



# A scoping review of computational thinking assessments in higher education

Chang Lu<sup>1</sup> · Rob Macdonald<sup>1</sup> · Bryce Odell<sup>1</sup> · Vasyi Kokhan<sup>1</sup> ·  
Carrie Demmans Epp<sup>1</sup> · Maria Cutumisu<sup>1</sup> 

Accepted: 9 December 2021 / Published online: 23 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

The field of computational thinking (CT) is developing rapidly, reflecting its importance in the global economy. However, most empirical studies have targeted CT in K-12, thus, little attention has been paid to CT in higher education. The present scoping review identifies and summarizes existing empirical studies on CT assessments in post-secondary education, aiming to reveal the current trends of empirical research in this domain and key features of recent CT assessment instruments. It examines 33 peer-reviewed journal articles published between 2013 and 2019 from six databases. Results show that most assessment tools are designed for computing science and engineering undergraduates or pre-service and in-service teachers in these subjects. Most tools involve in-class interventions to promote CT skills. Several assessment formats were adopted in the selected studies, including selected-response questions, constructed-response questions, Likert scales, interviews, programming artefacts, observations, and interviews. Finally, most assessment instruments attempt to measure skills from a combination of dimensions including CT Concepts, Practices, and Perspectives from a hybrid-competency framework. More specifically, the skills assessed in most studies are algorithmic thinking, problem solving, data, logic and logical thinking, and abstraction. Findings may help instructors to select CT assessments for higher education and researchers to focus on less explored research areas.

**Keywords** Computational thinking · Programming and programming languages · Learning and assessment · Higher education

---

✉ Maria Cutumisu  
cutumisu@ualberta.ca

Extended author information available on the last page of the article

## Introduction

Computers and programming have revolutionized the world and have promoted technology literacy as a crucial skill to achieve academic and career success in the digital 21st century (Shute et al., 2017). Accordingly, computational thinking (CT), a complex competency defined as a way of thinking that can be applied to various fields requiring problem-solving skills, has been gaining popularity in the field of education. CT encompasses multidimensional skills related to digital literacy, such as data literacy and problem-solving. It was popularized in the early 1980s by Seymour Papert, who coined the term to encompass many skills and terms that had previously been established (Papert, 1980). Most of the original components of CT originated within the computing science (CS) field, due to the emphasis on sequential and logical analysis. Nevertheless, it has emerged as an immensely useful tool due to its employment as a way of thinking which can be applicable to broader fields beyond CS.

Since 2006, CT has undergone a powerful revival, spurred by Jeannette Wing's influential article (Wing, 2006). Wing defined CT as "an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing" (Wing, 2008: p. 1). In recent years, more definitions of CT have emerged (Brennan & Resnick, 2012; Grover & Pea, 2018; Weintrop et al., 2016), which link CT from its CS origins to a wider range of subjects, emphasizing that the CT competency is derived from but not limited to CS. Therefore, it can also be used to facilitate problem solving in various subjects such as mathematics (Bussaban & Waraporn, 2015; Buteau & Muller, 2017; Sung et al., 2017), general science problems (Bussaban & Waraporn, 2015; Flores, 2018; Sneider et al., 2014), and even the social sciences and liberal arts (Knochel & Patton, 2015).

Given the importance of CT competency, research on different aspects of CT has come under the spotlight increasingly in the past few years. In the last decade, many endeavors have been made to boost the understanding of CT as a complicated higher-order skill (Shute et al., 2017), settle a universal framework of CT constructs (Brennan & Resnick, 2012; Grover & Pea, 2018; Weintrop et al., 2016), develop curriculum for teaching CT in K-12 contexts (Hsu et al., 2018), and design platforms or instruments to assess CT skills (Csernoch et al., 2015; Dichev & Dicheva, 2017; Flanigan et al., 2017; Kordaki, 2013; Korkmaz, 2012; Sheehan et al., 2019; Sherman & Martin, 2015; Topalli & Cagiltay, 2018). Additionally, CT has been included in the next iteration of the PISA large-scale assessment (OECD, 2018). However, most of the recent research on CT has focused on the K-12 education context, with only a few studies addressing research in post-secondary education and fewer focusing on CT assessments (Cutumisu et al., 2019).

The present study has identified a gap in the existing literature on CT assessments. Specifically, there is a dearth of CT assessments in higher education, even though there is an acute need for CT assessments at this level to ensure that students are prepared for the digital workforce of the 21st century. Several factors have encumbered the prevalence of CT assessments in post-secondary education. For example, attempts to incorporate CT principles into non-STEM (Science,

Technology, Engineering, and Mathematics) fields without appropriate training provided to students and instructors might lead to frustration, lowered self-efficacy, and lessened motivation towards learning (Miller et al., 2013). In addition, to date, there is no universal assessment tool nor definitive method to select assessments for CT in higher education (Cutumisu et al., 2019). It is increasingly apparent that CT skills, as a vital component of 21st century competencies, are not only useful in computer science or engineering programs in post-secondary institutions but also applicable to a wider variety of fields to facilitate future success on tasks requiring problem-solving skills. Therefore, much more research is needed on this topic. The present study explores aspects of CT assessments in post-secondary education. More specifically, we attempt to answer the following questions:

1. *Who are the audiences for CT assessment instruments?*
2. *Are the assessments embedded in courses or programs?*
3. *What are the key features of the assessment instruments?*
4. *What measurement instruments does each assessment adopt?*
5. *What CT skills does each measurement instrument assess?*

## Theoretical framework

Wing (2006) asserted that CT “involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p. 33). However, CT is decoupled from ‘computing science’ and, instead, it represents ‘a model of thinking’ that is fundamental for everyone and applicable to a broad range of domains, including mathematics, engineering, and biology, as well as to daily life scenarios concerned with problem solving (Angeli & Giannakos, 2020; Li et al., 2020). To accurately discuss a topic as highly nuanced as CT, it is necessary to address and integrate the various theoretical frameworks within CT.

First, Brennan and Resnick’s (2012) three-dimensional framework is widely acknowledged, and it includes CT concepts, practices, and perspectives. Within each, there are various sub-components that describe them in more detail. More specifically, CT concepts are the essential ideas utilized to describe CT investigations, such as: sequence—a structure of series of actions in a predetermined order; loop—a sequence of instructions that is iterated until the specific condition is met; events—occurrences detected by a program which may trigger the execution of subsequent steps; parallelism—multiple events that perform concurrently; conditionals—programmer-specified rules for execution of dependent instructions; operators—constructs that behave like functions to manipulate data but are expressed semantically; data—information processed or stored digitally, can be represented by binary data, numbers, text, images, audio, videos, programs, or other types of data. In comparison to CT concepts, which are quite theoretical, CT practices are very pragmatic. They specify how CT concepts come to life and how they are manipulated. CT practices include how to interact with colleagues in a project, how to apply principles

of CT, how to overcome problems, the processes that enable us to learn new information, the ways we approach learning iteratively and methodically, testing and debugging to improve parameters of a current project, reusing and remixing to use solutions to current problems which have already been addressed by colleagues, and abstracting and modularizing to convert a practical problem by reframing it in CT terms. Finally, CT perspectives focus on understanding the bigger picture surrounding the dynamics of people, places, and things. This includes such terms as expressing—bring personal aspects of one's unique self-expression into a project; connecting—coordinate one's activities in a project with others to foster optimal outcomes, and questioning—examine current conventions or ways of doing things and explore what methods can be used to reach towards higher ideals, increased performance, or more optimal outcomes.

Later, Weintrop et al. (2016) proposed a taxonomy of CT in mathematics and science including four main dimensions. First, *data practices* include collecting, creating, manipulating, analyzing, and visualizing data. Second, *modeling and simulation practices* include (1) using computational models to understand a concept and to find and test solutions; and (2) assessing, designing, and constructing computational models. Third, *computational problem solving practices* include preparing problems for computational solutions, programming, choosing effective computational tools, assessing different approaches/solutions to a problem, developing modular computational solutions, creating computational abstractions, and troubleshooting and debugging. Fourth, *systems thinking practices* consists of five subskills including investigating a complex system as a whole, understanding the relationship within a system, thinking in levels, communicating information about a system, and defining systems and managing complexity (Weintrop et al., 2016). Several aspects of the framework proposed by Weintrop et al. (2016) are consistent with Brennan and Resnick's (2012) work, such as modeling and simulation as well as testing and debugging, but the former is more directly aligned with mathematics and science learning, aiming to deepen the pedagogical practices of the two subjects.

