Hiding in Plain Sight: Identifying Computational Thinking in the Ontario Elementary School Curriculum

Eden J.V. Hennessey[1], Julie Mueller[2,*], Danielle Beckett[2] & Peter A. Fisher[3]

[1]Department of Psychology, Wilfrid Laurier University, Waterloo, Canada

[2]Faculty of Education, Wilfrid Laurier University, Waterloo, Canada

[3]Lazaridis School of Business and Economics, Wilfrid Laurier University, Waterloo, Canada

*Correspondence: Department of Education, Wilfrid Laurier University, Waterloo, Ontario, N2L 3C5 Canada. Tel: 519.884.0710 x2115. E-mail: jmueller@wlu.ca

Received: February 15, 2017 Accepted: March 14, 2017 Online Published: April 10, 2017

doi:10.5430/jct.v6n1p79 URL: https://doi.org/10.5430/jct.v6n1p79

Abstract

Given a growing digital economy with complex problems, demands are being made for education to address computational thinking (CT) – an approach to problem solving that draws on the tenets of computer science. We conducted a comprehensive content analysis of the Ontario elementary school curriculum documents for 44 CT-related terms to examine the extent to which CT may already be considered within the curriculum. The quantitative analysis strategy provided frequencies of terms, and a qualitative analysis provided information about how and where terms were being used. As predicted, results showed that while CT terms appeared mostly in Mathematics, and concepts and perspectives were more frequently cited than practices, related terms appeared across almost all disciplines and grades. Findings suggest that CT is already a relevant consideration for educators in terms of concepts and perspectives; however, CT practices should be more widely incorporated to promote 21st century skills across disciplines. Future research would benefit from continued examination of the implementation and assessment of CT and its related concepts, practices, and perspectives.

Keywords: *computational thinking; problem solving; technology; elementary education; curriculum*

1. Introduction

Canada's growing digital economy has outpaced the overall economy by over 4 to 1, resulting in a need for 182,000 employees by 2019 (ICTC-CTIC, 2016). Furthermore, there is an emerging demand from politicians (e.g., Barack Obama's CS for All Initiative), industry leaders, educators, and parents to prepare students for careers in the digital world. This demand, combined with a call from computer scientists to attract students, suggests that education include concepts and skills related to Computer Science (CS) (Guzdial, 2008; Hudkins, 2013; Barr, & Stephenson, 2011; Voogt, Fisser, Good, Mishra, & Yadav, 2015; Wing, & Stanzione, 2016). One way to instill this knowledge is to teach computational thinking (CT), broadly defined as how computer scientists think; solving problems using the fundamental tenets of computing (Cuny, Snyder, & Wing, 2010; Wing, 2006).

According to Brennan and Resnick's (2012) framework, CT consists of three components: concepts, practices and perspectives. Seven core concepts are useful in programming, including *sequences, loops, parallelism, events, conditionals, operators,* and *data*. In this framework, eight CT practices refer to how one is learning: *experimenting and iterating*, *testing and debugging, reusing and remixing*, and *abstracting and modularizing*. Finally, three computational perspectives capture how programmers' general outlooks are impacted during CT: *expressing*, *connecting*, and *questioning*. This framework guides the implementation of MIT's Scratch(Note 1), a graphical programming language for beginners. Notably, CT skills are not confined to CS, but can be developed across disciplines and developmental stages.

1.1 CT across Disciplines and Development

Although CT is most often considered in Mathematics and CS, it offers great value across disciplines. CT goes beyond specific skills to refer to a way of thinking that is required to contribute to the digital world (Partnership for 21st Century Skills, 2009). Barr and Stephenson (2011) discuss how *data analysis* in CS refers to writing a statistical program, while in Social Studies *data analysis* refers to identifying trends in data. In Mathematics, *abstraction* is used in algebra, while metaphors in Language are verbal *abstraction*. When writing a paper, students identify a problem or thesis, operationalize the problem, and make arguments to support claims. In CT terms, "computational thinking describes the mental activity in formulating a problem to admit a computational solution," (Wing, 2014, p. 1). Drafting a thesis can be compared to *modularizing*, whereby a solution is reached by assembling something larger with collections of smaller parts (Brennan & Resnick, 2012). One could view supporting a thesis as *debugging*, or testing for functionality. Brennan and Resnick (2012, p. 7-8) describe debugging in a series of steps:

Identify (the source of) the problem

Read through your scripts

Experiment with scripts

Try writing scripts again

Find example scripts that work

These steps can be applied to any task requiring identification and definition of a problem. Bransford and Stein (1993) paralleled Brennan and Resnick's steps in CT with the IDEAL problem solving acronym: identifying problems and opportunities; defining goals and representing the problem; exploring possible solution strategies; anticipating, acting; and, looking back. The process of problem solving in other subjects is therefore in line with the skills, concepts, and perspectives of CT in CS.

Developmental differences could also impact how students learn CT concepts and skills. Developmental psychology pioneer, Piaget, described childrens' use of mental symbols in the preoperational stage (2 to 7 years old). An understanding of concepts as symbols could provide a foundation for learning graphical coding languages. However, Piaget suggested that in the preoperational stage, children are unable to focus on more than one aspect of complex situations (Piaget, 1952), making processes like *connecting* a challenge. In the concrete operational stage (7 to 12 years old), Piaget argued that children grasp logic, a foundational concept in computer programming. Some research indicates that with practice children can develop complex cognitive abilities like abstraction (Carey & Spelke, 1994). Thus, as children develop cognitively, they acquire additional skills that underlie important CT processes. Research on early integration of computer technology suggests that children as young as four years old can learn basic CT concepts (Bers, 2008; 2012). Foundational CT concepts can be useful in children's mathematical development (Kazkoff, Sullivan, & Bers, 2013). One reason that CT concepts may be useful is that children are more capable, earlier, of grasping concepts like abstraction. Vygotsky's (1978) instruction from a, "more knowledgeable other," suggests that we should teach children abstraction early in structured schooling. Indeed, CS experts suggest that concepts like abstraction help younger students to manage complex ideas and tasks (Wing, 2008).

Current approaches to teaching CT exist around the globe, with national mandates to integrate coding in Estonia and the UK (Saez-Lopez, Gonzalez, & Cano, 2016). In Canada, some provincial curricula now include coding from Kindergarten to Grade 12 (The Canadian Press, 2016). Approaches to integrating CT vary; children often learn basic concepts in Mathematics and Language, which will later be valuable in cultivating a CT mindset (Wing, 2008). Other educators use different disciplines to teach CT; for example, a group of educators combined CS with music by teaching students 'musical live coding' with MIT's Scratch program (Ruthmann, Heines, Greher, Laidler, & Saulters, 2010). Integrating subjects reinforced CT concepts and increased student engagement, highlighting the benefits of interdisciplinary collaborations.

Successful introduction of CT is done in a context that interests and engages students (Mueller & Shodiev, 2014). Yet, a full curriculum, course preparation, and ministry requirements place a substantial strain on educators. It is therefore possible that some educators would be disheartened by the possibility of adding CT to their existing teaching load. This paper suggests it is useful to examine where CT concepts and terms already exist in the Ontario elementary school curriculum. If CT terms are already present in the curriculum, educators can adapt existing instruction to emphasize a deeper understanding of CT and how to assess it (Mueller, Beckett, Hennessey, & Shodiev, 2017).

1.2 Is CT Hiding in Plain Sight?

Given the growing importance of CT to a 21st century education (Guzdial, 2008; Barr, & Stephenson, 2011; Voogt et al., 2015; Wing, & Stanzione, 2016), it is of related interest where and how much the present curriculum addresses CT. This inquiry provides a foundation for effective instruction and assessment of CT concepts, practices and perspectives across disciplines. If educators identify CT in their practices, further integration of specific CT skills will be a logical progression. Perhaps CT is merely 'hiding in plain sight' within the Ontario elementary school curriculum; that is, present, but not explicit. We expected to find ‘computational thinking’ and ‘problem solving’ most often in Mathematics and Science and Technology; however, we also predicted that some CT terms would appear in other subjects, although with less frequency.

