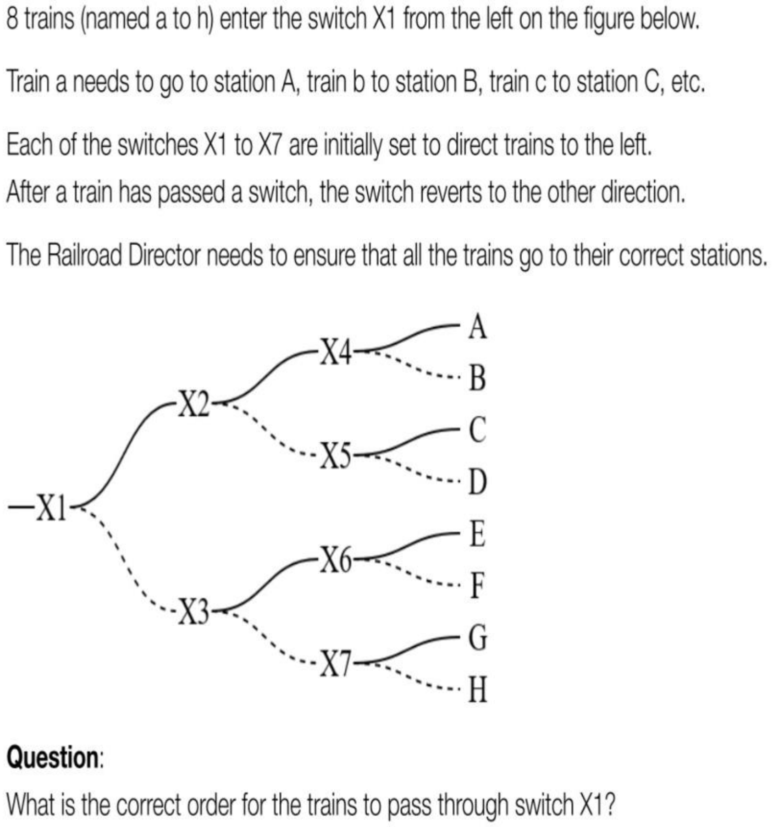
Note: Adapted from Ramdass (2019)

Discussion. The data collected from both quantitative and qualitative instruments showed that students had improvements in all CT components—abstraction, debugging, decomposition, generalisation and algorithmic-thinking, but the lowest change was in abstraction. Students showed evidence of describing the use of computational thinking components when playing the games. Accompanying journal entries of the low, medium, and high performers indicate that students who were able to interact with the games deeply (as evidenced by their articulation of the CT skills) had higher test scores. These changes in CT levels were also noted in similar studies done by Berland and Lee (2011) and Kazimoglu et al. (2012), where they both examined the appropriateness of games as a teaching strategy to promote complex and higher-order skills.

A key finding of the study is the capturing of girls' experiences at different levels (low, medium, high), which is the subject of current research. The findings show that students experienced positive emotions when playing the games only when the students were appropriately challenged. When the challenges posed in the games were achievable, students were able to move through the levels in the games, and they felt a sense of accomplishment. Furthermore, students stated that they experienced feelings of joy and fun when this occurred. However, when the challenges proved too difficult for the students to solve, or required mental effort, they became frustrated and confused as they could not find a winning game plan. As such, they felt frustrated, anxious and disquieted—emotions identified by Dahn and DeLiema (2020) in a study of three girls as they coded.

Whether this frustration is unique to girls provides opportunities for further studies. Regardless of gender, the need to research students' experiences with computational thinking can provide explanations of performance and attainment, thus, paving the way for improved student outcomes. While the study did not distinguish the types of challenge that girls faced, it did identify how students felt while playing games, while all of them increased in computational thinking.

Implications. This study combined plugged and unplugged resources for teaching CT, and the teacher attempted to align the CT components to the game. This study would have been improved if a reliable CT assessment framework was used. Results showed evidence of improvement through pre- and post-test data and understanding of CT. When a game-based learning strategy is utilized, it can provide educators with information to make instructional decisions. Moreover, student-centred approaches to teaching CT, allowing for ample reflection and communication of ideas, thoughts, and feelings, can benefit girls. In computer science contexts, failure and frustration has been well documented, and if we are to tip the balance in favour of greater female access to Computer Science and STEM education and careers (Vitores & Gil-Juarez, 2016), we must look carefully at girls' early experiences with computational thinking.



8 trains (named a to h) enter the switch X1 from the left on the figure below.

Train a needs to go to station A, train b to station B, train c to station C, etc.

Each of the switches X1 to X7 are initially set to direct trains to the left.
After a train has passed a switch, the switch reverts to the other direction.