A more recent and well-rounded CT competency framework was proposed for K-12 STEM education by Grover and Pea (2018). Two main dimensions are described in this framework: (1) *CT concepts* include logic and logical thinking, algorithms and algorithmic thinking, patterns and pattern recognition, abstraction and generalization, and evaluation and automation; and (2) *CT practices* include problem decomposition, creating computational artefacts, testing and debugging, iterative refinement (incremental development), and collaboration and creativity. In contrast with Brennan and Resnick's (2012) framework, Grover and Pea's framework moves abstraction and generalization into the CT Concept Category, and CT Perspectives into the CT Practices dimension. The new framework attempts to shift the core skills of CT from computing literacy towards a more problem-solving competency that transcends the computing science domain.

Although there is no universally-accepted CT definition or taxonomy, CT is generally viewed as a complex problem-solving competency, which requires both *cognitive skills* such as abstraction, problem decomposition, planning, testing and debugging, as well as *non-cognitive skills* such as creativity, communication, and collaboration. Moreover, CT is not limited to a single academic discipline and it can

facilitate deeper learning in various domains. The underlying subskills of CT are flexible and may be adjusted to adapt to different applications. However, this makes it more difficult to assess CT competency using a single instrument. The current study adopts a hybrid framework drawing from Weintrop et al. (2016) framework of CT for mathematics and science classrooms, Grover and Pea's (2018) two-dimensional framework, and Brennan and Resnick's (2012) three-dimensional framework to extensively examine and evaluate CT assessments in the contexts of post-secondary education. The hybrid, umbrella framework enables us to cover more dimensions and skills of CT. Moreover, it is more generic and independent of specific subjects. Therefore, it is more feasible to incorporate the framework into diverse programs of studies.

## Literature review

Previous studies have made efforts to systematically group the existing studies related to computational thinking in different contexts. Lye and Koh (2014) examined 27 empirical studies on teaching and learning of computational thinking through programming in K-12 settings. Their findings show that K-12 students often learn CT by using visual-programming tools to create artefacts like digital stories or games. The authors pointed out that although most research studies have yielded knowledge gains from CT interventions, they have mainly focused on CT concepts, with only a few studies measuring students' CT practices or perspectives. Lye and Koh (2014) recommended the incorporation of CT practices and perspectives into a constructionism-based problem-solving classroom that facilitates deeper information processing, scaffolding, and reflection activities.

Shute et al. (2017) summarized 45 theoretical and empirical articles to examine the emerging field of computational thinking through a systematic review of its various definitions, main components, interventions, models, and assessments in K-16 settings. They concluded that CT is an ambiguous yet compound competency that is hard to differentiate from similar terms such as algorithmic thinking or mathematical thinking, because CT encompasses many overlapping skills, such as problem solving, modeling, and data analysis. The authors reviewed related articles published from 2006 to 2017, summarizing the main characteristics and most-discussed CT topics. They devised a comprehensive definition to demystify CT mainly as a "logical way of thinking, not simply knowing a programming language" (Shute et al., 2017: p. 16), highlighting the main components of CT, which include problem decomposition, abstraction, algorithmic design, debugging, iteration, and generalization. More importantly, they suggested CT is not strictly tied to a particular discipline, but rather a problem-solving competency that can be taught and embedded into multiple disciplines. They also argued that the key to successfully integrate CT into existing disciplines is to implement a "reliable and valid CT assessment" (Shute et al., 2017: p. 15). However, considering the multitude of CT definitions and models, assessing CT becomes a major weakness. Specifically, there is no widely-adopted valid instrument nor gold standard for current empirical research.

Hsu et al. (2018) also conducted a meta-review of the studies related to learning and teaching CT in K-12 settings published from 2006 to 2017. Through analyzing the courses, participants, teaching tools, and programming languages used in the selected studies, the authors proposed five suggestions for future research on CT. Two of them are concerned with effectively assessing CT among students who employ various learning strategies in different subjects as well as accurately detecting students' learning status so that instructors could provide feedback.

More recently, Zhang and Nouri (2019) conducted a more focused systematic review of empirical studies on learning CT in K-9 through Scratch. They examined 55 studies to identify all CT skills learned and assessed for K-9 students through the visual-programming language Scratch, drawing on Brennan and Resnick's (2012) three-dimension framework. Their results show that CT skills proposed by Brennan and Resnick's framework (2012) can be effectively delivered and learned through the use of Scratch. Moreover, additional CT skills were identified beyond the scope of Brennan and Resnick's framework through Scratch, including reading, interpreting, communicating code, as well as employing multimodal design, predictive thinking, and human-computer interaction. Furthermore, Zhang and Nouri (2019) discussed the urgent needs as well as challenges associated with assessing the learning outcomes of CT learning. They also recognized the importance of monitoring students' learning progression throughout the CT interventions. Similar to previous reviews, they found that most studies focused on CT concepts, with only limited studies assessing CT practices and perspectives. They suggested that CT concepts were more amenable to a wide variety of assessment forms, such as selected-response items (e.g., multiple-choice questions) or programming artefacts; while CT perspectives were difficult to evaluate directly. Thus, many studies chose the convenient way to assess CT skills by leaving out CT perspectives or even some CT practices from Brennan and Resnick's (2012) three-dimensional framework.

Cutumisu et al. (2019) conducted a scoping review of 39 empirical studies on assessments of computational thinking published from 2013 to 2018. This review examined several aspects of CT assessments, including forms of assessment, skills assessed, measures, instrument reliability and validity, courses in which CT was embedded, and student demographics. Cutumisu et al. (2019) found that most assessment instruments measure CT concepts and practices rather than CT perspectives, and that algorithmic thinking, abstraction, problem decomposition, and logical thinking are the most assessed skills. In addition, the results showed that most CT assessments focused on K-12 settings, with only a few focusing on higher education. Meanwhile, Angeli and Giannakos (2020) summarized the achievements of current research on CT education and discussed five main issues and challenges of future research on computational thinking education. Among the five future directions, two of them are about CT in higher education and CT assessments. More specifically, one suggestion is concerned with preparing pre-service as well as in-service teachers with CT literacy so that they can develop knowledge on how to teach and assess CT with confidence and to guide their K-12 students in acquiring CT knowledge as well. Systematic CT professional development is necessary to prepare and equip in-service and pre-service teachers with CT skills and to help them design effective learning activities. The other suggestion highlights another area that is

underdeveloped in CT, namely the identification of the most effective assessment tools to measure students' CT skills.

Some common gaps emerge from the reviews discussed above. First, most empirical studies focus on the K-12 context, while few have examined CT in higher education or among pre-service or in-service teachers. Second, the field of CT assessment is still under rapid development, lacking consensus regarding the identification of a validated instrument for a specific grade level. Finally, few studies have discussed the optimal ways to embed CT activities into an existing curriculum. Therefore, this study sets out to fill this gap by summarizing recent empirical studies on assessing CT skills in the context of post-secondary education.

## Method

### Purpose

According to Tang et al. (2020), recent research on CT can be separated into two time periods: 2006–2012 and 2013–2019. The 2006 resurgence of CT was followed by a slow growth between 2006 and 2013. After 2013, there was a marked boost in the average number of yearly publications on CT. Moreover, Cutumisu et al. (2019) indicated that CT has not been fully explored and the work on CT assessments is still lagging behind the rest of the research on CT. Particularly, our preliminary search has identified a gap in the CT assessment literature in higher education. Thus, we conducted a scoping review to examine the research studies on CT assessments in post-secondary education published between January 2013 and June 2019.

This review is informed by Arksey and O'Malley's (2005) five-stage approach to conduct a scoping review: (1) identifying the research questions; (2) identifying relevant studies; (3) selecting studies; (4) charting the data; and (5) collating, summarizing, and reporting the results.