2. Method

2.1 Materials

The Ontario curriculum is accessible to the public in multiple formats.(Note 2) The Ministry of Education mandates that various subjects be reviewed annually, so materials remain relevant and age-appropriate. We analyzed the most current text files versus the print or PDF(Note 2) files given the large volume of data. Subjects included Mathematics, Arts, Science and Technology, Health and Physical Education, Social Studies, Language, French as a Second Language, Native Languages, Kindergarten, and Growing Success (i.e., a document outlining assessment, evaluation, and reporting in Ontario). Documents varied in length, ranging from 45 (Native Languages) to 322 pages (French as a Second Language; average 182). Most documents (i.e., Mathematics, Arts, Science and Technology, Health and Physical Education, Social Studies, Language, Native Languages), were organized into sections including the ‘front matter’ (i.e., introduction to the subject, program planning, assessment and evaluation), curriculum expectations for each grade (i.e., overall and specific), and a glossary. The document ‘front matter’ addresses how to approach teaching in the discipline and the ‘curriculum expectations’ include the actual content. Three documents strayed from these sections: Growing Success is an assessment and evaluation guide with sections like fundamental principles of teaching and learning; Kindergarten is a general curriculum document with integrated curriculum areas containing sections such as building community and relationships; and French as a Second Language begins in fourth versus first grade.

2.2 Analysis Strategy

We conducted a comprehensive content analysis of the Ontario elementary school curriculum (Grades 1-8) for 44 terms associated with CT processes (e.g., *compute, decomposition, abstraction*; derived from Brennan & Resnick, 2012, see Appendices A and B). Content analysis is a research technique for making inferences from data according to context (Krippendorff, 2012). The goal of this analysis is to review and extract text into meaningful categories, which can then be used to draw conclusions. To first determine frequencies of all terms across the various curricula, we wrote computer code using the Python(Note 3) programming language to extract all CT terms including the focal CT terms from Brennan and Resnick (2012). Various iterations of each term were also included; for example, to search ‘code’, we entered:

```
‘cod(e|ing|ed|er|es)’
```

This strategy permitted us to analyze frequencies of terms per document, but also the broader grade, location, and context surrounding each term. Beyond frequencies, a qualitative inquiry revealed *how* CT terms were used. Following the frequency analysis, we identified key terms by Brennan and Resnick (2012) in context; our code extracted the sentence(s) in which terms appeared. This allowed us to examine whether terms were used about CT or other content. To illustrate the usefulness of such analyses, first consider an example from the present analysis: for example, that various iterations of the term ‘decomposition’ appeared in Mathematics and Science and Technology. However, these citations do not indicate whether the usage was about the computational procedure of breaking down a larger problem into smaller parts or the biological process of breaking down organic tissue. It was therefore necessary to probe frequencies to uncover *how* CT terms were being referenced to prevent erroneous conclusions. Qualitative analyses were conducted by capturing the sentence in which the keyword(s) appeared and creating a database of context for each term. A principal researcher read sentences and determined definitions, later confirmed by a second researcher. Terms included ‘computational thinking,’ ‘problem solving,’ and CT concepts, practices, and perspectives (Brennan & Resnick, 2012).

3. Results and Discussion

3.1 Frequencies and Context Across Curriculum Subject Areas

Problem solving. Iterations of ‘problem solve’ were common in Mathematics (459) and Science and Technology (134). Figure 1 shows frequencies of ‘problem’ and ‘solve’ and their various iterations per grades across subjects. Results showed that ‘problem solve’ appeared most in Mathematics and this terminology increased across grades, peaking in the seventh grade. Problem-solving terms appeared consistently and increased alongside grades in Science and Technology, whereas problem-solving terms peaked in third-grade Arts. Problem-solving language appeared in earlier grades in Language, whereas, these terms were consistently cited throughout grades in Health and Physical

Education. Problem solving appeared most in Social Studies in the seventh grade, whereas, the French as a Second Language curriculum rarely cited problem-solving terms, nor did Native Languages, regardless of grade.

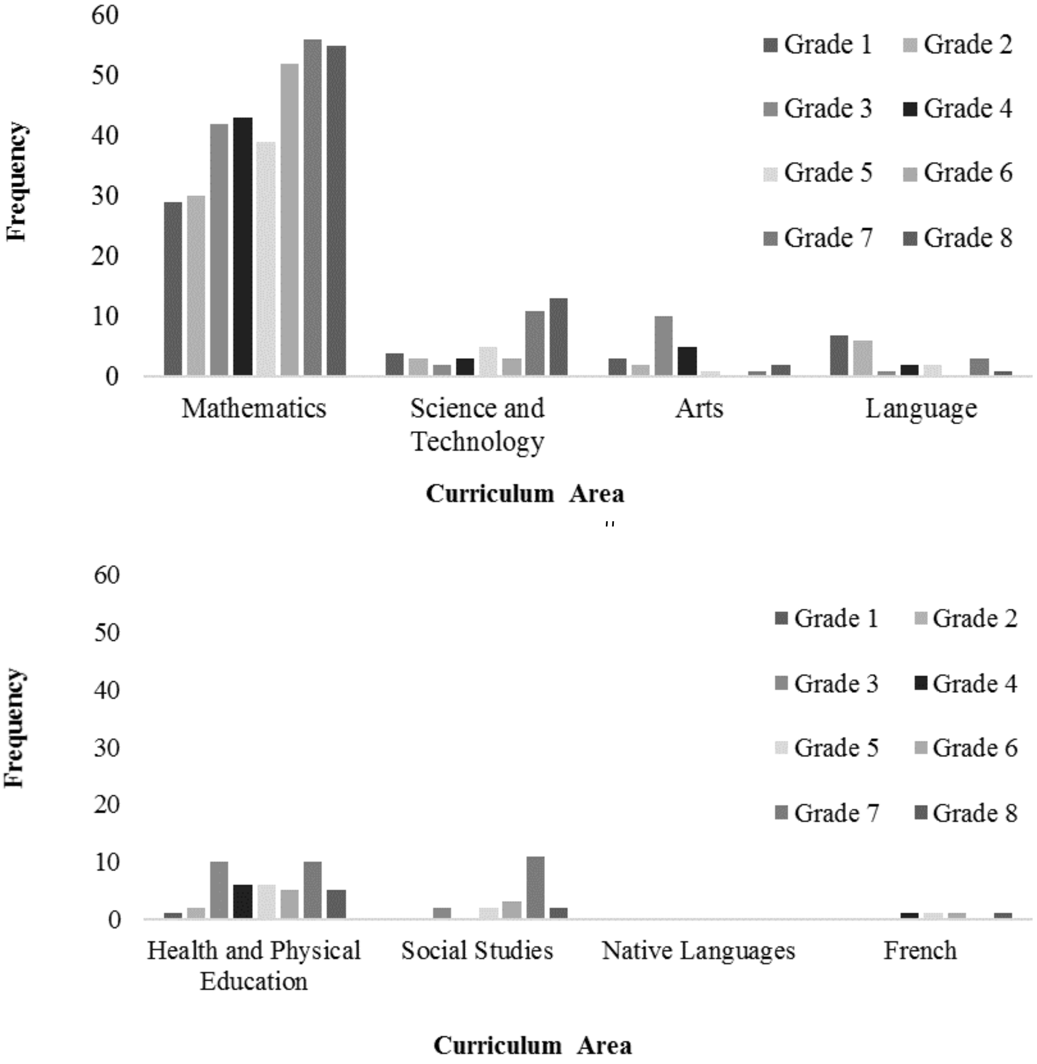

Figure 1. Problem Solving Terminology across All Grades and Curriculum Areas
This figure shows search results for ‘problem + solve’ and all subsequent iterations (i.e., “problem(|)solv(e|ing|ed|er|es)” and “solve (|a |the)problem(|s)”) across curriculum area by grade.

‘Problem’ was used as a noun describing a question, whereas ‘problem solve’ or ‘problem solving’ were used to reference the act of finding solutions. This definition appeared consistent across disciplines, despite being cited with variability. In Mathematics, iterations of ‘problem solving’ (459 instances) referenced a process through which students generate ways to approach challenges: “Encourage talk at each stage of the problem-solving process,” (p .17). Although iterations of ‘problem solving’ were less frequent in Social Studies (49 instances) versus Mathematics, ‘problem solving’ consistently referred to how one solves a dilemma: “Demonstrate collaborative, innovative problem solving,” (p .10). In Arts, iterations of ‘problem solving’ appeared 76 times, all of which referred to the act of finding solutions: “The communication and reflection that occur during and after the process of problem solving help students not only to articulate and refine their thinking but also to see the problem they are solving from different perspectives,” (p. 21).

Computational thinking. The phrase ‘computational thinking’ was not referenced in the Ontario elementary school curricula. ‘Computational’ alone appeared in Mathematics (frequencies in parentheses; 25), and once in Social Studies (i.e., “In addition to providing opportunities for literacy development, social studies, history, and geography also reinforce mathematical literacy, in areas involving computational strategies and data management and, in particular, the ability to read and construct graphs," p. 49). Iterations of 'compute' appeared mostly in Mathematics (30), but were also cited in Social Studies (12), Language (11), Arts (10), Kindergarten (9), HPE (6), Science and Technology (3), French as a Second Language (2), and Native Languages (1). Iterations of 'compute' were absent from Growing Success.

