View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Scholarship at UWindsor

University of Windsor
Scholarship at UWindsor

Major Papers | Theses, Dissertations, and Major Papers

October 2020

A Literature Review for the Implementation of Computational Thinking for Ontario K-12 Classrooms

Stephan Rogers
rogersl@uwindsor.ca

Follow this and additional works at: https://scholar.uwindsor.ca/major-papers

Part of the Curriculum and Instruction Commons, Elementary Education Commons, and the Secondary Education Commons

Recommended Citation
Rogers, Stephan, "A Literature Review for the Implementation of Computational Thinking for Ontario K-12 Classrooms" (2020). *Major Papers*. 150.
https://scholar.uwindsor.ca/major-papers/150

This Major Research Paper is brought to you for free and open access by the Theses, Dissertations, and Major Papers at Scholarship at UWindsor. It has been accepted for inclusion in Major Papers by an authorized administrator of Scholarship at UWindsor. For more information, please contact scholarship@uwindsor.ca.

A Literature Review for the Implementation of Computational Thinking for Ontario K-12 Classrooms

By

Stephan Rogers

A Major Research Paper

Submitted to the Faculty of Graduate Studies

Through the Faculty of Education

In Partial Fulfillment of the Requirements for

The Degree of Master of Education

at the University of Windsor

Windsor, Ontario, Canada

© 2020 Stephan Rogers

A Literature Review for the Implementation of Computational Thinking for Ontario K-12 Classrooms

By

Stephan Rogers

APPROVED BY:

K. Smith

Faculty of Education

G. Salinitri, Advisor

Faculty of Education

August 5, 2020

Declaration of Originality

I hereby certify that I am the sole author of this major research paper and that no part of this major research paper has been published or submitted for publication.

I certify that, to the best of my knowledge my major research paper does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my paper, published or otherwise, are fully acknowledged with the standard referencing practices. Furthermore, to the extent that I have included copyright material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in my major research paper and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of major research paper, including any final revisions, as approved by my major research paper committee and the Graduate Studies office, an that this major research paper has not been submitted for a higher degree to any other University or Institution.

Abstract

The importance of the problem-solving skills involved in computational thinking has gained significant traction since its introduction. As Ontario seeks to implement coding into the school curriculum, an analysis of previous implementation of computational thinking could provide a framework for which to formulate new curriculum in the province. A literature review was completed to investigate the following three questions: (1) How has computational thinking been implemented into education in a K-12 environment? (2) What barriers will affect the implementation of computational thinking in a K-12 environment? (3) What grade levels are appropriate for implementing the varying competencies of computational thinking? This literature review sheds light on the need for teacher support, the political implications involved in introducing new curriculum, and where computational thinking best fits into current K-12 curriculum.

Acknowledgements

I’d like to acknowledge those who continue to encourage and support the pursuit of higher learning. Thank you to Lina, Tania, and Mandy for your help in the office. Thank you to Dr. Xu for your support while working with the RLP. Thank you to Dr. Salinitri for having put up with me for this long and the encouragement along the way. Thank you to my family for your compassion and support through the good and the bad. Thank you to my friends who have helped keep me grounded. Remember to love each other.

Table of Contents

Introduction

The world today continues to make astounding technological advancements. Research in quantum mechanics, artificial intelligence, and space travel continue to develop through modeling techniques and experimentation as a part of computer-aided research (Dunjko, & Briegel, 2018). What appears to be lacking in our pursuit for further technological advancement, is educating our youth to fully adopt technology and the processes of how the technology operates. Many countries such as the United Kingdom, Australia, Israel, and South Korea have adopted new educational policies to implement computer coding as a core portion of Kindergarten to Grade 12 curriculum (Kim, Jeong, Lu, Debnath, & Ming, 2016). Canada has also joined in on the recent globalization convergence by adopting computational thinking and coding as a significant part of the school curriculum. Nova Scotia and British Columbia had announced the implementation of computer coding and computational thinking in September 2016, which has led to a convergence in Ontario schools (Silcoff, 2016; Rushowy & Benzie, 2020). The need for adopting coding has revealed itself, as Ontario schools push for education in science, technology, engineering and mathematics (STEM). The demand for workers in Canadian sectors like manufacturing has decreased, as automation has infiltrated the market (STATSCAN, 2011). As a result, a new pathway has appeared in preparing students towards the upcoming shift in the workforce. Jobs in the STEM fields are expected to grow by approximately 12 percent between 2013 and 2022, and 35 percent of those are expected to be in computer science-related fields (CBC News, 2015). A report by the Information and Communication Technology Council of Canada has suggested that by 2019, over 182,000 information and communication technology (ICT) positions will be left unfilled (Faisal, Asliturk, Bourgi, Savard, Aquilina, & Castillo, 2015). Policymakers have now begun to realize that further integrating skills related to technology would be beneficial for the future workforce.

The Ontario Science and Technology curriculum was last updated in 2007 at the elementary level, and the Computer Studies curriculum was last updated in 2008 at the secondary level. This was so long ago, most social media companies, such as Facebook and Twitter, were still in their early infancy. This is also true with the idea of computational thinking, as it had only just received recognition for its approach to problem-solving (Wing, 2006). Implementing new policy involving computational thinking will face roadblocks. It is important to understand previous experiences of policy implementation involving computational thinking to improve the future rollout of this policy in Ontario schools.