### Identifying the research questions

The key purpose of this study is to thoroughly identify and analyze computational thinking assessment tools for post-secondary education in recent empirical studies. Therefore, this review is guided by the following research questions:

1. *Who are the audiences for CT assessment instruments?* What are the participants' countries and current educational levels, and what are the corresponding assessment tools employed? What are their current programs of study?
2. *Are the assessments embedded in courses or programs?* On what subjects/courses are the assessments based?
3. *What are the key features of the assessment instruments?* What forms does each assessment instrument adopt: automatic or non-automatic (i.e., manual) scoring, cognitive or non-cognitive, paper-based or computer-based?

4. *What measurement instruments does each assessment adopt? What measures does each tool adopt to assess CT competency?*
5. *What CT skills does each measurement instrument assess? Does the instrument assess all three dimensions of CT? What subskills are assessed in each assessment instrument?*

## Keyword search

We chose the following six databases that cover popular journals of interest in computer science and education: SpringerLink, IEEE Xplore, SCOPUS, ACM Digital Library, ScienceDirect, and ERIC. The keywords employed were “computational thinking” AND (“assessment” OR “measurement”). Results were restricted to peer-reviewed journal articles published in the last six and half years, from January 2013 to June 2019. The search results can be found in Fig. 1. The initial search returned 356 results, whose distribution by database is presented in Table 1.

## Selecting studies

Table 2 presents the result screening process. Three rounds of article inclusion and exclusion were conducted. In the first round, duplicates and articles that did not

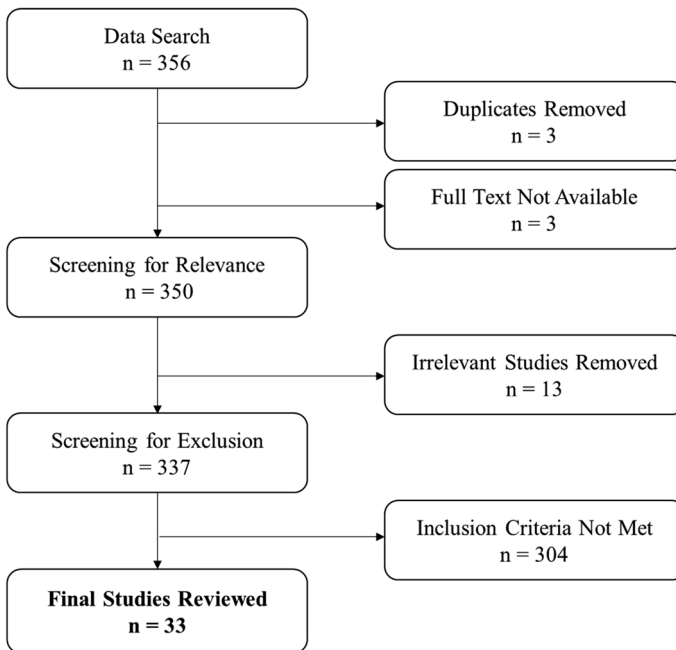


Fig. 1 Flow diagram of the article search and selection process



**Table 1** Initial search results distributed by database

Database	SpringerLink	IEEE Xplore	SCOPUS	ACM digital library	ScienceDirect	Eric	Total
N of papers	197	20	14	13	98	44	356

**Table 2** Article screening process

Screening for exclusion	Total
Not empirical	67
Duplicates	3
Not related to education or assessment	13
Involved education or assessment but no integration of computational thinking	77
Involved computational thinking but the assessments used did not assess CT skills	44
Not for higher education	116
Unavailable	3
Total after exclusion	33

contain full texts were removed from the initial search results; in the second round, four coders screened articles collected from one or two of the databases to exclude irrelevant articles and articles that did not meet the inclusion criteria. The inclusion and exclusion criteria can be found as follows:

1. Not empirical or the full text is not available;
2. Not related to education or assessment;
3. Involved education or assessment but did not integrate CT;
4. Involved CT but did not assess CT;
5. The assessment is not designed for post-secondary education.

In the final round, a fifth researcher verified that all selected articles meet the requirements and resolved any disputes that had occurred in the article inclusion process. After inclusion and exclusion of the articles, 33 articles were retained through three rounds of filtering.

## Results

Table 3 summarizes the assessment instruments used in each article. Most assessment tools are designed for undergraduate students with a Computer Science (n=11) major, followed by pre-service teachers in education (n=7), in-service teachers in STEM subjects (n=6), and undergraduate students in STEM related majors (n=3). Tools such as the Computational Thinking Scale (CTS) (Doleck et al., 2017; Korkmaz, 2012; Korkmaz et al., 2017) and the Computational Thinking

**Table 3** The details of the articles reviewed in the current study: year, authors, CT assessment instrument, programs of studies, and publication venue

Year	Author(s)	Assessment instrument	Programs of studies	Journal name
[1]	2017 Buteau and Muller (2017)	MICA assessment	Math	Digit Exp Math Educ
[2]	2019 Dağ (2019)	Multiple tools: General self-efficacy scale; course evaluation questionnaire; semi-structured electronic form; course academic achievement; Small Basic (code-based)	Computer majors in Education	J. Comput. Educ
[3]	2017 Doleck et al. (2017)	Computational thinking scale (CTS) and GPA	Science	J. Comput. Educ
[4]	2019 Itiguez-Berrozpe and Boeren (2019)	OECD's PIAAC online survey	Not specified	Technology, Knowledge and Learning
[5]	2017 Jaipal-Jamani and Angeli (2017)	Robotics/Game-Based Assessment; pre- and post- surveys	Elementary education	J. Sci Educ Technology
[6]	2019 Lazarinis et al., (2019)	Online Moodle course with Scratch and quizzes	Not specified	Education and Information Technologies Journal
[7]	2019 Lee and Cho (2020)	Light-Bot Quizzes	Not specified	Information Technology and Management
[8]	2015 Mishra and Iyer (2015)	Quizzes on knowledge of CT	Data structures and artificial intelligence	Research and Practice in Technology Enhanced Learning
[9]	2018 Peteranetz et al. (2018)	CT knowledge test	Engineering	IEEE Transactions on Education
[10]	2014 Ricketts (2014)	NRC's Framework Assessment (National Research Council)	Elementary education	Sci & Educ
[11]	2017 Romero et al. (2017)	#5c21	Elementary education	International Journal of Educational Technology in Higher Education
[12]	2017 Sentance and Cszizmadia (2017)	CAS Online Survey	Not specified	Educ Inf Technology
[13]	2013 Shell and Soh (2013)	CT, knowledge test; Student Perceptions of Classroom Knowledge Building (SPOCK) test	Computer Science (CS), Business/CS honors, engineering majors, combined CS/Engineering/Science	J Sci Educ Technology
[14]	2016 Tuttle et al. (2016)	Concept mapping, Lesson plans & Video recording of lesson delivery	Not specified	J Sci Teacher Education

Table 3 (continued)

Year	Author(s)	Assessment instrument	Programs of studies	Journal name
[15]	2015 Csernoch et al. (2015)	Testing Algorithmic and Application Skills (TAaAS) project & CAAD (Computer Algorithmic and Debugging) Test	Software Engineering, System Engineering and Business Information Management	Informatics in Education
[16]	2015 Sherman and Martin (2015)	Mobile Computational Thinking (MCT) rubric and Generic CT, App Inventor	9 CS, 9 non-CS	Journal of Computing Sciences in Colleges
[17]	2014 Berkaliev et al. (2014)	Programmatic Assessment Report	Math	PRIMUS
[18]	2013 Libeskind-Hadas and Bush (2013)	Projects in "Genetic algorithms for robot control"	Biology	Briefings in Bioinformatics
[19]	2014 Yadav et al. (2014)	Computational-Thinking Quiz & Computing Attitude Questionnaire	Education	ACM Transactions on Computing Education (TOCE)
[20]	2017 Korkmaz et al. (2017)	Computational thinking scale (CTS)	Not specified	Computers in Human Behavior
[21]	2017 Flanigan et al. (2017)	CT knowledge test; Implicit Theories of Intelligence Scale; profile measures; Course achievement measures	CS	Contemporary Educational Psychology
[22]	2017 Dichev and Dicheva (2017)	Pre- and post-course tests	CS, Information Technology, Business Administration, and Psychology	Procedia Computer Science
[23]	2018 Topalli and Cagiltay (2018)	Scratch & GPA	Engineering	Computers and Education
[24]	2019 Sheehan et al. (2019)	ScratchJr	Not specified	Computers and Education
[25]	2013 Kim et al. (2013)	The Group Assessment of Logical Thinking (GALT) Test	Elementary education	Journal of Educational Computing Research
[26]	2013 Kordaki (2013)	Structured interview, observations	Not specified	Computers and Education
[27]	2013 Robertson et al. (2013)	Attitude and Computational Thinking Quiz	Not specified	International Journal of Child-Computer Interaction
[28]	2014 Magana (2014)	FS2C framework assessment	CS	Computers and Education