While the specific phrase 'computational thinking' was absent from the curricula, 'computational' appeared 25 times in Mathematics to reference strategies to facilitate problem solving: "Students need to develop the ability to select the appropriate electronic tools, manipulatives, and computational strategies to perform particular mathematical tasks" (p. 111; eighth-grade specific expectations). Thus, 'computational' referenced a calculation strategy rather than a general way of thinking. 'Computational' appeared once in the front matter of Social Studies to reference computational strategies learned elsewhere in the curriculum: "… social studies, history, and geography also reinforce mathematical literacy, in areas involving computational strategies," (p .49). Across disciplines 'computational' consistently referred to a series of steps used to arrive at a solution to a problem.

'Compute' in Mathematics referenced performing calculations: "They [students] also develop a solid understanding of the four basic operations and learn to compute fluently, using a variety of tools and strategies," (p. 8). 'Computer' appeared in Mathematics to reference the technological device, reflecting the presence of technology in the classroom: "The computer and the calculator should be viewed as important problem-solving tools to be used for many purposes," (p. 15). In Language, 'computer' referenced digital equipment or software: "Computer programs can help students collect, organize, and sort the data they gather and to write, edit, and present reports on their findings," (p .30). Across different subjects 'computer' consistently referred to technological equipment that facilitates learning.

3.2 Comprehensive List of CT Terminology

Initial frequency analyses indicated that all CT terms (including concepts, practices, and perspectives; frequencies in parentheses), appeared most in the Mathematics (1259), French as a Second Language (1199), and Social Studies (1075) curricula. The full list of terms was less frequent in Arts (935), Science and Technology (886), Health and Physical Education (676), Language (522), and Kindergarten (372). The full list of terms was least common in Growing Success (186), and Native Languages (103).

To assess the most frequent terms across all curricula, we sorted initial frequencies into descending order and compiled a 'top 15' list (Table 1). Table 1 shows that 'questioning' was the most common CT term in Language, Social Studies, and Kindergarten. Across all subjects, iterations of 'applying' were second-most cited. Third-most common was 'events' in Language, French as a Second Language, and Social Studies.

Table 1. Top 15 Most Frequently Cited CT Terms Across All Curriculum Subjects

<table>
<tr><th>Arts</th><th><i>N</i></th><th>Language</th><th><i>N</i></th><th>Math</th><th><i>N</i></th></tr>
<tr><td>1. represent(|ation|ing|ed|s)</td><td><i>145</i></td><td>1. question(|ing|ed|er|s)</td><td><i>95</i></td><td>1. problem(|s)(?!(-|)solv(e|ing|ed|er|es))</td><td><i>325</i></td></tr>
<tr><td>2. apply(|ing)</td><td><i>99</i></td><td>2. represent(|ation|ing|ed|s)</td><td><i>53</i></td><td>2. represent(|ation|ing|ed|s)</td><td><i>224</i></td></tr>
<tr><td>3. analysis</td><td><i>71</i></td><td>3. event(|s)</td><td><i>53</i></td><td>3. solve (|a |the)problem(|s)</td><td><i>96</i></td></tr>
<tr><td>4. problem(|s)(?!(-|)solv(e|ing|ed|er|es))</td><td><i>52</i></td><td>4. express(|ing|ed|es)</td><td><i>42</i></td><td>4. apply(|ing)</td><td><i>82</i></td></tr>
<tr><td>5. technologi(cal|es)</td><td><i>45</i></td><td>5. analy(s|z)e</td><td><i>33</i></td><td>5. model(|s)</td><td><i>56</i></td></tr>
<tr><td>6. abstract(|ion|ing|ed|s)</td><td><i>38</i></td><td>6. connect(|ing|ed|er|s)</td><td><i>32</i></td><td>6. collect(|ion|ing|ed|or|s)</td><td><i>55</i></td></tr>
<tr><td>7. technology</td><td><i>30</i></td><td>7. problem(|s)(?!(-|)solv(e|ing|ed|er|es))</td><td><i>28</i></td><td>7. event(|s)</td><td><i>47</i></td></tr>
<tr><td>8. analy(s|z)e</td><td><i>29</i></td><td>8. sequenc(e|ing|ed|er|es)</td><td><i>27</i></td><td>8. problem(-|)solv(e|ing|ed|er|es)</td><td><i>38</i></td></tr>
<tr><td>9. model(|s)</td><td><i>28</i></td><td>9. apply(|ing)</td><td><i>22</i></td><td>9. experiment(|ation|ing|ed|er|s)</td><td><i>37</i></td></tr>
<tr><td>10. appli(cation|ed|es)</td><td><i>19</i></td><td>10. logic(|al)</td><td><i>20</i></td><td>10. connect(|ing|ed|er|s)</td><td><i>37</i></td></tr>
<tr><td>11. experiment(|ation|ing|ed|er|s)</td><td><i>16</i></td><td>11. model(|l)(ing|ed)</td><td><i>16</i></td><td>11. comput(e|ation|ing|ed|er|es)</td><td><i>30</i></td></tr>
<tr><td>12. solve (|a |the)problem(|s)</td><td><i>15</i></td><td>12. model(|s)</td><td><i>13</i></td><td>12. general</td><td><i>22</i></td></tr>
<tr><td>13. model(|l)(ing|ed)</td><td><i>13</i></td><td>13. comput(e|ation|ing|ed|er|es)</td><td><i>11</i></td><td>13. express(|ing|ed|es)</td><td><i>22</i></td></tr>
<tr><td>14. general</td><td><i>11</i></td><td>14. solve (|a |the)problem(|s)</td><td><i>10</i></td><td>14. question(|ing|ed|er|s)</td><td><i>20</i></td></tr>
<tr><td>15. comput(e|ation|ing|ed|er|es)</td><td><i>10</i></td><td>15. appli(cation|ed|es)</td><td><i>10</i></td><td>15. decompos (e|ition|ing|ed|er|es)</td><td><i>19</i></td></tr>
</table>

Science & Technology	*N*	French as a Second Language	*N*	Health & Physical Education	*N*
1. technology	*202*	1. model(|s)	*297*	1. apply(|ing)	*159*
2. question(|ing|ed|er|s)	*110*	2. question(|ing|ed|er|s)	*277*	2. problem(|s)(?!(-|)solv(e|ing|ed|er|es))	*65*
3. technologi(cal|es)	*101*	3. event(|s)	*150*	3. question(|ing|ed|er|s)	*56*
4. problem(|s)(?!(-|)solv(e|ing|ed|er|es))	*82*	4. express(|ing|ed|es)	*134*	4. appli(cation|ed|es)	*55*
5. test(|ing|ed|er|s)	*75*	5. model(|l)(ing|ed)	*66*	5. connect(|ing|ed|er|s)	*52*
6. experiment(|ation|ing|ed|er|s)	*43*	6. apply(|ing)	*38*	6. sequenc(e|ing|ed|er|es)	*40*
7. problem(-|)solv(e|ing|ed|er|es)	*37*	7. analy(s|z)e	*29*	7. management	*31*
8. model(|s)	*33*	8. connect(|ing|ed|er|s)	*29*	8. model(|s)	*26*
9. appli(cation|ed|es)	*28*	9. represent(|ation|ing|ed|s)	*27*	9. express(|ing|ed|es)	*22*
10. apply(|ing)	*17*	10. problem(|s)(?!(-|)solv(e|ing|ed|er|es))	*23*	10. event(|s)	*21*
11. analy(s|z)e	*16*	11. technology	*22*	11. technology	*20*
12. solve (|a |the)problem(|s)	*15*	12. sequenc(e|ing|ed|er|es)	*16*	12. general	*17*
13. general	*14*	13. appli(cation|ed|es)	*15*	13. analy(s|z)e	*17*
14. represent(|ation|ing|ed|s)	*14*	14. logic(|al)	*8*	14. test(|ing|ed|er|s)	*14*
15. event(|s)	*14*	15. collect(|ion|ing|ed|or|s)	*8*	15. represent(|ation|ing|ed|s)	*13*
Social Studies	*N*	Native Languages	*N*	Kindergarten	*N*
1. question(|ing|ed|er|s)	*401*	1. express(|ing|ed|es)	*24*	1. question(|ing|ed|er|s)	*59*
2. analy(s|z)e	*169*	2. apply(|ing)	*23*	2. problem(|s)(?!(-|)solv(e|ing|ed|er|es))	*46*
3. event(|s)	*166*	3. question(|ing|ed|er|s)	*18*	3. model(|s)	*30*
4. represent(|ation|ing|ed|s)	*56*	4. technology	*8*	4. represent(|ation|ing|ed|s)	*28*
5. apply(|ing)	*39*	5. represent(|ation|ing|ed|s)	*7*	5. problem(-|)solv(e|ing|ed|er|es)	*27*
6. technologi(cal|es)	*34*	6. general	*4*	6. apply(|ing)	*23*
7. problem(|s)(?!(-|)solv(e|ing|ed|er|es))	*31*	7. appli(cation|ed|es)	*4*	7. express(|ing|ed|es)	*21*
8. connect(|ing|ed|er|s)	*30*	8. event(|s)	*4*	8. experiment(|ation|ing|ed|er|s)	*18*
9. appli(cation|ed|es)	*22*	9. test(|ing|ed|er|s)	*4*	9. technology	*16*
10. collect(|ion|ing|ed|or|s)	*19*	10. analysis	*2*	10. connect(|ing|ed|er|s)	*15*
11. model(|s)	*18*	11. comput(e|ation|ing|ed|er|es)	*1*	11. solve (|a |the)problem(|s)	*14*
12. solve (|a |the)problem(|s)	*14*	12. analy(s|z)e	*1*	12. model(|l)(ing|ed)	*13*
13. technology	*13*	13. collect(|ion|ing|ed|or|s)	*1*	13. event(|s)	*13*
14. comput(e|ation|ing|ed|er|es)	*12*	14. sequenc(e|ing|ed|er|es)	*1*	14. comput(e|ation|ing|ed|er|es)	*9*
15. express(|ing|ed|es)	*12*	15. conditional(|s)	*1*	15. sequenc(e|ing|ed|er|es)	*9*
Growing Success	*N*	Growing Success	*N*	Growing Success	*N*
1. test(|ing|ed|er|s)	*24*	6. collect(|ion|ing|ed|or|s)	*11*	11. question(|ing|ed|er|s)	*7*
2. represent(|ation|ing|ed|s)	*22*	7. cod(e|ing|ed|er|es)	*10*	12. model(|l)(ing|ed)	*6*
3. appli(cation|ed|es)	*21*	8. problem(|s)(?!(-|)solv(e|ing|ed|er|es))	*9*	13. management	*6*
4. technology	*15*	9. apply(|ing)	*9*	14. general	*4*
5. express(|ing|ed|es)	*14*	10. model(|s)	*8*	15. logic(|al)	*4*