Teacher perception of a policy is important, as the implementation of any new policy can be thwarted by internal politics (Delaney, 2014). The effectiveness of implementation is dependent upon teachers' abilities and their will to implement the policy. As there is extensive evidence of consistency and certainty of policy convergence flowing across the Canadian provinces, and given the current need for skilled individuals in information and communication technology professions, it comes as no surprise that coding policy has been moving through the Canadian provinces (Wallner, 2014). Research in barriers for implementing computational thinking as a policy is limited, but what is known is essential in developing the next steps for large-scale implementation. Developing a pathway for implementation would be inconsequential without the support of the teaching staff (Delaney, 2014). Without a firm grasp of content knowledge for understanding in the computational thinking domain, it could prove difficult to rollout such a policy.

While content knowledge of computational thinking is important, pedagogical approaches and technological knowledge play a role in teachers' ability to deliver instruction to students (Mishra & Koehler, 2006). An educator responsible for nurturing 21st-century skills should be able to demonstrate competency with emerging technology. Developing these skills for in-service and pre-service educators, while also promoting community-driven groups for computational thinking skills could prove to meet the desired outcomes. Pre-service educators in Ontario are required to complete an integration and computing technology course, which does not address pedagogical, content, or technological knowledge related to computational thinking and coding (Ontario College of Teachers, 2020). As most pre-service students will be entering education from backgrounds outside of computer science, introducing a strategy system for problem-solving should require additional support for this group if they are expected to demonstrate competency.

Another stakeholder affected by this large-scale change would be Ontario students. Learners with disabilities and students identified as gifted or at-risk will have different needs for student success. Students who have grown-up with less access to technology could have an impact on their ability to use and manipulate tools used in developing computational thinking knowledge. As roughly two million students will be affected between primary and secondary schools, accommodations would need to be made (Government of Ontario, 2020).

As a secondary school teacher, getting through the entire curriculum for a given course in the required timeframe is a massive undertaking. Would coding be taught as a stand-alone science, or incorporated into current curriculum expectations? Understanding the outcomes of practical and innovative approaches to implementation, as it pertains to computational thinking, should serve as the foundation if Ontario were to improve upon existing computational thinking practices. This should also serve to further student outcomes.

Instilling computational thinking should be a priority for any government looking to implement a new policy. Having the foresight to understand the complications involved in this massive undertaking should recognize unintended side-effects. When the United States had implemented the No Child Left Behind Act, it would have been difficult to believe that there was no evidence of improved student achievement in reading (Dee, Jacob, Hoxby & Ladd, 2010). Duncan, Bell, and Tanimoto (2014) suggested that there could be a significant cost in equipping teachers to deliver programs surrounding coding and that substantial time teaching other subjects could be lost. Teaching coding exclusively, rather than the problem-solving skills applied through computational thinking, may negatively affect student's perception of what is computational thinking. Would students' perception of computational thinking change if they felt they were developing work skills through the required curriculum, or would it foster students who are already interested in computing? A negative experience from a student could turn him/her off from coding and computational thinking for the rest of their education.

While the push for computational thinking and coding exists to meet the demand for workers with computer science skills, the problem-solving competencies have shown other benefits. In a study by Calao, Moreno-León, Correa, and Robles (2015), computational thinking was integrated into some sixth-grade mathematics' classes demonstrating significant improvement in students' understanding of mathematics processes as compared to a control group that did not have computational thinking in its math class. The study reported a significant increase in problem-solving and critical thinking skills. Other studies (Van Dyne & Braun,

2014) have reported similar findings, which should provide further encouragement for integrating computational thinking into the curriculum.

Statement of Problem

As the province of Ontario has moved to implement coding into the school curriculum, they must include the tool of computational problem-solving strategies to develop student coding capabilities appropriately. As coding is the process for using a computing language, computational thinking is the process of problem-solving for the new language. Implementation should cultivate an environment that positively engages students in computing to later meet the economic needs for filling information technology positions. The Next Generation Science Standards have also identified computational thinking as key scientific and engineering practices that must be understood and applied in learning about the sciences (National Research Council, 2012). Understanding how to successfully implement computational thinking will not only help those develop their computing skills but establish new tools that further develop higher-order thinking.

Kong (2016) states that the "young generation today is expected to maintain a competitive power and be willing to contribute to social enhancement by problem-solving creatively with digital technologies" (p. 371). A curriculum poorly implemented, or without the appropriate tools to understand coding or the thought processes involved in coding, may struggle to attract young learners to develop an interest in computing.

In this literature review, I will investigate previous implementations of computational thinking into K-12 schools. These recommendations will be from studies completed at a local, state or provincial, and national implementation to better understand the issues that Ontario will face as Ontario moves to implement computational thinking problem-solving skills into its provincial curriculum. Having a better understanding of the issues involved in the implementation of computational thinking should serve as an indicator of best practices in overcoming social, economic, or political issues when modifying the current curriculum. Understanding implementation best practices should have a positive effect on the stakeholders involved in the rollout of a new curriculum utilizing computer coding.

Language and Terminology

The following are a list of terms used through this literature review:

Computational thinking (CT)

Computational thinking was initially defined as "taking an approach to solving problems, designing systems and understanding human behaviour draws on concepts fundamental to computing (Wing, 2006)." A definition for CT today is not universally agreed upon in the context of K-12 education, but the widely recognized competencies include problem decomposition, algorithmic thinking, abstraction, data collection, automation, parallelization, and simulation (Barr & Stephenson, 2011; Yadav, Mayfield, Zhou, Hambrusch & Korb, 2014; Mouza, Yang, Pan, & Ozden, 2017). CT today is a problem-solving methodology that uses competencies to be applied across different subjects (Barr & Stephenson, 2011). These competencies can be further defined as:

a. Decomposition – breaking down large significant problems into manageable parts
b. Algorithmic thinking – using a precise sequence of instructions to solve a problem
c. Abstraction – removing unnecessary parts to better understand a problem without losing any important information
d. Data collection – accessing, evaluating, and representing data using words, images, or models
e. Automation – using tools to automate solutions
f. Parallelization – organizing resources to simultaneously carry out tasks
g. Simulation – creating models to represent a process

It is important to recognize that computational thinking is not merely simple coding or using computers, but a separate domain within computer science that can be misunderstood (Mouza et al., 2017).