**Table 3** (continued)

Year	Author(s)	Assessment instrument	Programs of studies	Journal name
[29]	2015 Biró et al. (2015)	Testing Algorithmic and Application Skills (TAaAS) project	Software Engineering (SOE), System Engineering (SYE), and Business Information Management (BIM)	Procedia—Social and Behavioral Sciences
[30]	2015 Figl and Laue (2015)	Questionnaire: theoretical knowledge test of process modeling	Business	Int. J. Human–Computer Studies
[31]	2015 Israel et al. (2015)	Etoys and Scratch	Not specified	Computers and Education
[32]	2016 Chao (2016)	Visual programming projects	Information communication	Computers and Education
[33]	2016 Magana et al. (2016)	Control-Value Theory of Achievement Emotions Self-belief measure and academic performance	Engineering	Computers in Human Behavior

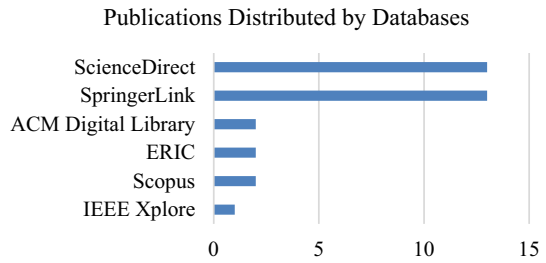
knowledge test (Flanigan et al., 2017; Peteranetz et al., 2018) were used in multiple studies to assess students' CT perceptions and skills. However, most studies implemented their own instruments together with interventions to assess CT growth outcomes. For example, Dağ (2019) used a set of assessment tools including a general self-efficacy scale, a course evaluation questionnaire, a semi-structured electronic form containing five questions on teaching programming and self-development, course academic achievement, and Small Basic tasks to evaluate pre-service teachers' use of CT. Mishra and Iyer (2015) taught C++ and Scratch in two CS courses for four weeks and quizzed students to assess their knowledge of CT. Sherman and Martin (2015) created a rubric for grading students' CT assignments on app design for mobile devices, such as smartphones and tablets in App Inventor. The rubric, named 'mobile computational thinking' (MCT), measures 14 specific mobile app dimensions that are related to general CT and mobile CT, including screen interfaces and location awareness. Apart from assessment tools specifically designed for a course or a study, several research studies adapted existing large-scale assessments to evaluate participants' CT competency, such as the OECD's PIAAC online survey (Iñiguez-Berrozpe & Boeren, 2019) and the Group Assessment of Logical Thinking (GALT) Test (Kim et al., 2013). Other studies opted for developing assessment tools that are aligned with an existing framework or curriculum, including the NRC's Framework Assessment (National Research Council) to assess pre-service teachers' thoughts on scientific practices in K-12 education (Ricketts, 2014) and the FS2C framework assessment (Magana, 2014) to inform the design or selection of multimedia for learning tools. Some studies simply used course academic achievements (programming assignments, final exam scores, and GPA) or in-class pre-tests and post-tests to represent students' CT growth, as the interventions and assessments were embedded in the course syllabus (Dichev & Dicheva, 2017; Doleck et al., 2017; Jaipal-Jamani & Angeli, 2017; Topalli & Cagiltay, 2018). As in the case of K-12 CT assessments, there is no universal CT tool for assessments in post-secondary settings. Previous reviews emphasize a trend of adopting multiple tools to assess more dimensions of sophisticated CT skills, compared with CT assessments in K-12 settings (Cutumisu et al., 2019). Similarly, the present review revealed that most studies also combined multiple tools to measure different CT dimensions. The current study identified a total of 30 unique assessment tools (see in Table 3) in the 33 selected articles.

## Characteristics of empirical studies on CT assessments in higher education

### Distribution of articles on CT assessments by database

We conducted our literature search in June 2019. First, we examined the articles retrieved to ensure they met our inclusion criteria. We retained 33 CT assessment-related studies. Figure 2 displays the distribution of the included articles across the searched databases. The results revealed that ScienceDirect and SpringerLink returned the most related studies (13 articles each), followed by the ACM Digital

**Fig. 2** Selected publications distributed by databases



Library, ERIC, and Scopus, that yielded two articles each, whereas IEEE Xplore only contributed one related study.

### Distribution of articles on CT assessments over time

The current review focused on studies performed in the past 6.5 years. Over that time, we noticed a fluctuation in the number of published studies that met inclusion criteria, with the highest number of studies being published in 2017 ( $n=8$ ) and the lowest number of studies published in 2018 ( $n=2$ ). Although we have not included the articles published since June 2019 when this literature search was concluded, there were only 5 other studies published until the end of 2019 (Fig. 3).

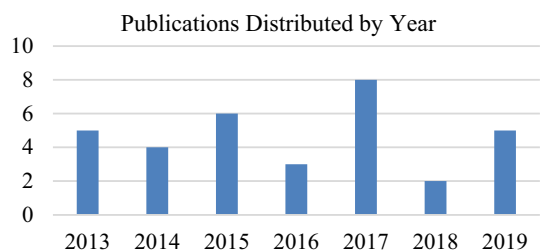
### Students' countries featured in CT assessment studies

The countries represented in the search results in Fig. 4 include a majority from North America, of which the US contributed the most ( $n=14$ ), followed by Canada ( $n=4$ ). Some studies came from European countries such as Turkey ( $n=3$ ), the UK ( $n=2$ ), Hungary ( $n=2$ ), Greece ( $n=2$ ), and Germany ( $n=1$ ). Asian countries such as India ( $n=1$ ), Taiwan ( $n=1$ ), and South Korea ( $n=2$ ) also contributed to the search results, however far fewer than those from Western countries.

### Participants' grade levels and programs of studies

One of the main goals of the present literature review is to identify who the audiences of the CT assessment tools are and their background to better inform future implementation or selection of appropriate instruments. The number of participants

**Fig. 3** Publications distributed by year



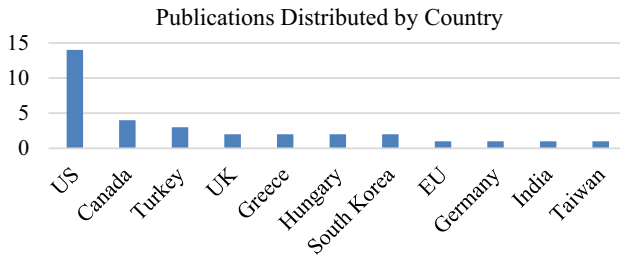


Fig. 4 Publication distributed by country

sampled, their countries, grade levels, and programs of studies are presented in Table 4.

Figure 5 shows that most studies employed the assessment tools with populations that included undergraduate students majoring in Computer Science ( $n=11$ ), followed by pre-service teachers (i.e., undergraduate students majoring in Elementary Education,  $n=7$ ); in-service teachers in subjects such as science, math, or physics ( $n=6$ ); undergraduate math majors ( $n=2$ ); and undergraduate biology majors ( $n=1$ ). Among the 33 articles reviewed, 3 studies did not explicitly describe the participants' programs of study. With regards to the sample size, results shown in Fig. 6 revealed a great variation in recruitment, with most assessment studies ( $n=15$ ) sampling 100–200 participants. The results show that most empirical studies of CT assessment in post-secondary institutions have fairly large sample sizes, which enables researchers to validate the assessment tool with more solid evidence.