3.3 CT framework Terminology

Table 2 presents results of a search for Brennan and Resnick's (2012) specific CT framework terms (i.e., concepts, practices, and perspectives). This framework was ideal for this analysis because it includes explicit terms more specifically related to coding and CT. In general, these terms were most common in Mathematics, Social Studies, and French as a Second Language. CT framework terms were also somewhat frequent in Language, Science and Technology, Arts, Health and Physical Education, and Kindergarten. These terms were least common in Growing Success, and Native Languages. Table 2 indicates that CT *concepts* were common in Mathematics and Social Studies, whereas CT practices were only sparsely cited in Science and Technology and Growing Success and totally absent from most other curricula. This may be a result of searching for very particular set of CT terms; it is possible that other language may indicate CT practices (i.e., perhaps instead of using s *remixing* and *debugging*, practices are described as *trying different methods* and *identifying problems*), but there was no evidence of specific CT practices. CT perspectives are relevant to language acquisition and it is therefore not surprising to see French as a Second Language and Language top the list. However, these results do suggest that across disciplines, students are taught to express, question, and connect – all of which are key perspectives underlying CT.

Table 2. Frequencies of CT Concepts, Practices and Perspectives across All Curriculum Subjects

Subjects	M	S	L	A	HPE	FSL	SS	GS	NL	K
CONCEPTS										
Conditionals	0	0	0	0	0	0	0	0	0	0
Data	253	28	5	3	3	3	114	18	0	8
Events	21	12	33	36	18	84	136	0	4	12
Loops	0	0	0	0	0	0	0	0	0	0
Operators	0	3	0	0	0	0	2	0	0	0
Parallelism	0	0	0	0	0	0	0	0	0	0
Sequences	0	0	3	8	3	0	0	0	0	0
Subtotals	*274*	*43*	*41*	*47*	*24*	*87*	*252*	*18*	*4*	*20*
PRACTICES										
Abstracting	0	0	0	1	0	0	0	0	0	0
Debugging	0	0	0	0	0	0	0	0	0	0
Incremental	0	0	0	0	0	0	0	0	0	0
Iterative	0	0	0	0	0	0	0	0	0	0
Modularizing	0	0	0	0	0	0	0	0	0	0
Remixing	0	0	0	0	0	0	0	0	0	0
Reusing	0	3	0	0	0	0	0	0	0	0
Testing	0	19	0	0	0	0	0	2	0	0
Subtotals	*0*	*21*	*0*	*1*	*0*	*0*	*0*	*2*	*0*	*0*
PERSPECTIVES										
Connecting	23	2	17	7	2	16	0	0	0	1
Expressing	2	1	4	2	4	21	0	0	2	2
Questioning	1	1	9	4	10	14	8	2	0	5
Subtotals	*26*	*4*	*30*	*13*	*16*	*51*	*8*	*2*	*2*	*8*
Totals	300	69	71	62	40	138	260	22	6	28

M = Mathematics
S & T = Science and technology
L = Language
A = Arts
HPE = Health and physical education
FSL = French as a Second Language
SS = Social Studies
GS = Growing Success
NL = Native Languages

Concepts. CT *concepts* were most common in Mathematics and Social Studies. CT concepts were next most frequent in French as a Second Language, Arts, Science and Technology, Language, Health and Physical Education, Kindergarten, Growing Success, and Native Languages. ‘Data’ was cited in all subjects but Native Languages to refer to information. ‘Data’ appeared 253 times in Mathematics to reference quantitative representations of information: “Students can use calculators or computers to perform operations, make graphs, and organize and display data,” (p .14). Mathematics also cited ‘databases’ to refer to large collections of information: “Students also learn how to locate relevant information from a variety of sources such as statistical databases, newspapers, and reports,” (p. 29). ‘Data’ and ‘database’ appeared less often in Arts (5 instances), and referenced a tool to project visual information: “Use a data projector to project evocative imagery,” (p. 125).

‘Events’ appeared in all subjects except Growing Success and referred to points in time. Across disciplines, the meaning of ‘events’ was consistent (e.g., in Mathematics, “use probability models to make predictions about real-life events,” p. 118; in CT terms, “one thing causing another thing to happen,” Brennan & Resnick, 2012, p. 4). In Arts, ‘events’ also referred to points in time, for instance, in a dramatic arts class students may be instructed to “Use group improvisation to work out a time line of events in a drama story,” (p. 151).

‘Operators’ appeared sporadically (three instances), but consistently, in Science and Technology to refer to people in specific roles; for instance, as ‘landfill operators’ (p. 105), or in Social Studies as ‘ecotourism operators’ (p. 167). This usage differs from the CT definition of operators; that is, as terms in programming language that, “provide support for mathematical expressions enabling the programmer to perform numeric and string manipulations” (Brennan & Resnick, 2012, p. 5). The CT concept ‘sequences’ appeared in Language, wherein each of the three citations referred to ‘letter sequences’ (p. 43). In Health and Physical Education, all three citations of ‘sequences’ referenced ‘movement or dance sequences’ (p.118). The same was true in Arts wherein eight citations of ‘sequences’ referred to a series of dance steps (p.100). This may appear different from the CT definition of sequences (i.e., “a sequence of programming instructions that specifies the behaviour or action that should be produced”; Brennan & Resnick, 2012, p. 3); however, both citations simply refer to a series of steps.

The remaining CT concepts (i.e., *loops*, *parallelism*, and *conditionals*) were absent from the curricula. Concepts like ‘data’ and ‘sequences’ appeared across subjects, suggesting their cross-disciplinary relevance. It is however, possible that different language indicating similar concepts existed; for instance, while ‘loops’ was absent (i.e., “a mechanism for running the same sequence multiple times,” Brennan & Resnick, 2012; p. 4), ‘repeating’ was common in Mathematics (31 instances; e.g., in Overall Expectations: “By the end of Grade 2, students will identify, describe, extend, and create repeating patterns,” p. 49). Thus, specific CT concept terminology may be absent from the curricula while broader CT concepts hide in plain sight.

Practices. CT *practices* appeared less frequently than concepts and perspectives, but were nonetheless most common in Science and Technology, Growing Success, and Arts. CT practices were absent from Mathematics, French as a Second Language, Language, Social Studies, Health and Physical Education, Kindergarten, Growing Success, and Native Languages.