Technological pedagogical content knowledge (TPCK)

Mishra & Koehler (2006) developed a theoretical framework for educational technology that interprets the domains for the use of technology in an educational setting. The three domains are content knowledge (CK), pedagogical knowledge (PK), and technological knowledge (TK). All three domains are not mutually exclusive, which gives rise to pedagogical content knowledge (PCK), technological content knowledge (TCK) and technical, pedagogical knowledge (TPK). Where CK, PK, and TK intersect is referred to as technological pedagogical content knowledge (TPCK). TPCK is the basis of effective teaching with technology. This requires an understanding of the pedagogical techniques that use technology in constructive ways, the knowledge of what makes concepts difficult or easy to learn and how technology can address these problems that students face, and how technologies can be used to build on existing knowledge.

Literature Review

Creswell and Creswell (2017) outline that a literature review is used to share with the reader the results of other studies that are closely related to the one being undertaken. Results or emergent themes are presented after patterns or categories have been identified (Creswell & Creswell, 2017).

Methodology

Literature Review

A literature review was conducted to understand the current scope of information on the implementation of CT and its suggested frameworks. The literature map was developed chronologically to better understand the progress of CT implementation, as Wing's introduction of the CT was introduced nearly a quarter-century ago. As outlined by Creswell and Creswell (2017), a literature review may include:

1. Identifying your area of research
2. Identifying relevant keywords
3. Searching for literature with the keywords in mind
4. Code the literature through drafted summaries with the most relevant articles
5. Structure the information thematically by organizing important concepts
6. Reporting the results

These steps were used to develop the literature review, based on the research questions below.

Research Question

For this paper, the following questions guide this literature review:

1. How has CT been implemented into education in a K-12 environment?
2. What barriers will affect the implementation of CT in a K-12 environment?
3. What grade levels are appropriate for implementing the varying competencies of CT?

Relevant Studies

This literature review adapts the article selection process for relevant studies, as outlined by Creswell & Creswell (2017). The following steps were conducted to encapsulate relevant studies:

1. Beginning with a broad review of literature, such as overviews and summaries of the topic presented in journal articles or abstract series.
2. Utilizing journal articles from respected journals that report research studies.
3. Utilize books related to the topic that may utilize a group of authors or books that contain chapters written by different authors.
4. Follow recent conference papers from major notational conferences and the articles delivered at them.

For this paper, I conducted a literature review from the following peer-reviewed journals focusing on computational thinking and the implementation in K-12 schools:

1. Computer science education journals
2. Educational technology journals
3. Science education journals
4. Psychology journals

The research questions listed intended to focus on school systems that replicated Western educational policies and practices. As practical knowledge for the implementation of computational thinking in curricula is limited in North America, outside sources were investigated.

Study Selection

The peer-reviewed journals focusing on the implementation of computational thinking in classrooms were collected through online search engines. The relevant articles were selected based on their application to the research question, within the context of education, and written within the last ten years (2010-2020). The following databases were used:

1. University of Windsor, Leddy Library
2. ProQuest
3. ERIC – Educational Resources Information Center
4. ACM Digital Library
5. PsycINFO
6. Statistics Canada
7. JSTOR
8. Google Scholar

The following journals were used to collect literature:

1. Computers in Schools
2. International Journal of Computer Science Education in Schools
3. Computers & Education
4. Journal of Educational Technology & Society
5. Computers in Education

Sources that did not appear in peer-reviewed journals, major conference papers and dissertations, published books, and government publications were not included. While researching the topic, articles selected from peer-reviewed journals focused on the following list of core topics:

1. Appropriate title and abstract relevance
2. CT implementation experiences
3. Teacher perception of CT
4. Framework for implementation
5. Barriers for implementation of CT
6. Relevant references

Keywords

Keywords were used in varying combinations to identify relevant articles to the research topic. The keywords used to gather literature for this review were:

1. computational thinking (CT)
2. implementation
3. education
4. curriculum design
5. policy
6. teachers
7. quantitative/qualitative study
8. framework
9. computing
10. K-12

Results

Table 1: A brief overview of the articles used in this research.

Title	Author(s)	Date Published	Quantitative /Qualitative	Pop.	Pop. Size
1. Bringing CT thinking to K-12: what is involved, and what is the role of the computer science education community?	Barr, V., & Stephenson, S.	2011	Qualitative		United States
2. Computational thinking in elementary and secondary teacher education	Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T.,	2014	Quantitative	US Midwestern university	357 pre-service teachers from the educational psychology course
3. Supporting all learners in school-wide computational	Israel, M., Pearson, J. N., Tapia, T., Werfel, Q.	2015	Qualitative	US Midwestern elementary	Seven teachers

thinking: A cross-case qualitative analysis.	M., & Reese, G.			school teachers	
4. Computational Thinking Curriculum for K-12 Education – A Delphi Survey	Chuang, H. C., Hu, C. F., Wu, C. C., & Lin, Y. T.	2015	Quantitative	Taiwanese stakeholders	12 computer scientists, K-12 computer teachers, CS educators, industry experts
5. A K-6 CT Curriculum Framework: Implications for teacher knowledge	Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J.	2016	Qualitative		
6. A framework of curriculum design	Kong, S. C.	2016	Qualitative		

for computational thinking development in K-12 education					
7. Computational thinking in teacher education	Yadav, A., Stephenson, C. & Hong, H.	2017	Qualitative	Historical Analysis	United States
8. A computational thinking approach to the development of technological pedagogical content knowledge	Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L.	2017	Quantitative /Qualitative		21 pre-service teachers in their early 20s
9. Educational policy and implementation of computational thinking and programming: a	Seow, P., Looi, C. K., Howe, M. L., Wadhwa, B., & Wu, L. K.	2019	Qualitative	Historical Analysis	Singapore

case study of Singapore					

Summary of findings from each of the articles considered:

1. Bringing CT thinking to K-12: what is involved, and what is the role of the computer science education community? (Barr & Stephenson, 2011)
    a. K-12 education today is a highly complex, highly politicized environment where multiple competing priorities, ideologies, pedagogies and ontologies all vie for codominance (p.114).
    b. Acknowledges that many disciplines require, promote, and teach problem-solving skills, logical thinking, or algorithmic thinking and that implementing CT into schools can be accomplished through systemic change (p.112).
    c. Two major strategies for achieving systemic change are to gain resources to inform policymakers about the importance of CT. The other is to educate teachers to appropriately and effectively integrate new concepts into their content and pedagogical knowledge while transforming that knowledge into content practice (p.119).
2. Computational thinking in elementary and secondary teacher education (Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T., 2014)
    a. Allowing students to maximize the benefits of CT would require integration into core content areas at the K-12 level. This would need teachers to have adequate knowledge of CT and how to incorporate it into their disciplines (p. 4)

b. CT modules were created to demonstrate to pre-service teachers the general concepts of probabilistic reasoning, algorithmic thinking, heuristics, hypothesis testing, and problem-solving. When tested against a control group who did not receive instruction on CT, treatment group participants were able to form an understanding that CT was a cognitive tool to solve complex problems. In contrast, the control group tended to label CT as "the use of computers" (p. 7).

c. In the computing attitude survey, participants in the treatment group were more likely to agree that CT involved logically solving problems and abstracting general principles. This treatment group was also more likely to recognize that CT could be implemented in the classroom through problem-solving (p. 10).

d. Those in the control group were more likely to report that CT was simply the use of computers and how they worked (p. 10).

e. While females are currently underrepresented in the computing education pipeline, survey results showed that females and males are equally comfortable with computing, and both genders see computing having a role in their careers. Understanding the importance of computing could be significant in increasing the number of females pursuing computer science (p.14).

f. One way for systemic change is to incorporate CT modules into core teacher education courses to expose future teachers, as the modules influenced teacher perception through a greater understanding of CT practices (p. 14).

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

a. Computing in K-12 instruction is important as there is a growing demand for workers with computer science skills. Computing skills have also been found to improve learners' higher-order thinking skills and the development of algorithmic problem-solving skills (p. 264)
b. Teacher and student perceptions greatly influence their attitudes about CS learning and careers. How a teacher demonstrates computer science concepts to students can leave experiences thinking that computer science is boring, confusing, and too difficult to master. This can create misconceptions about future computing career opportunities (p. 264).
c. Research on teaching practices indicated that teachers who were initially skeptical of implementing computing found computer programs such as Scratch and E-toys to be both valuable and accessible (p. 264).
d. The authors identify that no literature exists related to how teachers implement computing within the context of school-wide computer science initiatives at the elementary level, especially with diverse learners. This gap is addressed in the research that includes students from diverse backgrounds and those at-risk for academic failure due to poverty or disability (p. 265).
e. In the case-study analysis, it was found that integrating computing into content areas was key to successful implementation. The teachers in the study agreed that the rapid pacing of the curriculum was too rapid to add computing as a distinct area of instruction (p. 268).

f. Teachers were initially apprehensive about integrating computing (K-5) but were eager to integrate it into their instruction when given access to support and expertise (p. 268).
g. Undergoing professional development, with embedded coaching, and computing expertise was key to successful implementation (p. 270)
h. Teachers had mentioned the vital importance of administrative support for computing to be implemented and sustained (p. 271).
i. The three major barriers to the implementation found in the study were a lack of technology, a lack of computing expertise, and students' status as at risk for academic failure due to poverty and disability (p. 271).
j. Access to technology was difficult as students rotated to different classrooms to access the computer lab. Teachers utilized 'Donors Choose' to gain access to more technology (p. 271).
k. A university faculty and graduate student were used for support and coaching while utilizing online resources were used to overcome a lack of expertise (p. 271).
l. To overcome struggling learners, peer support and collaboration, as well as one-on-one supports, were implemented. A balance of explicitly and open instruction was used, which allowed students to explore without "correct" answers (p. 271).
m. Students with poverty and disability risk factors encouraged participation proactively rather than accept any limitations (p. 272).

n. Students with disabilities who struggled with reading had difficulty reading within Scratch and E-toys as well as with complex problem solving involved in some of the computing activities (p. 272).

o. It was found that students who lacked access to technology due to poverty struggled more than students with mild to moderate disabilities (e.g. Learning or emotional behaviour disorders). Students who did not have access to technology did not have the opportunities to learn fundaments skills such as using a mouse/trackpad, dragging, double-clicking, etc. (p. 273)

p. The teacher prompted collaboration and peer mentoring as collaboration both proved to be successful models in instruction delivery (p. 274).