### Classification of CT assessments

The 33 studies selected can be categorized into several main types: block-based assessments, knowledge/skill tests, self-reported Likert scales, text-based programming projects, academic achievements of CS courses, as well as interviews and observations. This section will discuss the details of these different types of assessments.

**Block-based assessments** Some popular block-based programming languages including *Scratch*, *Scratch Jr.*, *Alice*, *Light Bot*, *Small Basic*, and *Etoys* were widely used to promote participants' CT skills and adapted to assess the outcomes of the interventions across K-12 and post-secondary contexts.

Scratch is a visual block-based programming language designed by the MIT Media Lab. It is an effective tool to introduce basic computational concepts and practices to beginners; it is the most popular tool to promote CT skills in the empirical research across K-12 grade levels (Cutumisu et al., 2019). In the context of post-secondary institutions, Scratch is also used in some studies to teach and assess computational thinking among first-year CS students as well as non-CS students. Israel et al. (2015) recruited  $n=9$  primary school teachers and conducted interviews and observations to investigate how easily they can integrate *Etoys* and *Scratch* into

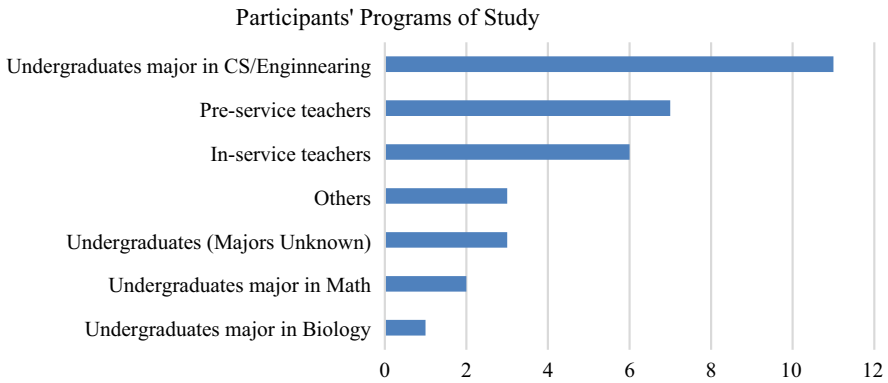
**Table 4** Participants' background information in the selected studies

Article	N	Country	Programs of studies	Grade levels
[1] Buteau and Muller (2017)	158	Canada	Math	1st and 2nd year undergraduates
[2] Dağ (2019)	26	Turkey	Computer majors in education	4th year undergraduates
[3] Doleck et al. (2017)	104	Canada	Science	Pre-university students
[4] Iniguez-Berrozpe and Boeren (2019)	20,034	EU	Not specified	Working age adults
[5] Jaipal-Jamani and Angeli (2017)	21	Canada	Elementary education	Pre-service teachers
[6] Lazarinis et al. (2019)	559	Greece	Not Specified	In-service teachers
[7] Lee and Cho (2019)	157	South Korea	Not Specified	Undergraduates
[8] Mishra and Iyer (2015)	104	India	Data structures and artificial intelligence	Undergraduates
[9] Peteranetz et al. (2018)	110	US	Engineering	1st and 2nd year undergraduates
[10] Ricketts (2014)	18	US	Elementary education	Senior pre-service teachers
[11] Romero et al. (2017)	120	Canada	Elementary education	3rd year undergraduates (pre-service teachers)
[12] Sentance and Csizmadia (2017)	339	UK	Not Specified	In-service teachers
[13] Shell and Soh (2013)	233	US	Computer Science (CS), Business/CS honors, engineering majors, combined CS/Engineering/Science	Undergraduates
[14] Tuttle et al. (2016)	11	US	Not Specified	In-service teachers
[15] Csernoch et al. (2015)	950	Hungary	Software Engineering, System Engineering and Business Information Management	1st undergraduates
[16] Sherman and Martin (2015)	18	US	9 CS, 9 non-CS	Undergraduates
[17] Berkaliev et al. (2014)	70	US	Math	Undergraduates
[18] Libeskind-Hadas and Bush (2013)	31	US	Biology	1st undergraduates
[19] Yadav et al. (2014)	357	US	Education	Pre-service teachers
[20] Korkmaz et al. (2017)	726	Turkey	Not Specified	Associate and undergraduate students, and distance learning student
[21] Flannigan et al. (2017)	443	US	CS	Undergraduates
[22] Dichev and Dicheva (2017)	18	US	CS, Information Technology, Business Administration, and Psychology	Undergraduates



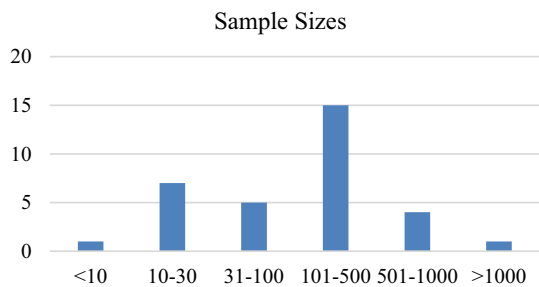
Table 4 (continued)

Article	N	Country	Programs of studies	Grade levels
[23] Topalli and Cagiltay (2018)	48	Turkey	Engineering	Undergraduates
[24] Sheehan et al., (2019)	31	US	Not Specified	Parents and 4.5–5 year-olds
[25] Kim et al. (2013)	132	South Korea	Elementary education	2nd year pre-service teachers
[26] Kordaki (2013)	25	Greece	Not Specified	Public high-school computing teachers
[27] Robertson et al. (2013)	31	UK	Not Specified	Primary and middle school teachers and students
[28] Magana (2014)	224	USA	CS	Undergraduates
[29] Biró et al. (2015)	950	Hungary	Software Engineering (SOE), System Engineering (SYE), and Business Information Management (BIM)	1st year undergraduates
[30] Figl and Laue (2015)	199	Germany	Business	Undergraduates
[31] Israel et al. (2015)	9	US	Not Specified	Teachers and administrators
[32] Chao (2016)	158	Taiwan	Information communication	Undergraduates
[33] Magana et al. (2016)	130	USA	Engineering	Undergraduates



**Fig. 5** Participants' programs of studies in the selected articles

**Fig. 6** Sample sizes in the reviewed studies



K-12 CS teaching and learning, and how well the teachers understand computing concepts and perspectives. Romero et al. (2017) examined 120 Canadian undergraduate students' abilities to utilize CT principles for creative programming tasks in *Scratch*. Topalli and Cagiltay (2018) sampled 48 engineering students from an introductory programming course to investigate whether introducing real-life problem-based game development with *Scratch* programming could enhance classical introductory programming. *Scratch Jr.* is also a block-based programming language designed at the MIT Media Lab but aimed for lower age groups. Lazarinis et al. (2019) examined the Greek K-12 teachers' programming abilities and CT concepts through an online course with activities on *Scratch*, and assessed participants with *Scratch* tasks and quizzes. Sheehan et al. (2019) observed 31 dyads of parents and their 4.5- to 5.0-year old children playing in *Scratch Jr.* together to examine their spatial talk, question-asking, task-relevant talk, and responsiveness. Most previous studies demonstrated the potential of block-based programming contexts for teaching and promoting CT.

Alice is another widely-used visual programming environment developed at Carnegie Mellon University, which provides a platform for students to create 3D virtual worlds or digital story-telling. Dağ (2019) examined the ability of a course to educate Turkish pre-service computer teachers in programming principles which target the use of CT. Participants' were Faculty of Education students

in their last year of their computer major. Perceived knowledge was assessed through projects done in *Scratch*, *Small Basic*, and story design in *Alice 3*.

*Light Bot* (Lee & Cho, 2020), a programming puzzle game based on programming concepts, *Small Basic* (Dağ, 2019), a programming language designed for transitioning students from block-based to text-based coding, and *Etoys* (Israel et al., 2015), a visual programming platform to foster students' computational thinking, are programming platforms that were created and used in the selected studies. The tools are usually integrated into traditional classrooms to boost students' CT skills or raise awareness among pre-service teachers or in-service teachers' regarding the incorporation of CT into classic computing or science-related subjects.