‘Testing’ appeared once in Mathematics in curriculum expectations: “collecting data, questioning, testing, revising, modelling, solving, inferring, forming conclusions,” (p. 22), but appeared more often in Science and Technology, wherein each of the 19 citations referred to trying different approaches to solving a problem: “When designing, building, and testing devices, it is important that students be able to identify and explain the importance of practices that ensure their personal safety and the safety of others,” (p. 146). ‘Testing’ appeared twice in Growing Success in the context of assessing competencies: “…reviewing the impact of testing on students,” (p. 29). ‘Reusing’ appeared three times in the Science and Technology document in the context of environmental sustainability: “reusing and recycling what we can,” (p. 5). This is similar to the CT definition of reusing, which refers to creating new code by building on existing projects (Brennan & Resnick, 2012; p. 8). ‘Abstracting’ appeared once in Arts as a teacher prompt to ask how an artist manipulates the subject: “What message is the artist conveying by distorting and abstracting the subject?” (p. 108). In CT terms, abstraction is often paired with modularizing, and refers to “building something large by putting together collections of smaller parts,” (Brennan & Resnick, 2012; p. 9). In computing, abstraction involves disguising computational intricacies that may be involved in a complex process, with more accessible and relatable concepts, allowing programmers to provide users with only the details they want them to see. Similarly, a complex social issue may be represented with a single piece of art, which captures the artist’s perspective and the message they hope to convey, while ignoring any details the artist deems unnecessary. Thus, across disciplines ‘abstraction’ shared a similar definition.

Other CT practices (i.e., incremental, iterative, debugging, remixing, and modularizing) were absent from the curricula.

However, it is possible that the documents used different terminology to refer to similar processes delineated by Brennan and Resnick (2012). For example, according to Brennan and Resnick (2012), ‘incremental’ refers to “the process of identifying a concept, developing a plan for the design, and then implementing the design in code,” (p. 7). Conceptually, incremental refers to gradual processes –if one searches the Mathematics curriculum for ‘gradual,’ the following passage is among the results and supports incremental learning practices:

> “The development of mathematical knowledge is a gradual process. A continuous, cohesive program throughout the grades is necessary to help students develop an understanding of the “Big ideas” of Mathematics – that is, the interrelated concepts that form a framework for learning Mathematics in a coherent way” (p. 4).

CT practices like ‘testing’ and ‘reusing’ were used consistently (although infrequently) across disciplines. Regardless of context (e.g., ‘reusing’ in science to refer to environmental conservation versus ‘reusing’ a piece of computer code), our findings suggest that the CT practices are somewhat present in the current curriculum. This is encouraging because the quantitative analysis showed that CT practices were virtually absent from the curriculum. However, if it is a matter of semantics and not that practices are missing, perhaps transferring these practices between disciplines may be easier than previously thought.

Perspectives. CT *perspectives* appeared in various parts of the curricula, but were most common in French as a Second Language, Language, Mathematics Health and Physical Education, Arts, Social Studies, Science and Technology, followed by Growing Success and Native Languages. ‘Connecting’ appeared in Mathematics 23 times and each citation referred to linking ideas. For example, a specific curriculum expectation in first-grade reads: “Connecting: To make connections among simple mathematical concepts and procedures, and relate mathematical ideas to situations drawn from everyday contexts,” (p. 32). Similarly, in Language, ‘connecting’ was referenced as a way to increase knowledge: “extend understanding of texts by connecting the ideas in them to their own knowledge, experience, and insights, to other familiar texts, and to the world around them,” (p. 83). In French as a Second Language, ‘connecting’ appeared 16 times, consistently referring to meaning making in learning: “Activating prior knowledge and connecting learning to past experiences help students acquire French literacy skills,” (p. 45).

‘Expressing’ appeared 21 times in French as a Second Language to refer to communicating ideas. For instance, one instruction from eighth-grade specific expectations stated, “teachers can suggest students use various verb tenses when expressing their opinions about an advertising campaign,” (p. 301). In Brennan and Resnick’s (2012) framework, ‘expressing’ represents a similar perspective; that is, “A computational thinker sees computation as a medium and thinks, ‘I can express my ideas through this new medium,’” (p. 10). ‘Expressing’ was reliably cited across disciplines in which students learn language (i.e., French or programming).

‘Questioning’ appeared 14 times in French as a Second Language as a perspective that students acquired over time: “The questioning they [students] practiced in the early grades becomes more sophisticated as they learn that all sources of information have a particular point of view,” (p. 46). Similarly, in Health and Physical Education, ‘questioning’ appeared 10 times as a skill that educators should support because, “rich, open-ended questioning is the starting point for effective inquiry or for addressing a problem,” (p. 71). According to Brennan and Resnick (2012), ‘Questioning’ involves interrogating the taken-for-granted, and, in some cases, responding to that interrogation through design,” (p. 11). Across seemingly unrelated disciplines like Health and Physical Education and CS, the value of cultivating a questioning perspective was evident.

In sum, the content analysis demonstrated that CT concepts were most frequent in Mathematics and CT practices were most frequent in Science and Technology. This finding is logical given that Mathematics concepts underlie some CT processes and scientific methods (e.g., testing and re-testing). A more surprising finding related to the location of CT terminology was that CT practices appeared in the Growing Success document, suggesting that measurement of problem solving and CT practices may be discussed in terms of assessment and evaluation. Finally, results showed that CT perspectives were most common in French as a Second Language and Language, suggesting that learning oral languages requires similar perspectives to learning programming languages.

3.4 Frequencies and Context Across Grades

Four focal subjects (i.e., Arts, Language, Science and Technology, and Mathematics) were further explored by grades to illustrate how problem solving, broader CT terms, and terms from Brennan and Resnick (2012) appeared across developmental stages. We chose these subjects because the first three are core subjects, whereas the latter provides contrast in terms of time dedicated to teaching and assessment.

Arts curriculum. Iterations of ‘computational thinking’ were absent in Arts from grades one to eight. Figure 2 shows that iterations of ‘problem solving’ appeared throughout all grades, but to a lesser degree, peaking in the third grade. CT *concepts* appeared in all grades to some extent, whereas CT *practices* appeared only once in fourth grade. CT *perspectives* appeared in the first and eighth-grades only, potentially indicating the relevance of conceptual learning at the start and end of elementary education.

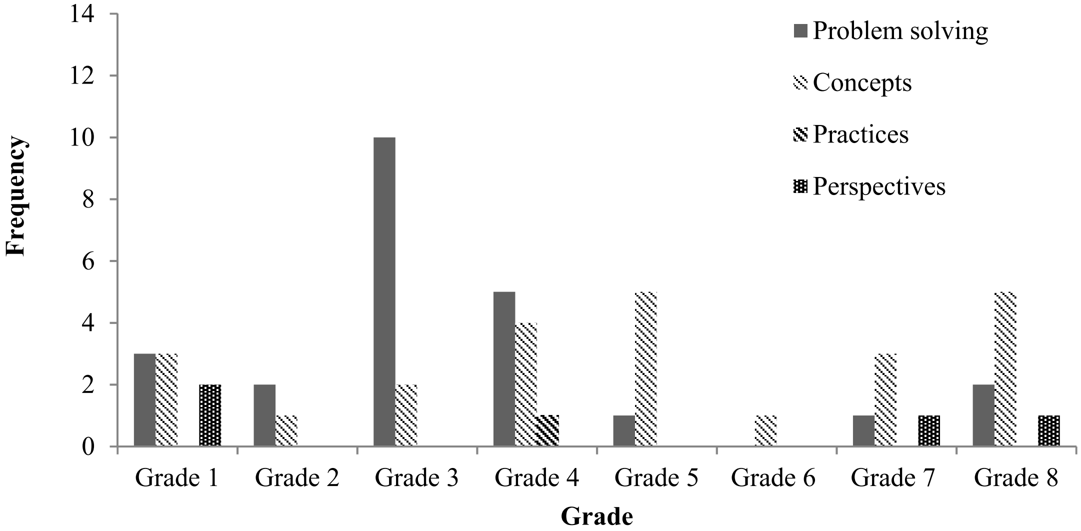

Figure 2. Problem Solving and CT Framework Terms across Grades in Arts

This figure depicts frequencies of ‘problem solving,’ and CT framework terms in Arts by grades.

Language curriculum. Iterations of ‘computational thinking’ were absent in Language from grades one to eight. Figure 3 shows that iterations of ‘problem solving’ appeared sparsely in all grades, peaking in first grade. CT *concepts* were most frequent in second grade, whereas, CT *practices* were completely absent across all grades. CT *perspectives* appeared most in first grade Language, suggesting the importance of such perspectives in disciplines beyond CS.