4. Computational Thinking Curriculum for K-12 Education – A Delphi Survey (Chuang, H. C., Hu, C. F., Wu, C. C., & Lin, Y. T., 2015)

   a. To determine the core ability and training of CT at different grade levels, a Delphi research methodology was used to collect different views. The consensus was driven by thirteen experts that included computer scientists, computer science educators, K-12 computer teachers, and industry experts (p. 213).

   b. From K-2, students are capable of problem-solving and problem decomposition (p. 213).

   c. From 3-5, students are capable of algorithms, data analysis, modeling and simulation, and automation (p. 214).

   d. Data representation and abstraction are trained from grades 7-9 (p.214).

   e. Data representation, modeling and simulation, as well as algorithms are the most important computational themes for grades 10-12 (p. 214).

f. Most of the themes of CT are suitable in every grade except problem decomposition, data representation and abstraction, as these are modeled in the higher grades (p. 214).

5. A K-6 CT Curriculum Framework: Implications for teacher knowledge (Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J., 2016)

   a. Everyone needs knowledge that goes beyond 21st-century skills that have long-term value enabling them to understand the basics of computer structures and practices. Citizens must understand what computers can and cannot do, so they become effective authors/creators of computational tools (p. 47).

   b. There are concerns to teaching computer science in K-6 that are linked to the incompatibility between abstraction and children's weakness to understand it at a young age as they cannot understand concrete logic (p. 48).

   c. Early exposure during kindergarten is necessary as research has found that young children can think abstractly when concrete reference systems are used to situate their thinking (p. 48).

   d. The framework of CT curriculum should fit within the definition of CT as outlined by the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE). This definition includes the elements of abstraction, generalization, decomposition, algorithmic thinking, and debugging (p. 49).

   e. Boundaries for the elements of CT for each level may vary from school to school and from classroom to classroom, as students have varying needs. Refinements should also be made as data becomes available from pilot offerings (p. 50).

f. A holistic approach to teaching CT aims at eliminating compartmentalization and fragmentation by focusing on complex authentic learning tasks, without losing sign of the individual elements that make up the complex whole (p. 52).
g. The design of problem-solving tasks should focus on real-life issues, and the problem-solving tasks should be sequenced from simple to complex (p. 52).
h. For teachers to be effective at implementing CT, they should have technological pedagogical content knowledge (TPCK). For CT, TPCK is defined as knowing how to identify a range of creative and authentic CT projects, using technologies that are appropriate for practicing the range of CT skills and having the content and pedagogical knowledge to create an understandable experience for all learners.

6. A framework of curriculum design for computational thinking development in K-12 education (Kong, S. C., 2016)
    a. Calls for the integration of CT into K-12 education gives rise to the need for theory-based, tested, and successful approaches to curriculum design (p. 378).
    b. Kong suggests that an interest-driven creator (IDC) model be used for implementation to foster students' learning interests, capabilities in creation, and learning habits (p. 378).
    c. Every individual is expected to be digitally comfortable and competent to maintain their competitive power and also to be willing to contribute to social enhancement by solving problems creatively with digital technologies (p. 381).

d. The acquirement of digital empowerment through education is an inevitable means to equip learners to become influential members in the digital community (p. 381).

e. An interest-driven learning activity should be embedded in the curriculum for nurturing creativity (p. 382).

f. A CT curriculum should be increasingly complex, which means that learning activities should not only be interrelated but also built upon one another. This indicates that a top-down curriculum strategy should be implemented, by looking at the most challenging problem at the highest level and have the lower-tasks lead into the difficult problem (p. 384).

7. Computational thinking in teacher education (Yadav, A., Stephenson, C. & Hong, H., 2017)

   a. CT is a set of problem-solving thought processes derived from computer science but applicable in any domain, including biology, journalism, finance, and archaeology (p. 56).

   b. Pre-service teacher education is an opportune time to provide teachers with the knowledge and understanding they need to successfully integrate CT into their practice (p. 56).

   c. References a study by Calao, Moreno-León, Correa, & Robles (2015), where integrating CT in sixth-grade mathematics class significantly improved students' understanding of mathematics processes when compared to a control group that did not learn CT in their math class (p. 58).

d. Recent efforts to train teachers to embed CT have focused on in-service teacher professional development, but there is limited understanding of how to engage pre-service teachers from other content areas in computer science and CT (p. 59).
e. There is a need to develop pre-service teachers' knowledge and skills on how to think computationally and then how to teach their students to think computationally. They should then understand CT in the context of the subject area they will be teaching. This will require them to have a strong understanding of their discipline and how CT concepts relate to it (p. 59).
f. The authors suggest that CT should be introduced in pre-service teacher educational-technology courses, as they are typically disconnected from teaching theories and focused the technology itself (p. 60).
g. Educational technology courses based on their subject areas could allow teachers to develop CT knowledge within the context of their content knowledge and pedagogical knowledge (p. 60)

8. A computational thinking approach to the development of technological pedagogical content knowledge (Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L., 2017)
    a. TPACK has provided a unifying lens for researchers to understand teacher knowledge for effective use of technology tools, methodologies and practices across the curriculum, as it is a useful framework for studying knowledge in relation to CT (p. 61).
    b. Embedding CT knowledge and skills across the curriculum is essential for helping students understand how to use computing tools to represent knowledge and solve problems (p. 63).

c. A key obstacle to embedding CT in K-8 standards and curricula is teacher preparation (p. 64).
d. The authors reference a study by Bowers and Falkner (2015) where participants were unaware of the term CT and mistakenly considered CT as the basic use of technology as the participants were unaware of what they did not know (p. 65).
e. Post-survey data from the 15-week course had demonstrated an improved understanding of CT concepts through more detailed and conveying responses (p. 69).
f. Participants' post surveys were more specific and detailed, endorsing the use of coding, as well as problem-solving based assignments (p. 69).
g. Average mean scores on the technology integration assessment with CT related criteria involving curriculum, instructional strategies, technology selection, and fit (content, pedagogy and technology together) all showed positive outcomes (p. 70).
h. The CT concepts related to simulation and parallelization were absent from pre-service teachers' case reports. These concepts rely on the use of programming tools, such as scratch (p. 74).
i. CT skills must be integrated across teacher education curricula to foster a deeper understanding of CT concepts (p. 74).