**Text-based programming assessments** Since the present review focuses on using CT assessments at the post-secondary level or with working adults, some studies adopted text-based programming tasks to assess students' CT competency (Csernoch et al., 2015; Lee & Cho, 2020; Libeskind-Hadas & Bush, 2013). Kim et al. (2013) examined 143 South Korean undergraduate students' academic performance in comparison to their CT abilities during a computer programming course in Python. Light Bot as well as quizzes, exams, and homework in the courses were used for their ability to assess the pre and post CT abilities of students. Csernoch et al. (2015) tested 950 university students in traditional and non-traditional programming environments to see if the knowledge matches the final exams of CS courses and to ascertain knowledge transfer from secondary to tertiary education.

**CT skill written tests** Some studies utilized more generic forms of assessment such as multiple-choice questions or constructed response questions to assess CT skills (Figl & Laue, 2015; Flanigan et al., 2017; Robertson et al., 2013). The *Computational Thinking knowledge test* (CT knowledge test) is a validated Web-based knowledge test designed at the University of Nebraska-Lincoln (Peteranetz et al., 2018; Shell & Soh, 2013). This CT knowledge test initially contained 26 items and was later reduced to 13 items through several rounds of validation. The test targets students in higher education and covers a combination of questions regarding CT concepts and the application of problem-solving.

Adaptation or directly borrowing existing validated assessments is another method for developing CT assessment tools for post-secondary levels. Iñiguez-Berrozpe and Boeren (2019) assessed working adults' skills of Problem Solving in Technology Rich Environments (PS-TRE) using the *Programme for the International Assessment of Adult Competencies (PIAAC)*, which is an online survey that includes 72 items related to PS-TRE numeracy and 72 items related to PS-TRE literacy. More specifically, the survey includes three main sections: use of PS-TRE skills at home, use of PS-TRE skills at work, and proficiency of PS-TRE skills. Kim et al. (2013) used the paper-and-pencil programming strategy (PPS) to improve 132 non-computer majors pre-service elementary school teachers' understanding and use of computational thinking and adopted the Group Assessment of Logical Thinking (GALT) pre- and post-tests to measure the effectiveness of PPS.

**Self-reported scales/survey** CT perspectives are mainly concerned with inter- and intra-personal skills including communication, collaboration, or questioning, which are difficult to measure through programming-based or written tests. Some studies adopted self-reported Likert scales to assess students' self-efficacy or confidence on CT (Korkmaz et al., 2017; Sentance & Csizmadia, 2017).

*Computational thinking scales (CTS)* is a 5-point self-report Likert-scale that consists of 29 CT items measuring five main factors including communication, critical thinking, problem solving, creative thinking, and algorithmic thinking. It is adapted from the *How Creative Are You* questionnaire (Whetton & Cameron, 2002), the *Problem Solving Scale* (Heppner & Petersen, 1982), the *Cooperative Learning Attitude Scale* (Korkmaz, 2012), the *Scale of California Critical Thinking Tendency*, and the *Logical-Mathematical Thinking* scale (Yesil & Korkmaz, 2010). CTS is a validated tool that was used in two of the studies from the selected articles (Doleck et al., 2017; Korkmaz et al., 2017).

Some studies implemented their own questionnaire along with CT skill test to measure participants' attitudes, confidence, or awareness. Dağ (2019) asked students to complete a self-efficacy scale (17 items on a 5-point Likert scale), a course evaluation questionnaire (8 items on a 7-point Likert scale), and a semi-structured electronic form (5 questions relating to computer programming instruction and personal development, feelings about the course, and feelings about programming tools), in addition to using course academic achievement to comprehensively evaluate all three dimensions of CT. Shell and Soh (2013) employed a web-based survey to evaluate students' self-regulation, Student Perceptions of Classroom Knowledge Building (SPOCK), and motivation. The SPOCK test is comprised of 9 items on self-regulated strategy use, 10 items on knowledge building, 4 items on low-level question asking, 5 items on high-level question asking, and 10 items on lack of regulation; and the motivation scale contains 24 items on class goal orientation (learning/performance/task approach & avoid), 11 items on future time perspective (FTP) measured by questions regarding an individual's perception of their remaining time in life, and 20 items on course affect (Shell & Soh, 2013).

## Interviews and observations

In studies where the main goal is to observe pre-service or in-service teachers' practices of incorporating CT into traditional classrooms, interviews and observations are commonly-used methods. Kordaki (2013) conducted structured interviews and in-class observations to investigate the beliefs of 25 public High-School Computing (HSC) teachers about their motivation, self-efficacy, and self-expectations as Computing teachers. Teachers were also interviewed regarding the nature of HSC and its curricula, how their students could be better learners in Computing, the expectations they have of their students, their own teaching approaches, and the alternatives they propose to best teach HSC. Finally, they were asked about possible associations between their beliefs and teaching practices. Tuttle et al. (2016) examined the ability of a 2-week summer class to educate Pre-K to Grade 3 teachers in scientific inquiry and engineering design principles, which included using mathematics and

CT. The researchers administered pre- and post- concept-map grading, lesson plans, and video recordings of lesson delivery.

**Course Academic Achievemen** Following CT educational interventions, a commonly used method for CT assessment was academic performance in post-secondary coursework, based on a student's achievement during quizzes, exams, programming assignments, and class projects. One of the studies used academic performance as the only assessment method for CT (Topalli & Cagiltay, 2018). Some studies used course academic performance as a complementary measure. Dağ (2019) assessed academic performance based on assignments performed in *Scratch* and *Small Basic*, midterm exams, and final assessments of digital story design presentation in *Alice 3*. Others measured academic performance by utilizing the participants' overall GPA in the course after classroom instruction including CT principles (Basnet et al., 2018; Dağ, 2019; Flanigan et al., 2017; Lee & Cho, 2020).

## Characteristics of the assessment instruments

### Assessment media and courses embedded

In Table 5, we summarized the interventions conducted in the selected studies, the formal/informal medium of the assessments, and, if applicable, the courses in which the CT interventions and assessments were embedded in the 33 studies. Results presented in Fig. 7 show that, in the 33 studies, 27 employed instruction in a formal classroom setting during an academic term, 1 was conducted off-line but on campus, as an after-school workshop, and 7 included interventions that were informally performed online. Therefore, the mode of intervention and assessment was split between 82% for formal settings and 18% for informal settings. Figure 8 shows that, among the studies that included in-class interventions and assessments, 16 were courses taught by Computer Science faculty members, 5 were courses on other STEM-related topics, and 3 were courses taught by Education faculty members. Table 5 contains more details about the other types of instructional activities, including the workshops. In sum, most CT assessments were delivered in a formal educational setting and CS was the most dominant discipline to convey content related to CT skills. Other STEM related courses were also favored, followed by courses delivered by faculty members in Education. In the education courses, apart from promoting CT skills, the studies also focused on cultivating participants' abilities to integrate CT into traditional syllabi (Dağ, 2019; Kordaki, 2013).