#
#
#
#

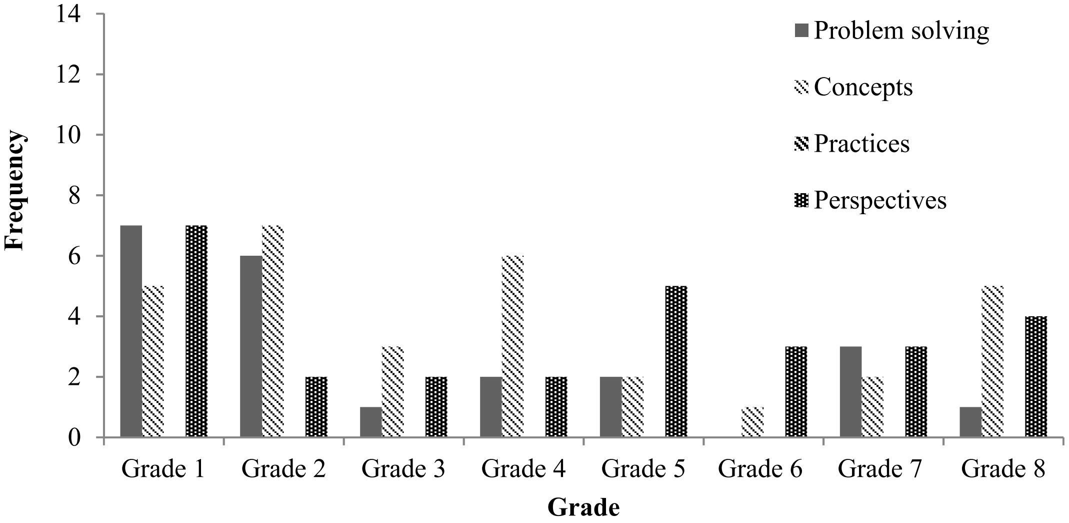

Figure 3. Problem Solving and CT Framework Terms across Grades in Language

This figure depicts frequencie of ‘problem solving,’ and CT framework terms in Language by grades.

Science and technology curriculum. Iterations of 'computational thinking' were absent from Science and Technology in grades one to eight. Figure 4 shows that iterations of 'problem solving' appeared throughout all grades to some extent, peaking in eighth-grade Science and Technology. CT *concepts* most commonly appeared in fifth-grade, whereas CT *practices* appeared with some frequency from fifth through eighth-grade. CT *perspectives* appeared only once in third-grade.

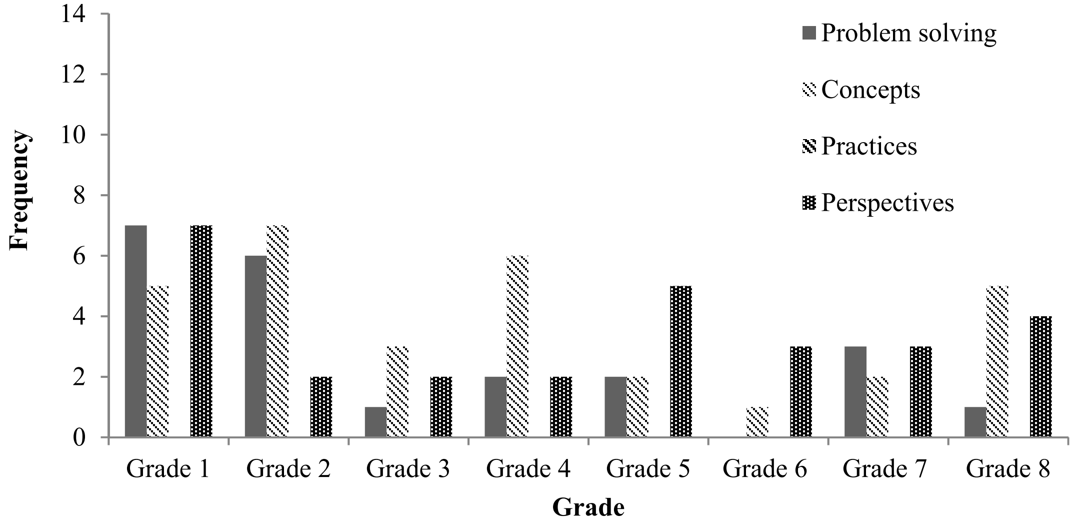

Figure 4. Problem Solving and CT Framework Terms across Grades in Science and Technology

This figure depicts frequencies of 'problem solving,' and CT framework terms in Science and Technology by grades.

Mathematics curriculum. Figure 5 shows that CT terminology was cited with greater frequency in Mathematics compared to other subjects. Iterations of 'computational thinking' appeared in Mathematics at least once across grades one to eight. Iterations of 'problem solving' appeared across all levels, peaking in senior grades. CT *concepts* were frequent in Mathematics versus other subjects, with most references in fifth-grade. CT *practices* were absent from all grades, whereas CT *perspectives* appeared in first-grade and sparsely in later grades.

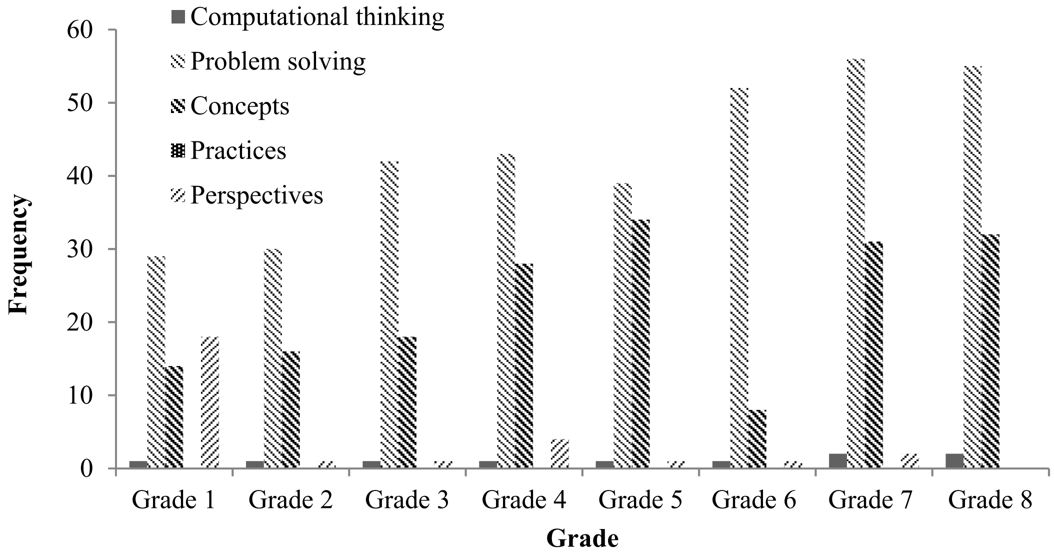

Figure 5. CT Terms and Problem Solving across Grades in Mathematics

This figure depicts frequencies of 'computational thinking,' 'problem solving,' and CT terms in Mathematics separated by grades.

3.5 Location Within the Curriculum Documents

The location of all terms was documented according to a coding scheme (Appendix C), which was collapsed into four categories: 1) front matter – introduction, 2) overall curriculum expectations, 3) specific curriculum expectations, and 4) the glossary. Table 3 displays results for subjects excluding Kindergarten and Growing Success. CT concepts were frequent in specific expectations within Mathematics, and the most common term was ‘data.’ This is predictable, given Mathematics houses CT concepts like *data, events*, and *sequences*. In Science and Technology, most CT concepts were in the front matter and 26 of 31 instances were ‘data.’ In Arts, CT concepts were mostly cited in specific expectations; however, concepts were also in the front matter, and most references were to ‘events.’ Language, like Mathematics, cited CT concepts most often in specific expectations and most references were to ‘events.’ CT concepts were distributed somewhat equally across the front matter and specific expectations of Social Studies and most references were to ‘events’ and ‘data.’ In French as a Second Language, CT concepts were mostly in specific expectations, and the most frequent term was ‘events.’ In Health and Physical Education, most CT concepts appeared in the front matter, yet, the most common term was still ‘events.’ CT concepts were absent from Native Languages.