9. Educational policy and implementation of computational thinking and programming: case study of Singapore (Seow, P., Looi, C. K., How, M. L., Wadhwa, B., & Wu, L. K., 2019)

a. To address the shift from a knowledge/information economy to an economy driven by computation, the national government introduced educational policies that would prepare their citizens for the future (p. 346).

b. Singapore launched the ‘Smart Nation’ initiative, which was a nationwide effort to harness technology in the sectors of business, government and home to improve urban living, build stronger communities, grow the economy and create opportunities for all residents to address the ever-changing global challenges (p. 347).

c. Kindergarten and preschools were introduced to CT through the use of robotically programmable toys that would engage young children in play while developing CT skills such as algorithmic thinking (p. 347).

d. Preschool teachers do not use much technology as the emphasis is on literacy development and play. To address concerns and apprehensions, seminars and hands-on workshops were provided to improve teachers’ technological knowledge (p. 348).

e. Primary schools implemented “Code for Fun” programming activities, which included funding for a visual programming language (Scratch) combined with robotic kits, aiming to make students appreciate programming and develop CT skills in problem-solving and logical thinking. Clubs and competitions were also expanded to encourage participation (p. 350).

f. In secondary schools, the Ministry of Education introduced an open level ‘Computing’ subject, replacing ‘Computer Studies.’ The new curriculum implements coding, and developing CT skills to solve problems, moving away from the previous course that revolved around using software applications (p. 350).

g. Unlike Finland, England and Korea, Singapore did not include computing or CT as compulsory education. This nurturing approach is intended for students to take an interest at an early age (p. 352).

h. Initiatives were launched to offer free programming lessons to underprivileged children to assist in enthusing a broader base of students in computing and expose them to possibilities of technology through enrichment programs and co-curricular activities.

Research Analysis

As nine articles were chosen for this literature review, some common themes were apparent. The following is a discussion of these themes, and the common occurrences found that are relevant to the research topic of this major paper.

Theme 1: Teachers will need support with CT

In studies completed by Yadav et al. (2014) and Mouza et al. (2017), many pre-service teachers still do not understand the concept of computational thinking. In control groups for both studies, pre-service teachers had misconceptions about what computational thinking was and related the concept to computer literacy. Test groups in these studies and the study by Angeli et al. (2016) demonstrated a significantly improved understanding of CT and programming concepts after having received theory related training. With an improved understanding of CT, most pre-service teachers were able to successfully integrate curriculum expectations with CT competencies (Angeli et al., 2016).

Varying resources can provide supplemental training, such as through classes at local colleges and universities, as well as through peer-training groups (Barr & Stephenson, 2011; Israel et al., 2014; Seow et al., 2019). Providing these workshops through joint partners can be an effective strategy for maximizing resources for school boards (Barr and Stephenson, 2011). During the implementation of CT into the classroom transition, small groups who shared resources and held regular meetups proved to be great supports for each other as teachers may lack in technological knowledge skills (Mouza et al., 2017; Seow et al., 2019).

Theme 2: Computational thinking can be implemented across all subjects

Wing (2006) had initially suggested that CT be a multi-disciplinary approach to problem-solving, as it was a necessary 21st-century skill. The authors of this literature agreed with Wing's designation. Mouza et al. (2017) suggested that CT knowledge and skills were essential for helping students understand how to use computing tools to create and discover new questions within specific disciplines. Yadav et al. (2017) state that although the analytical thinking skills draw on concepts from computer science, it has practice to all central sciences, as well as influence in fields such as biology, journalism and finance. Angeli et al. (2016) suggest that CT implementation focuses on problem-solving tasks with a focus on real-life issues, rather than compartmentalizing the CT competencies in computer science. As time constraints may cause issues fitting in an entirely new subject, taking on a holistic approach without compartmentalization could overcome the time constraints of CT as a stand-alone discipline (Israel et al., 2015; Angeli et al., 2016).

Illustrations of different applications for varying courses were also provided as cross-curricular examples were produced. Students in computer science high school courses could be designing phone applications presented as a final project that is interest-driven, providing a meaningful experience for students (Kong, 2016). In the study by Israel et al. (2015), some students integrated science, language arts, and mathematics while using computing software to investigate the life cycle of a tree. In science courses, collecting and analyzing data from experiments while summarizing the data are parts of the scientific method process (Yadav et al., 2017). The CT competencies promote generalizable thinking skills which students will use as they pursue higher levels of achievement (Mouza et al., 2017).

Theme 3: Implementation of CT has political implications

As a curriculum change is required for the implementation of a new strategy for problem-solving, any change will be scrutinized by all stakeholders involved in education. Barr and Stephenson (2011) highlight that, “K-12 education today is a complex, highly-politicized environment where competing priorities ideologies, pedagogies, and ontologies vie for dominance” (p. 114). Highlighting what curriculum must go or what outcomes or standards are replaced would be a difficult decision for policymakers (Chuang et al., 2015). Additional resources for connecting learning goals to CT and developing teachers to have the appropriate TPCK will have a financial cost that will require justification (Barr and Stephenson, 2011; Angeli et al., 2016; Mouza et al., 2017).