The Railroad Director needs to ensure that all the trains go to their correct stations.

Question:
What is the correct order for the trains to pass through switch X1?

Mark only one

○ acedfgh
○ aecgbfd
○ adcgbfe
○ agcdbfe

Figure 4. One Sample Question from the Test Given to the Female Students

Question 6

Note. From "Using game-based learning to promote computational thinking levels in fourth form girls at a high-achieving secondary school in Trinidad," by S. Ramdass, 2019, p. 87, The University of the West Indies, St. Augustine.

Discussion and Implications of CT for all

The two studies in this paper show that CT can be included in secondary curricula in the local context. The two school contexts were diverse: one a government-assisted denominational all-girls school in an urban environment and the other a government 7-year mixed-sex school in a rural area. Although neither one is a large-scale study, the selected studies actually provide insights for future studies and for application of CT curricula and pedagogy in similar settings.

Findings from both studies showed improvements in computational thinking. Similar studies where plugged and unplugged resources using game-based approaches within classrooms show improvement in CT abilities and skills at primary level (Hooshyar et al., 2021; Hsu & Liang, 2021; Tsarava et al., 2017) and at secondary level (Looi et al., 2018; Rodriguez et al., 2017; Papadakis, 2020). Using a game-based approach seems to dominate the literature to promote CT due to the motivational and adaptive affordances of game-based approaches (Kamalodeen et al., 2019; Hooshyar et al., 2021; Papadakis, 2020). In particular, the selection of unplugged resources with plugged (digital) ones allows teachers in low resourced schools to still teach CT successfully (Huang & Looi, 2021), if schools are not digital ready. The strategy can inform curriculum developers and practioners alike about making content and teaching strategy decisions, especially when designing core components of national curricula. Both studies suggest games need to be selected carefully, and that alignment of the games and activities to CT components can bring stronger evidence of student understanding. An assessment framework for CT is needed (Grover & Pea, 2018; Shute et al., 2017), and more research is warranted here.

In computer science contexts, the attitudinal factors of perseverance and persistence are well documented, and problem-solving skills are not easy to teach or develop or even master. Computational thinking allows for this higher order thinking skill to be broken down into smaller components (Korkmaz et al., 2017), thus, making it easier to measure. The students in the selected studies ranged from ages 12-16. For girls to select CS and STEM careers, we must look carefully at girls' early experiences with computational thinking (Yadav et al., 2018). The literature on the gender gap in Computer Science (Sax et al., 2017), and in STEM generally, abounds, but this paper serves to highlight not only successful students but also successful teachers who were female. These two cases are not unlike those described of three successful girl coders by Dahn and DeLiema (2020). In both cases, the two female teachers did not perceive that they were in a masculine field, a similar finding by Othman and Latih (2006) in Malaysia.

The role of the teacher in designing units of instruction is deemed to be important as careful selection of games is needed to bring challenge and accomplishment in the classroom (Beyer, 2014). Further, eliciting ways to make student thinking visible (through blogs and journals) takes time and effort but allows students' emotions to be seen. Girls' confidence in problem solving and embracing a growth mindset is rewarding and needed for them to be successful. These innovative elements allow us to envision new classroom practices in Computer Science which, according to Dahn and DeLiema (2020), place youth in positions of power. While this paper focuses on two samples at secondary level, there is a need for similar work at primary and early childhood levels.

Teacher preparation and education programs must place emphasis on the concepts of CT as teachers are being asked to integrate CT in existing curricula globally. There is a need for effective instruction and strategies to teach this critical 21st-century skill at primary and secondary levels (Garvin et al., 2019; Yadav et al., 2017). While a number of efforts have been seen globally and locally to incorporate it in the curriculum (Yadav et al., 2018), research at every level is still in its infancy. Existing literature argues for the early introduction of CT at primary and secondary levels are most effective for CT, but empirical research is lacking (Butler & Leahy, 2020). A pedagogical framework that involves unplugged resources and tinkering, and exploration can help to guide teachers' work (Kotsopoulos et al., 2017). As research on computer science at higher education still dominates the literature, there is some opportunity for useful frameworks to emerge and for teacher educators to plan and design effective programmes for CT. This paper argues that even small-scale classroom action research projects that combine theory and practice can provide evidence and useful insights for design of teacher education and professional development programmes. Thus, modules and courses can be developed in non-decontextualised contexts using inter and trans disciplinary approaches.