Table 3. CT Concepts, Practices, and Perspectives Across Subjects by Location

				Location
CONCEPTS		Overall	Specific	
Subject	Front Matter	Expectations	Expectations	Glossary
Mathematics	13	64	117	61
Science and Technology	31	3	7	1
Arts	10	3	21	13
Language	3	0	31	6
Social Studies	56	20	66	10
French as a Second Language	12	0	19	1
Health and Physical Education	19	0	4	0
Native Languages	3	0	1	0
		Overall	Specific	
PRACTICES	Front Matter	Expectations	Expectations	Glossary
Subject				
Mathematics	0	0	0	0
Science and Technology	8	6	7	0
Arts	0	0	1	0
Language	0	0	0	0
Social Studies	0	0	0	0
French as a Second Language	0	0	0	0
Health and Physical Education	0	0	0	0
Native Languages	0	0	0	0
		Overall	Specific	
PERSPECTIVES	Front Matter	Expectations	Expectations	Glossary
Subject				
Mathematics	7	0	14	3
Science and Technology	3	1	0	0
Arts	7	1	3	2
Language	2	0	23	3
Social Studies	8	0	0	0
French as a Second Language	9	0	8	1
Health and Physical Education	15	0	1	0
Native Languages	1	0	1	0

CT practices were absent from Mathematics; however, in Science and Technology CT practices appeared somewhat evenly distributed across the front matter and expectations sections. ‘Testing’ was most frequent, yet, in Language, Social Studies, Health and Physical Education, French as a Second Language, and Native Languages, ‘testing’ was not referenced in any location. This is interesting because testing is important across disciplines; for instance, to test one’s vocabulary in another language is to test one's code to achieve the desired outcome. However, testing in science could imply an empirical approach, while other subjects use different approaches to demonstrate knowledge.

CT perspectives appeared most in specific expectations within Mathematics, and all references were to 'connecting.' However, the front matter also referenced 'connecting' emphasizing the importance of a connecting approach to teaching. Mathematics differed from other subjects in that the front matter referenced connecting, whereas others cited 'questioning.' One possible explanation for this difference is that Mathematics emphasizes connecting ideas, whereas other subjects emphasize questioning existing knowledge.

A single instance of each CT perspective appeared in the front matter of Science and Technology, whereas Arts referenced CT perspectives most in the front matter, and cited 'connecting' and 'questioning.' Language referenced 'connecting' most often and in specific expectations. In Health and Physical Education and Social Studies, 'questioning' appeared most often and in the front matter section. In French as a Second Language 'questioning' appeared most frequently of the three CT perspectives and was distributed somewhat equally across the front matter and specific expectations. In Native Languages, 'expressing' appeared once in the front matter, and once in specific expectations.

4. Conclusion

The importance of CS and CT is clear in many contexts, including education, business, and politics. What is less clear is the extent to which CT already exists within the current Ontario elementary curriculum and where the development of CT might best be integrated. The current study therefore conducted a comprehensive content analysis of the Ontario Ministry of Education Elementary curriculum to identify where and how much CT terminology could be 'hiding in plain sight.' That is, we explored the curriculum to see if CT terms appeared, and if so, to what extent and how. Such analysis will be useful as educators draw connections between the existing and already full curriculum and emerging CT concepts, skills, and perspectives.

The content analysis examined frequencies, locations and context of CT terms through both quantitative and qualitative methods. Results showed that although the specific phrase 'computational thinking' was not present at all, problem-solving terms as a process were apparent across various subjects, especially in the Mathematics curriculum. Similarly, the full list of CT terms (and their iterations) appeared mostly in the Mathematics curriculum, but also in unexpected subjects such as French as a Second Language and Social Studies. Identifying the top 15 most common CT terms in each curriculum showed that 'questioning,' 'applying,' and 'events' were frequent across disciplines. This suggests that educators already use some concepts and perspectives central to the definition of CT (Brennan & Resnick, 2012). Results also showed that CT concepts and perspectives were more common than CT practices. This finding could be linked to our framework, or, an actual gap in the curriculum related to CT practices.

The implementation of new content into the curriculum can be challenging for educators. Integrating digital technology, for example, is an ongoing issue for educators in modern classrooms. Although barriers have included external variables, like lack of hardware and connectivity issues, internal variables like lack of knowledge and motivation can also make implementation difficult (Mueller, Wood, Willoughby, Ross, & Specht, 2008). Requiring educators to address new concepts, practices, and perspectives could be stalled if new curriculum content must be simultaneously added. Examination of existing curricula and the identification of CT terms can therefore act as a scaffold to incorporating computational content.

In terms of when the curriculum references CT, results demonstrated that in first-grade, students are exposed to problem-solving terms, whereas specific CT terms appeared in senior grades. Most CT terms were in specific expectations, indicating that educators already use some CT terms in their practices. Incorporating new CT terms, including concepts, practices and perspectives may therefore be less onerous than assumed. Finally, our qualitative analysis was a useful exercise in identifying how CT concepts like 'data' were cited consistently across subjects.

Our analyses reflect only one provincial curriculum, but nonetheless, it is valuable to provide an initial content analysis to spark discussion about where CT concepts, practices and perspectives exist in the curriculum. As illustrated by results of the present study, current implementation of CT expectations occurs mostly in Mathematics and associated curriculum areas. However, there will likely be unexpected consequences for limiting CT to these particular disciplines. Indeed, only a select portion of the student population will be exposed to expectations that prepare them to work and learn in technological fields. The limited frequency of CT terms may be related to the specificity of the search terms included in our analyses. While the list of terms was drawn from a current and reputable framework and definition of CT (Brennan & Resnick, 2012; Wing, 2006), it is possible that our search was too narrow to identify instances of CT terminology. Therefore, a next step in examining CT across areas would be to explore instances of skills, practices and perspectives using additional vocabulary to capture CT constructs across contexts and expand understanding of what CT looks like across disciplines.

A strength of this paper is that we approached the research question from an interdisciplinary perspective; that is, psychologists, educators, and computer scientists conducted the study. This provided a unique perspective from which to analyze the data and an opportunity to increase productivity. The efficiency of our analyses was improved with CT itself by writing code to readily identify all instances of keywords and iterations across subjects. This paper contributes to the theoretical literature on CT that still debates the precise definition of the construct (Berry, 2015; Perkovic, Settle, Hwang, & Jones, 2010). Our findings suggest that the present Ontario elementary school curriculum references CT, but also that CT concepts and perspectives exist beyond Mathematics and CS. This research has applied implications for educators and students in recognizing where CT may already occur in the current curriculum. It is increasingly important to instill CT foundations in early education, given that using computers to solve problems can enhance students' abilities in solving real-world problems (Voskoglou & Buckley, 2012). Future research and practice should therefore focus on increasing CT practices across disciplines so students can competently exercise 21st century skills. Further advances in curriculum design and instructional strategies need to address the assessment of CT and effective practice to best support the development of CT and its related concepts, skills, and perspectives.

Acknowledgments

The authors acknowledge the support of a Social Science and Humanities Research (SSHRC) Insight Development Grant awarded to the second and third authors (430-2014-00680).

References

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the CS education community? *ACM Inroads, 2*(1), 48-54. https://doi.org/10.1145/1929887.1929905

Berry, M. (2015). Coding and digital skills. *Remarks for eSkills at Schools event, European Parliament, Brussels.* Retrieved from http://milesberry.net/2015/10/coding-and-digital-skills/

Bers, M. (2008). *Blocks to robots: Learning with technology in the early childhood classroom.* Teachers College Press, New York.

Bers, M., U. (2012). *Designing digital experiences for positive youth development: From playpen to playground.* Oxford, Cary, NC.

Bransford, J. D., & Stein, B. S. (1993). *The ideal problem solver: A guide to improving thinking, learning, and creativity* (2nd ed.). New York: Worth Publishing.

Brennan, K., & Resnick, M. (2012). *Using artifact based interviews to study the development of computational thinking in interactive media design*. Paper presented at annual American. Educational Research Association meeting, Vancouver, BC, Canada.

British Columbia Government. (2016). $6 million to help connect students with coding, new curriculum and computers. Retrieved August 11, 2016, from https://news.gov.bc.ca/releases/2016PREM0065-000994

Carey, S., & Spelke, E. (1994). Domain-specific knowledge and conceptual change. *Mapping the mind: Domain specificity in cognition and culture*, 169-200.

Cuny, J., Snyder, L., & Wing, J.M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress. Retrieved from http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Grover, S., Bienkowski, M., & Snow, E. (2015, February). *Assessments for Computational Thinking in K-12.* In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 700-700). ACM. https://doi.org/10.3102/0013189X12463051

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM, 51*(8), 25-27. https://doi.org/10.1145/1378704.1378713

Hudkins, D. (2013). Why we must require CS education now. *Independent School, 72*(4), 76-80.

ICTC- CTIC. (2016). Digital talent road to 2020 and beyond: A national strategy to develop Canada's talent in a global digital economy. Retrieved November 8, 2016.

Kazkoff, E., Sullivan, A. & Bers, M.U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245-255. https://doi.org/10.1007/s10643-012-0554-5

Krippendorff, K. (2012). *Content analysis: An introduction to its methodology*. Sage.