Countries like South Korea and Singapore have added initiatives to embrace programming and CT strategies in nationwide shifts to improve their business sectors and grow their economies (Seow et al., 2019). Implementing this at an educational level allows students to grow their competitive power by solving problems creatively within the digital community (Kong, 2016). Students who can utilize the strategies of CT independently should have a competitive edge if CT competencies are correctly implemented by pedagogically sound practices (Yadav et al., 2017).

All nine articles were connected in these varying themes, and there were distinctive gaps in research with respect to an accepted definition of CT for K-12 education. While most of the journal articles posited their own definition of CT for a K-12 environment, many had accepted the previously defined competencies of CT.

Limitations

The focus for this literature review was from articles and studies taken from peer-reviewed journals, or conference presentations. While these are excellent resources for a literature review, ignoring non-peer-reviewed articles, or recent dissertations in an emerging field may have limited the research. Six of the selected articles were qualitative artifacts, as the amount of quantitative data related to CT implementation was limited. The journal articles also highlight the limited body of knowledge regarding post-implementation experiences from teachers and students' learning outcomes. While I am a secondary school teacher, my expertise is not in computer science, which may have limited my understanding of the subject due to my personal biases with respect to computational thinking.

Discussion

After a literature review of the peer-reviewed articles, common themes and analyses were evident. The following discussion is based on the initial guiding questions:

1. How has CT been effectively implemented into education in a K-12 environment?
2. What barriers will affect the implementation of CT in a K-12 environment?
3. What grade levels are appropriate for implementing the varying competencies of CT?

Authors of this literature review suggest a framework for effective implementation of CT in the K-12 curriculum, but very little quantitative data exists of what made the implementation effective. As a result, a discussion of effective implementation must focus on areas of success in the research and case studies available.

In the study by Israel et al. (2015), whole-class and peer-mentor instructional methods were found to be effective ways of presenting CT materials. In whole-class instruction, frontloading new information assisted in reducing student frustration by outlining the task. Teachers in this study had also agreed that explicit tasks were required to develop independence for the more complex tasks that develop higher-order CT competencies (p. 268). Kong (2016) supports this method, suggesting a top-down approach to activity design, which should prepare learners with appropriate knowledge for complex tasks. With the use of peer-mentor instructional methods, student collaboration elicited problem-solving, minimizing the support of the teacher as an expert (Israel et al., 2015). In the study by Israel et al. (2015), it was found that there should be a balance between explicit instruction, individualization and scaffolding inquiry to better support all learners.

Independently developing CT problem-solving skills is also an important component for the students' personal development. Kong (2016) suggests that a curriculum designed through interest-driven learning activities is appropriate for nurturing student creativity. It is valuable to take this heuristic approach at an early age to foster student interest in enrichment programs, special interest clubs, and after school activities (Seow et al., 2019). An interest-driven approach benefited pre-service educators, as they were able to demonstrate TPCK related to CT when creating lessons within their respective majors (Mouza et al., 2017). Yadav et al. (2017) suggest that pre-service teachers enter educational technology courses based on their subject area to further develop their understanding of CT in a collaborative setting while deepening their content and pedagogical knowledge.

Many challenges are facing Ontario education that will complicate the implementation of CT into K-12 education. While the need for teacher training and the need for an interest-driven approach are mentioned throughout the literature, very little is mentioned about barriers faced when implementing CT content into subject-specific areas. The three main barriers to implementation mentioned by Israel et al. (2015) were a lack of technology, lack of computing expertise and students' status as at risk for academic failure due to poverty and disability. A lack of technology directly refers to the technological infrastructure in a school building. While the use of a computer is not required to demonstrate CT skills, if students are to develop all CT competencies, access to technology will be required. Teachers and students unfamiliar with programming will typically begin their experiences in block-based visual programming of robots through the Scratch programming language. They will need access to the technology that can access the software. Mouza et al. (2017) mention the importance of teacher preparation when implementing CT, as teachers will require the appropriate technological expertise to utilize these types of robots and have the expertise to pass on this knowledge to students.

Implementation of CT into the curriculum will require funding to train teachers but will also require investment in new resources. These resources could include curricular materials, models and simulations, model activities, and websites for independent study activities (Barr and Stephenson, 2011). As an initial rollout, an opt-in model was used in Singapore, with interested schools applying for resources from a list of approved vendors (Seow et al., 2019). A regional or pilot project approach could be undertaken, as previous Ontario governments had participated in these studies with financial literacy (Cross, 2017). As teachers develop their CT skills and resources through professional development, the availability of subject-specific artifacts should increase.

Two other major challenges relate to what content to teach across different educational levels, and what body of knowledge do teachers need to teach competently (Angeli et al., 2016). While Wing's (2006) initial introduction of CT was meant to be a problem-solving method across all disciplines, the initial body of research in this literature review involved implementing CT into computing and computer science programs exclusively. Chronologically speaking, more recent literature frames CT strategy applying to all disciplines, while providing little evidence of how to do so. Until a definition of CT for K-12 students can be accepted and defined, it will be difficult to evaluate students and educators of their CT competencies (Barr & Stephenson, 2012; Yadav et al., 2014; Kong, 2016; Angeli et al., 2016; Mouza et al., 2017).

Deciding what grade levels are appropriate for varying competencies of CT is also challenging. Chuang et al. (2015) and Angeli et al. (2016) agree that younger children have difficulty understanding abstraction, as they struggle with concrete logic. While most competencies are suitable for students at all grade levels, problem decomposition, data representation and abstraction are more useful in different subjects in grades 10 to 12 (Chuang et al., 2015). As the rest of the competencies are appropriate for all grade levels, understanding what curriculum expectations for CT competencies across all K-12 education remains an enormous up-taking. Applying Kong's (2016) top-down method of developing curriculum across the board would require the expertise of all subject-specific educators. As teachers generally accept that there is already limited instructional time with curriculum expectations, integrating CT into outcomes will need to be vigorously explored (Israel et al., 2015).