Conclusion

This paper contributes to the discourses on computational thinking as a core curriculum concept and a relevant skill for the 21st-century learner. It is suggested that CT can be placed in STEM/STREAM education where access and equity concerns of underserved populations are the focus. To understand the critical role of CT, its relationship to problem solving competency development and transdisciplinary nature were discussed, even as its clear beginnings and relationships with programming and Computer Science remain entrenched. I suggest that CT should gain its own identity as a 'rithms or r' literacy and be developed as a core skill for every learner at all educational levels, starting from primary. I further detailed examples of how teachers crafted CT into their existing subject curricula in two diverse school settings. They both used game-based approaches, which are literature-supported. It is noteworthy that the two teachers are female in what is considered globally as a male-dominated field. The two examples demonstrate that CT can be taught to all students, regardless of SES and gender or even age. Finally, the paper provides guidance and insights for practitioners, curriculum developers, teacher educators and educational researchers.

References

Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, *55*, 832–835.

Bachu, E., & Bernard, M. (2011, April). Enhancing computer programming fluency through game playing. *In Proceeding of the 4th Annual International Conference on Computer Games, Multimedia and Allied Technology* (CGAT 2011).

Barr, V., & Stephenson, C. (2011 March). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48-54.

Berge, O. (2017). Rethinking digital literacy in Nordic school curricula. *Nordic Journal of digital literacy*, *12*(01-02), 5-7.

Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, *1*(2), 65-81. doi:10.4018/ijgbl.2011040105

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157.

Beyer, S. (2014). Why are women underrepresented in Computer Science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS course-taking and grades. *Computer Science Education*, *24*(2-3), 153-192.

Burke, Q., Roschelle, J., Bailey, C., Angevine, C., Weisgrau, J., & Mills, K. (2020, March). Examining teacher perspectives on computational thinking in K-12 classrooms. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)* (Vol. 1, pp. 1-7). IEEE.

Butler, D., & Leahy, M. (2020). Developing pre-service teachers' understanding of computational thinking: a constructionist lens. *British Journal of Educational* Technology, 52(3), 1060-1077.

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, *64*(1), 29-36.

Code.org. (2022). *Introducing CS connections-Hour of code activities*. https://code.org/hourofcode/overview

Dahn, M., & DeLiema, D. (2020). Dynamics of emotion, problem solving, and identity: Portraits of three girl coders. *Computer Science Education*, *30*(3), 362-389.

Denning, P. J. (2005). Is computer science?. *Communications of the ACM*, *48*(4), 27-31.

Espino, E. E. E., & González, C. S. G. (2015, September). Influence of gender on computational thinking. In *Proceedings of the XVI International Conference on Human Computer Interaction* (pp. 1-2).

Garvin, M., Killen, H., Plane, J., & Weintrop, D. (2019, February). Primary school teachers' conceptions of computational thinking. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 899-905).

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

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, *19*, 1257-1258.

Guenaga, M., Eguíluz, A., Garaizar, P., & Gibaja, J. (2021). How do students develop computational thinking? Assessing early programmers in a maze-based online game. *Computer Science Education*, *31*(2), 259-289.

Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M., & Lim, H. (2021). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior*, *114*.

Hsu, T. C., & Liang, Y. S. (2021). Simultaneously improving computational thinking and foreign language learning: Interdisciplinary media with plugged and unplugged approaches. *Journal of Educational Computing Research*, *59*(6), 1184-1207.

Hu, C. (2011, June). Computational thinking: What it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 223-227).

Huang, W., & Looi, C. K. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, *31*(1), 83-111.

Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014, January). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In *future directions in computer science education summit meeting, Orlando, FL*.

K–12 Computer Science Framework. (2016). http://www.k12cs.org

Kalloo, R., Kamalodeen, V., Jaggernauth, S. J., Ramsawak-Jodha, N., Sabeerah, V. J. K., Abdul-Majied, S., Dedovets, Z., & Barrow, D. (2019). An exploratory study of game-based approaches in primary mathematics and science classrooms in Trinidad and Tobago. *Journal of Educational Development in the Caribbean*, *18*(1), 21-72

Kamalodeen, V.J., Ramsawak-Jodha, N., Figaro-Henry, S., *Jaggernauth, S.J., & Dedovets, Z. (2021).* Designing gamification for geometry in elementary schools: insights from the designers. *Smart Learn. Environ. 8(36)*, . https://doi.org/10.1186/s40561-021-00181-8

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522-531. doi: 10.1016/j.procs.2012.04.056

Keiny, S., & Orland-Barak, L. (2009). Educational action research as a paradigm for change. In S. E. Noffke, & B. Somekh (Eds.), *The SAGE handbook of educational action research* (pp. 166-177). SAGE Publications Ltd, https://dx.doi.org/10.4135/9780857021021.n16

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.

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154-171.

Looi, C. K., How, M. L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education*, *28*(3), 255-279.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). On computational thinking and STEM education. *Journal for STEM Education Research*, *3*(2), 147-166.