### Assessment measures

As shown in Fig. 9, most of the 33 studies adopted a combination of multiple measures to assess participants' CT skills from different dimensions (n=12). Yadav et al. (2014) administered a *Computational-Thinking Quiz* as well as a *Computing Attitude Questionnaire* to comprehensively assess students' CT competency. Flanigan et al. (2017) employed a CT knowledge test, *Implicit Theories of Intelligence*

**Table 5** Assessment medium, courses in which it was embedded, and assessment type

Instruments	Media	Course/Format	Assessment type
[1] MICA assessment	In class	Mathematics Integrated with Computers and Applications (MICA) course	Programming assignments (Subject: Math)
[2] Small Basic	In class	15-week programming course	Programming course academic achievement, perceived knowledge of CT, and CT self-efficacy test
[3] CTS and GPA	In class	Québécois English Diploma of College Studies program	CT questionnaire, academic performance
[4] OECD's PIAAC online survey	Online	Online survey	Programme for the International Assessment of Adult Competencies (PIAAC) Test
[5] Robotics/Game-Based assessment	In class	12-week robotics course	Pre- and Post-in-class surveys
[6] Online moodle course with Scratch and quizzes	Online	Scratch coding: Code-Create-Play with Scratch Online course	Moodle course, Scratch assessment
[7] Light-bot quizzes	Online	Lightbot online, Computer programming course	Lightbot quizzes
[8] Quizzes on knowledge of CT	In class	Introductory programming course CS1	Quiz, problem posing-based activity
[9] CT knowledge test	In class	Computer science course (CS1)	Self-efficacy evaluation, CT knowledge test, course grades
[10] NRC's framework assessment	In class	Elementary science teaching methods course	Written assignments, lesson plans, video segments
[11] #5c21	Online	Elementary education course: story2code creative challenge	Scratch projects
[12] CAS online survey	Online	Online survey	Questionnaire
[13] SPOCK test	Online	Introductory CS course	Questionnaire
[14] Lesson plans and video recording of lesson delivery	In class	2-week course on scientific inquiry and engineering design	Concept maps, lesson plans, video recording of lesson delivery
[15] CAAD Test	In class	Introductory CS course	Introductory test (two questionnaires), and CAAD test (knowledge test on computer algorithmic and debugging approach)

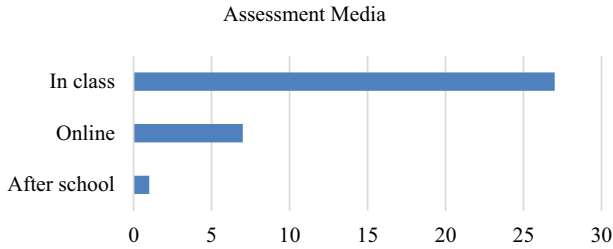
Table 5 (continued)

Instruments	Media	Course/Format	Assessment type
[16] MCT rubric	In class	8-week app development college course	App development assignment
[17] Programmatic assessment report	After-school in-class	Calculus I, Calculus II, Introduction to Computational Mathematics courses	Self-report, problem-solving tasks on math
[18] Genetic algorithms for robot control	In-class	A standard introductory CS course and a Bio-Camp course (combination of introductory CT and biology)	Programming course project
[19] Computational-Thinking Quiz & Computing Attitude Questionnaire	In class	Introductory educational psychology course	Computational-Thinking Quiz, Computing Attitude Questionnaire
[20] CTS	In class	NA	Questionnaire
[21] CT knowledge test; Implicit Theories of Intelligence Scale; profile measures; Course achievement measures	In class	Introductory CS course	Questionnaire, Profile Measures, and Course Achievement Measures
[22] Pre- and post-course tests	Online	Introduction to Data Science course	Pre- and post-knowledge test
[23] Scratch and GPA	In class	Introduction to Programming course	GPA
[24] Scratch Jr	Online	NA	ScratchJr, projects, comprehension of coding skills, and analysis of parents-children interaction
[25] GALT Test	Online	Programming course: 15 weeks, 2-h lecture per week	Pre- and post- Group Assessment of Logical Thinking (GALT) tests
[26] Structured interview, observations	In class	High-school computing course	Structured interviews, observations
[27] Attitude and computational thinking quiz	In class	2.5-day residential workshops	Exit surveys
[28] FS2C framework assessment	In class	Sixteen laboratory sessions	Five different tasks aimed to assess different levels of the FS2C framework proposed by Magana, Newby, and Brophy (2012); pre- and post- test

Table 5 (continued)

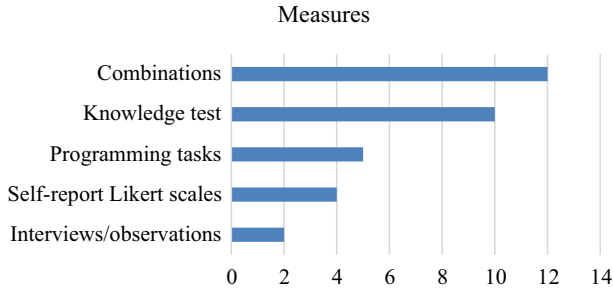
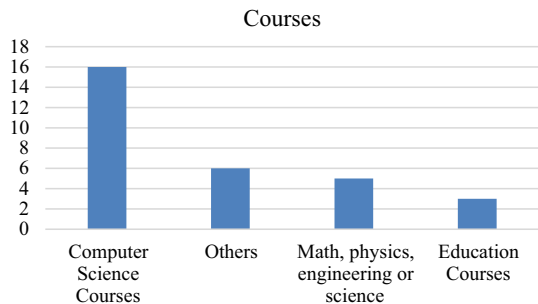
Instruments	Media	Course/Format	Assessment type
[29] TAaAS project assessment	Online	Software Engineering (SOE), System Engineering (SYE), and Business Information Management (BIM)	TAaAS project test
[30] Theoretical knowledge test of process modeling questionnaire	Online	NA	Pencil-and-paper test
[31] Etoys and Scratch	Online	NA	Teacher and administrator interviews; observations of teaching practices
[32] Visual programming projects	In class	Two one-week workshops using Etoys and Scratch	Log data about the participants' practice, strategies, and performance of computational problem-solving activities
[33] Control-value theory of achievement Emotions Self-belief measure	In class	C++ language programming course	Pre- and post-semester surveys (5-point Likert-scale survey, and 3 to 5 multiple choice question assessment)





**Fig. 7** Media of the assessments reviewed

**Fig. 8** Courses in which the assessments were embedded



**Fig. 9** Measures adopted by each assessment instrument

*Scale*, profile measures, and course achievement measures to evaluate all aspects of CT. Magana et al. (2016) used both the *Control-Value Theory of Achievement Emotions Self-belief* measure and academic performance to estimate participants' self-confidence and academic performance in an undergraduate engineering course. The second most frequently used measures are generic knowledge or skill tests. Ten studies adopted knowledge or skill tests, such as the CT knowledge test (Flanigan et al., 2017; Peteranetz et al., 2017), CAAD test (Csernoch et al., 2015), and Group Assessment of Logical Thinking (GALT) test (Kim et al., 2013). Programming tasks in block-based visual programming languages or text-based languages like Python or C++ are adopted as assessment approaches in 5 studies. Additionally, self-report

scales are used in 4 studies, whereas interviews and observations are used in 2 studies to evaluate participants' affective skills in addition to their mastery of CT knowledge.

### **Characteristic of the assessment instrument: format and types of skills assessed**

Table 6 presents the main characteristics of the CT assessment instruments in terms of their forms of administration, format (paper- or computer-based), as well as types of CT skills assessed. The most common form of administration of the assessment tools, as shown in Fig. 10, was a non-automatic, computer-based assessment, which measures both cognitive and non-cognitive skills. In the 33 studies, 5 utilized paper-based assessments, 18 used a computer-based assessment, 4 used both formats, and 6 articles did not clearly state the format. In terms of whether the assessment is automatic, we found that 5 of the studies were automatically assessed, 28 were non-automatically assessed by researchers or their associates, and 2 studies were assessed by a combination of automatic and non-automatic means. Of the studies we reviewed, 15 assessed cognitive skills exclusively, whereas 18 assessed both cognitive and non-cognitive skills. None of the studies assessed only non-cognitive skills related to CT perspectives.

### **CT dimensions assessed: CT concepts, practices, and perspectives**

Although many of the studies in this review utilized different CT frameworks, they all assessed some specific CT principles. Also, there is a large overlap among the various frameworks. Table 7 exhibits subskills within the hybrid framework adopted to evaluate each assessment tool. Figure 11 shows that the top five most commonly examined CT skills are: (a) algorithms and algorithmic thinking, precursor terms for many subskills, such as loops, arrays, and sequencing, which were assessed in 76% of the selected studies ( $n=25$ ); (b) problem-solving, which is assessed in 16 studies; (c) data, which is assessed in 13 studies; (d) logic and logical thinking ( $n=9$ ); and (e) abstraction ( $n=9$ ). Among the top five assessed skills, there are 3 under the category of CT Concepts, 2 from CT Practices, and none from CT Perspectives. The results are consistent with those from CT assessments in K-12 settings (Cutumisu et al., 2019; Shute et al., 2017), in that CT Concepts and Practices are the primary goals of interventions and assessments, whereas CT Perspectives have been largely neglected. However, it should be noted that CT Perspectives, such as communication, collaboration, expression, and questioning have been paid more attention in many of the selected studies, which indicates that post-secondary institutions emphasize non-cognitive abilities, such as creation, expression, communication, and collaboration.