Conclusion

Coding is coming to Ontario schools, and students should be equipped with the tools to find success in computing. Teachers should be provided with the appropriate professional development to be successful in implementing CT and utilizing established CT pedagogical practices. Teachers need to be equipped with knowledge on how to teach and assess the skills of CT, as well as creating and adopting real-world applications for instructional use. Rather than having CT and coding be a stand-alone subject, incorporating these skills into the existing curriculum will limit teachers' need to cram for more material. A curriculum design with this in mind is logical, as the original goal of CT was meant to be a skillset used throughout the sciences. The research in this literature review supports this pathway while also addresses the need to focus on pre-service teacher education by adapting the required computing technology course.

K-12 education can support computational thinking. It will train students with skills to prepare them for the future labour market, promoting economic growth in Ontario. While the parts of computational thinking added to the curriculum remain a political issue, all stakeholders should be involved for successful implementation.

REFERENCES

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.

Barr, V., & Stephenson, C. (2011). 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.

Bower, M., & Falkner, K. (2015, January). Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities. In *ACE* (pp. 37-46).

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In *design for teaching and learning in a networked world* (pp. 17-27). Springer, Cham.

CBC News. (2015). Back to school: Canada lagging in push to teach kids computer coding - Technology & Science - CBC News. Retrieved February 21, 2018, from http://www.cbc.ca/news/technology/back-to-school-canada-lagging-in-push-to-teach-kids-computer-coding-1.3185926

Chuang, H. C., Hu, C. F., Wu, C. C., & Lin, Y. T. (2015, April). Computational Thinking Curriculum for K-12 Education--A Delphi Survey. In *2015 International Conference on Learning and Teaching in Computing and Engineering* (pp. 213-214). IEEE.

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

Cross, J. (2017, March 23). Ontario launches pilot project to test financial literacy, entrepreneurship course. Retrieved July 28, 2020, from https://globalnews.ca/news/3329741/ontario-launches-pilot-project-to-test-financial-literacy-entrepreneurship-course/

Dee, T. S., Jacob, B. A., Hoxby, C. M., & Ladd, H. F. (2010). The impact of No Child Left Behind on students, teachers, and schools [with comments and discussion]. *Brookings papers on economic activity*, 149-207.

Delaney, J. G. (2002). Educational policy studies: A practical approach. *Brush Education*.

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

Dunjko, V., & Briegel, H. J. (2018). Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, *81*(7), 074001.

Faisal, S., Asliturk, E., Bourgi, S., Savard, A., Aquilina, A., & Castillo, D. D. (2015). The smart economy reshaping Canada's workforce: Labour market outlook 2015-2019. *The Information and Communications Technology Council*.

Government of Ontario. (2020). Education Facts, 2018-2019. Retrieved July 29, 2020, from http://www.edu.gov.on.ca/eng/educationfacts.html

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

Kim, D. K., Jeong, D., Lu, L., Debnath, D., & Ming, H. (2016). Opinions on computing education in Korean K-12 system: higher education perspective. *Computer Science Education*, 1-19.

Kong, S. C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, *3*(4), 377-394.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, *108*(6), 1017-1054.

Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & 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).

National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

Ontario College of Teachers. (2020). Requirements for Becoming a Teacher of General Education in Ontario. Retrieved July 12, 2020, from https://www.oct.ca/-/media/PDF/Requirements%20General%20Education%20Teacher/EN/general_education_teacher_e.pdf

Rushowy, K., & Benzie, R. (2020, June 23). New Ontario elementary math curriculum to include coding starting in Grade 1. Retrieved June 24, 2020, from https://www.thestar.com/politics/provincial/2020/06/23/new-ontario-elementary-math-curriculum-to-include-coding-starting-in-grade-1.html

Seow, P., Looi, C. K., How, M. L., Wadhwa, B., & Wu, L. K. (2019). Educational policy and implementation of computational thinking and programming: Case study of Singapore. In *Computational thinking education* (pp. 345-361). Springer, Singapore.

Silcoff, S. (2016, January 17). BC to add computer coding to school curriculum. *The Globe and Mail*. Retrieved January 18, 2018, from http://www.theglobeandmail.com/technology/bc-government-adds-computer-coding-to-school-curriculum/article28234097/

STATSCAN. (2014, September 2). Manufacturing. The Year in Review. Retrieved February 21, 2016, from http://www.statcan.gc.ca/pub/11-621-m/2015097/part-partie1-eng.htm

Van Dyne, M., & Braun, J. (2014, March). Effectiveness of a computational thinking (cs0) course on student analytical skills. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 133-138).

Wallner, J. (2014). Learning to school: Federalism and public schooling in Canada. *Toronto, ON: University of Toronto Press*.

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

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 (TOCE)*, *14*(1), 1-16.

Yadav, A., Stephenson, C. & Hong, H. (2017). Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer, Cham.

VITA AUCTORIS

NAME:	Stephan Rogers
PLACE OF BIRTH	Windsor, Ontario
YEAR OF BIRTH	1986
EDUCATION	Purdue University, Fort Wayne, Indiana
	2004-2008 B. Sc.
	University of Windsor, Windsor, Ontario
	2009-2010 B. Ed.
	University of Windsor, Windsor, Ontario
	2016-2020 M. Ed.