Lu, J. J., & Fletcher, G. H. (2009, March). Thinking about computational thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education* (pp. 260-264).

Margulieux, L. E., Morrison, B. B., & Decker, A. (2020). Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, *7*, 1-16.

Mindetbay, Y., Bokhove, C., & Woollard, J. (2019). What is the relationship between students' computational thinking performance and school achievement?. *International Journal of Computer Science Education in Schools*, *2*(5), 3-19.

Ministry of Education of Trinidad and Tobago MOETT. (2014). *Secondary Curriculum. Information and Communication Technology*. https://www.moe.gov.tt/secondary-3/

Onwuegbuzie, A. J., & Collins, K. M. (2007). A typology of mixed methods sampling designs in social science research. *The Qualitative Report*, *12*(2), 281-316.

Othman, M., & Latih, R. (2006). Women in computer science: No shortage here!. *Communications of the ACM*, *49*(3), 111-114.

Papadakis, S. (2020). Evaluating a game-development approach to teach introductory programming concepts in secondary education. *International Journal of Technology Enhanced Learning*, *12*(2), 127-145.

Papastergiou, M. (2009). Digital game-based learning in high school computer science education. *Computers & Education*, *52*(1), 1–12.

Ramdass, S. (2019). *Using game-based learning to promote computational thinking levels in fourth form girls at a high-achieving secondary school in Trinidad.* [Unpublished master's thesis]. The University of the West Indies, St. Augustine.

Romainor, N., Talib, C. A., & Hakim, N. W. A. (2018). The necessity of computational thinking in STEM education: An analysis with recommended research. *SEAMO Recsam*, 1-12.

Roberts, E. S., Kassianidou, M., & Irani, L. (2002). Encouraging women in computer science. *ACM SIGCSE Bulletin*, *34*(2), 84-88.

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017, March). Assessing computational thinking in CS unplugged activities. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 501-506).

Rountree, N., Rountree, J., Robins, A., & Hannah, R. (2004). Interacting factors that predict success and failure in a CS1 course. *ACM SIGCSE Bulletin*, *36*(4), 101-104.

Sax, L. J., Lehman, K. J., Jacobs, J. A., Kanny, M. A., Lim, G., Monje-Paulson, L., & Zimmerman, H. B. (2017). Anatomy of an enduring gender gap: The evolution of women's participation in computer science. *The Journal of Higher Education*, *88*(2), 258-293.

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. https://eprints.soton.ac.uk/356481/

Sharma, K., Papavlasopoulou, S., & Giannakos, M. (2019). Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children’s attitudes? *International Journal of Child-Computer Interaction*, *21*, 65-76.

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

Swaid, S. I. (2015). Bringing computational thinking to STEM education. *Procedia Manufacturing*, *3*, 3657-3662.

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, *5*(2), 25-44.

Turchi, T., Fogli, D., & Malizia, A. (2019). Fostering computational thinking through collaborative game-based learning. *Multimedia Tools and Applications*, *78*(10), 13649-13673.

US Bureau of Labor Statistics (2021, September 8). *Occupational Outlook Handbook- Computer and Information technology*. https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Valenzuela, J. (2020, September 22). *How to develop computational thinkers*. International Society for Technology in Education. https://www.iste.org/explore/how-develop-computational-thinkers

Vitores, A., & Gil-Juárez, A. (2016). The trouble with ‘women in computing’: a critical examination of the deployment of research on the gender gap in computer science. *Journal of Gender Studies*, *25*(6), 666-680.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, *25*(1), 127-147.

Wight, B. (2019). *Developing computational thinking among form 1 Information and Communication Technology (ICT) students in a rural government secondary school in Trinidad*. [post graduate diploma thesis, The University of the West Indies, St. Augustine]. Academia.edu. https://www.academia.edu/39695220/Developing_Computational_Thinking_Among_Form_1_Information_and_Communication_Technology_ICT_Students_in_a_Rural_Government_Secondary_School_in_Trinidad

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on empty: The failure to teach k--12 computer science in the digital age*. ACM.

Wing, J. M. (2006). Computational thinking. *Communications of the*

ACM, *49*(3), 33-35.

Woollard, J. (2016). CT driving computing curriculum in England. *The Voice of K–12 Computer Science Education and Its Educators*, *12*(1), 4-5.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, *28*(4), 371-400.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education, 14(1), 1–16.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education: A framework for developing teacher knowledge. *Communications of the ACM*, *60*(4), 55–62.

Zuber-Skerritt, O. (2001). Action learning and action research: paradigm, praxis and programs. *Effective Change Management through Action Research and Action Learning: Concepts, Perspectives, Processes and Applications*, *1*, 20.