Mueller, J., & Shodiev, H. (2014). Teaching and assessing computational thinking. *Final Conference Proceedings of the Society for Information Technology and Teacher Education (SITE) 26th International Conference, Las Vegas, NV.*

Mueller, J., Beckett, D., Hennessey, E. J.V., & H., Shodiev. (2017). Assessing computational thinking across the curriculum. In C. Hodges & P. Rich (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking.* AECT/Springer: Georgia USA.

Mueller, J., Wood, E., Willoughby, T., Ross, C., & Specht, J. (2008). Identifying discriminating variables between teachers who fully integrate computers and teachers with limited integration. *Computers & Education, 51*(4), 1523-1537. https://doi.org/10.1016/j.compedu.2008.02.003

Perkovic, L., Settle, A., Hwang, S., & Jones, J. (2010). A framework for computational thinking across the curriculum. *Presentation at ITiCSE'10, Bilkent, Ankara, Turkey.* https://doi.org/10.1145/1822090.1822126

Piaget, J. (1952). The origins of intelligence in children (Vol. 8, No. 5, pp. 18-1952). New York: International Universities Press.

Ruthmann, A., Heines, J.M., Greher, G.R., Laidler, P., & Saulters, C. (2010). Teaching computational thinking through musical live coding in scratch. In Proceedings of the 41st ACM technical symposium on CS education (SIGCSE '10). ACM, New York, NY, USA, 351-355. https://doi.org/10.1145/1734263.173438

Sáez López, J. M., González, M. R., & Cano, E. V. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools. *Computers & Education*, *97*, 129–141. http://doi.org/10.1016/j.compedu.2016.03.003

Seiter, L., & Foreman, B. (2013, August). *Modeling the learning progressions of computational thinking of primary grade students*. In Proceedings of the ninth annual international ACM conference on International Computing Education Research (pp. 59-66). https://doi.org/10.1145/2493394.2493403

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. University of Southampton (E-prints) 6pp. http://eprints.soton.ac.uk/id/eprint/356481

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728. https://doi.org/10.1007/s10639-015-9412-6.

Voskoglou, M. G., & Buckley, S. (2012). Problem solving and computational thinking in a learning environment. arXiv preprint arXiv:1212.0750.

Vygotsky, L. S., (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA.

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2014). *Computational thinking benefits society. Social Issues in Computing*. New York: Academic Press. Artigo disponível e consultado em: Socialissues. cs. Toronto. edu.

Wing, J., & Stanzione, D. (2016). Progress in computational thinking, and expanding the HPC community. *Communications of The ACM, 59*(7), 10-11. https://doi.org/10.1145/2933410

Ontario Elementary Curriculum Documents

Arts (2009): http://www.edu.gov.on.ca/eng/curriculum/elementary/arts18b09curr.txt

French as a Second Language (2001): http://www.edu.gov.on.ca/eng/curriculum/elementary/fsl48curr.txt

Health and Physical Education (2010): http://www.edu.gov.on.ca/eng/curriculum/elementary/healthcurr18.txt

Kindergarten (2016): Note that text file is from 2011*

https://files.ontario.ca/books/edu_the_kindergarten_program_english_aoda_web_oct7.pdf

Language (2006): http://www.edu.gov.on.ca/eng/curriculum/elementary/language18currb.txt

Mathematics (2005): http://www.edu.gov.on.ca/eng/curriculum/elementary/math18curr.txt

Native Languages (2001):

http://www.edu.gov.on.ca/eng/curriculum/elementary/nativelang18curr.txt

Science and Technology (2007): http://www.edu.gov.on.ca/eng/curriculum/elementary/scientec18currb.txt

Social Studies: Link to text file unavailable*

http://www.edu.gov.on.ca/eng/curriculum/elementary/sshg18curr2013.pdf

Growing Success: Link to text file unavailable*

http://www.edu.gov.on.ca/eng/policyfunding/growsuccess.pdf

Online Resources:

Barack Obama CS Initiative: http://1.usa.gov/21u4mxK

Partnership for 21st Century Skills: http://www.p21.org/

Python Programming Language: https://www.python.org/

The Canadian Press:
http://www.cbc.ca/news/canada/british-columbia/coding-education-school-computers-1.3631720

Notes

Note 1. http://scratched.gse.harvard.edu/ct/defining.html

Note 2. http://www.edu.gov.on.ca/eng/curriculum/elementary/subjects.html

Note 3. Latest versions of curriculum documents were converted from PDF to text format

Note 4. https://www.python.org/

Appendix A

Search Terms: Content Analysis

All Keywords and Iterations (searches were case sensitive)

1. Abstract(|ion|ing|ed|s)
2. Algorithm(s|ic)
3. Analy(s|z)e
4. Analysis
5. Appli(cation|ed|es)
6. Apply(|ing)
7. Automat(e|ion|ing|ed|es|ic|ically)
8. Cod(e|ing|ed|er|es)
9. Collect(|ion|ing|ed|or|s)
10. Comput(e|ation|ing|ed|er|es)
11. Computational(-|)thinking
12. Computational(?!(-|)thinking)
13. Conditional(|s)
14. Connect(|ing|ed|er|s)
15. Debug(|ging|ged|ger|s)

16. Decompos(e|ition|ing|ed|er|es)
17. Event(|s)
18. Experiment(|ation|ing|ed|er|s)
19. Express(|ing|ed|es)
20. General
21. Generaliz(e|ation|ing|ed|es)
22. Identification
23. Iterat(e|ion|ing|ed|or|es)
24. Logic(|al)
25. Loop(|ing|ed|s)
26. Management
27. Model(|l)(ing|ed)
28. Model(|s)
29. Modular(|ism|ize|ization|izing|ized|izes)
30. Operator(|s)
31. Parallel(|ed|s|ism|ize|ization|izing|ized|izes)
32. Problem(-|)solv(e|ing|ed|er|es)
33. Problem(|s)(?!(-|)solv(e|ing|ed|er|es))
34. Question(|ing|ed|er|s)
35. Recurs(e|ion|ing|ed|es)
36. Remix(|ing|ed|es)
37. Represent(|ation|ing|ed|s)
38. Reus(e|ing|ed|es)
39. Sequenc(e|ing|ed|er|es)
40. Simulat(e|ion|ing|ed|or|es)
41. Solve (|a|the)problem(|s)
42. Technologi(cal|es)
43. Technology
44. Test(|ing|ed|er|s)

Terms from Brennan & Resnick (2012)
Concepts

1. Conditionals
2. Data
3. Events
4. Loops
5. Operators
6. Parallelism
7. Sequences

Practices

1. Abstracting
2. Debugging
3. Incremental
4. Iterative
5. Modularizing
6. Remixing
7. Reusing
8. Testing

Perspectives

1. Connecting
2. Expressing
3. Questioning

Appendix B

Computational Thinking Definitions and Examples
Brennan & Resnick (2012)

COMPUTATIONAL THINKING
Our definition of computational thinking involves three key dimensions: (1) knowing certain computational concepts, (2) being able to employ those concepts using computational practices, and (3) developing new computational perspectives, an awareness of self, others, and world.

COMPUTATIONAL CONCEPTS
Concept & Description

1. Sequence: identifying a series of steps for a task
2. Loops: running the same sequence multiple times
3. Parallelism: making things happen at the same time
4. Events: one thing causing another thing to happen
5. Conditionals: making decisions based on conditions
6. Operators: support for mathematical and logical expressions
7. Data: storing, retrieving, and updating values

COMPUTATIONAL PRACTICES
Practice & Description

1. Experimenting and iterating: developing a little bit, then trying it out, then developing more
2. Testing and debugging: making sure things work – and finding and solving problems when they arise
3. Reusing and remixing: making something by building on existing projects or ideas
4. Abstracting and modularizing: exploring connections between the whole and the parts

COMPUTATIONAL PERSPECTIVES
Perspective & Description

1. Expressing: realizing that computation is a medium of creation, “I can create.”
2. Connecting: recognizing the power of creating with and for others, “I can do different things when I have access to others.”
3. Questioning: feeling empowered to ask questions about the world, “I can (use computation to) ask questions to make sense of (computational things in) the world.”

Appendix C

Content Analysis: Qualitative Coding Scheme (Curriculum)

1. Front matter - FM
    a. Introduction - FM-I
    b. Program - FM-P
    c. Processes (Mathematics only) - FM-Pr
    d. Assessment and evaluation - FM-AE
    e. Program planning - FM-PP
    f. Table of contents - FM-TOC
2. Curriculum expectations - CE
    a. Overview - CE-O
    b. Specific - CE-S
3. Glossary - G

Note. The above coding scheme was used in preliminary analyses. In this document, graphs show collapsed categories to provide a clearer context for interpretation of location (i.e., Curriculum Expectations [overall and specific], Front Matter, and the Glossary)