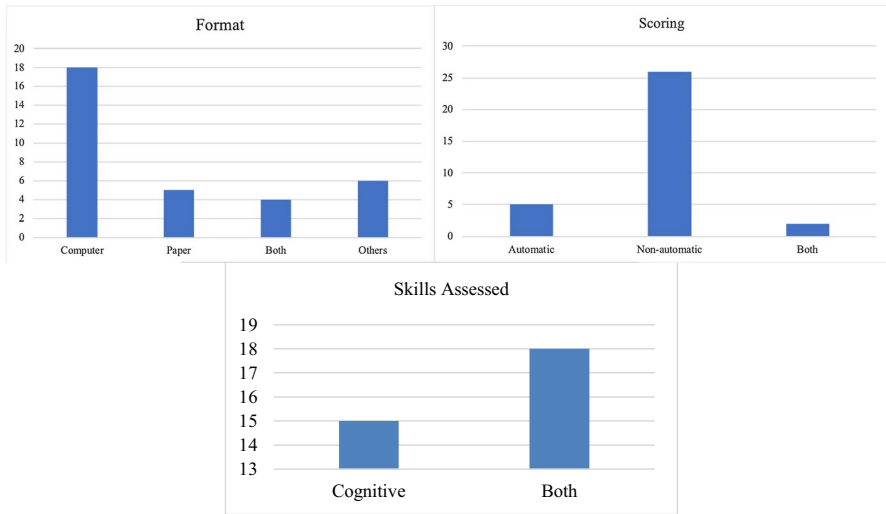
Figure 12 shows that all selected studies assessed CT Concepts, 28 studies (85%) assessed CT Practices, and 19 studies (58%) assessed CT Perspectives. More specifically, 16 assessments (48%) evaluate all three dimensions of CT, 12 assessments (36%) measure CT Concepts and Practices, 3 (9%) assess CT Concepts and Perspectives, and 2 assessed CT Concepts only. Generally, studies in the context of higher

**Table 6** Characteristics of the assessment instruments

Instrument	Paper-based or computer-based	Automatic or non-automatic	Cognitive or non-cognitive
[1] MICA assessment	Computer	Non-automatic	Both
[2] Small Basic	Both	Both	Both
[3] CTS and GPA	Unknown	Non-automatic	Both
[4] OECD's PIAAC online survey	Computer	Automatic	Cognitive
[5] Robotics/Game-based assessment; pre- and post- surveys	Both	Non-Automatic	Both
[6] Online moodle course with scratch and quizzes	Computer	Automatic	Cognitive
[7] Light-Bot Quizzes	Both	Both	Cognitive
[8] Quizzes on knowledge of CT	Paper	Non-automatic	Cognitive
[9] CT knowledge test	Computer	Non-automatic	Cognitive
[10] NRC's framework assessment	Paper	Non-automatic	Both
[11] #5c21	Computer	Automatic	Cognitive
[12] CAS Online Survey	Computer	Automatic	Cognitive
[13] SPOCK test	Computer	Automatic	Both
[14] Lesson plans and video recording of lesson delivery	Lesson plan submission, videos of teaching	Non-automatic	Both
[15] CAAD Test	Computer	Non-automatic	Both
[16] MCT rubric	Computer	Non-automatic	Cognitive
[17] Programmatic assessment report	Paper	Non-automatic	Cognitive
[18] Genetic algorithms for robot control	Computer	Non-automatic	Cognitive
[19] Computational-thinking quiz and computing attitude questionnaire	Computer	Non-automatic	Both
[20] CTS	Computer	Non-automatic	Both
[21] CT knowledge test; Implicit Theories of Intelligence Scale; profile measures; Course achievement measures	Computer	Non-automatic	Both
[22] Pre- and post-course tests	Unknown	Non-automatic	Both
[23] Scratch and GPA	Both	Non-automatic	Both

Table 6 (continued)

	Instrument	Paper-based or computer-based	Automatic or non-automatic	Cognitive or non-cognitive
[24]	Scratch Jr	Computer	Non-automatic	Both
[25]	GALT Test	Unknown	Non-automatic	Both
[26]	Structured interview, observations	Interview/observations	Non-automatic	Both
[27]	Attitude and Computational Thinking Quiz	Computer	Non-automatic	Both
[28]	FS2C framework assessment	Computer	Non-automatic	Cognitive
[29]	TAaAS project assessment	Paper	Non-automatic	Cognitive
[30]	Theoretical knowledge test of process modeling questionnaire	Paper and tasks	Non-automatic	Cognitive
[31]	Etoys and Scratch	Interview/observations	Non-automatic	Cognitive
[32]	Visual programming projects	Computer	Non-automatic	Both
[33]	Control-Value Theory of Achievement Emotions Self-belief measure	Computer	Non-automatic	Cognitive



**Fig. 10** Characteristics of assessments: formats and skills assessed

education aim to measure more dimensions of CT. Table 8 illustrates CT skills from the three dimensions assessed in each assessment tool.

## Discussion, conclusions, and implications

The implementation of CT as a universal problem-solving approach has great potential benefits within post-secondary institutions. However, by systematically reviewing the literature, we found a lack of consensus regarding CT frameworks and assessment instruments. Defining which framework most accurately describes CT is also proving to be contentious for researchers. The results identify four main themes in the CT assessments surveyed in the present work.

First, most CT assessments have relied on testing based on several text-based or visual (e.g., block-based) programming languages for CS or Engineering students. In studies that utilize programming to test CT abilities, it is often difficult to judge the degree to which a participant's responses reflect their competence in specific programming languages, CT principles, or a combination of both. When technical skills such as programming are required to demonstrate CT knowledge, there is a danger that the variables will confound each other, therefore researchers must anticipate this before concluding the reliability and validity of the instruments. This finding echoes the results of other scoping or systematic reviews of CT assessment in K-12 and higher education. Tikva and Tambouris (2021) proposed a CT conceptual model from a systematic review of CT research in K-12 and found that most of the studies viewed CT as associated with programming. They argued that CT drew greatly upon programming concepts and practices, and that programming acted as the backbone of the implementations of CT intervention and assessment tools

**Table 7** Concepts, practices, and perspectives from the hybrid CT framework assessed by the instruments

Dimension	CT skill	Assessment tool
<i>Concepts</i>	Logic and logical thinking	Small Basic, #5c21, CAS Online Survey, Computational-Thinking Quiz & Computing Attitude Questionnaire, GALT Test, Attitude and Computational Thinking Quiz, FS2C framework assessment, TAAAS project, Etoys and Scratch
	Critical thinking	MICA Assessment, CTS, Online Moodle course with Scratch and quizzes
	Data	Small Basic, GALT, Quizzes on knowledge of, NRC's Framework Assessment, Lesson plans & Video recording of lesson delivery, MCT rubric and Generic CT rubric, Programmatic Assessment Report, Genetic algorithms for robot control, Pre- and post-course, GALT Test, Etoys and Scratch, Control-Value Theory of Achievement Emotions Self-belief measure & academic performance, TaaAS project test
	Synchronization	Small Basic, Etoys and Scratch
	Algorithms/Algorithmic thinking	MICA Assessment, Small Basic, CTS, Robotics/Game-Based Assessment, Online Moodle course with Scratch and quizzes, Light-Bot Quizzes, Quizzes on knowledge of CT, CT knowledge test, #5c21, CAS Online Survey, CT knowledge test & SPOCK test, CAAD Test, MCT rubric and Generic CT rubric, Genetic algorithms for robot control, Computational-Thinking Quiz & Computing Attitude Questionnaire, Scratch & GPA, ScratchJr., Structured interview, observations, Attitude and Computational Thinking Quiz, TaaAS project, Etoys and Scratch, Visual programming projects, Control-Value Theory of Achievement Emotions Self-belief measure & academic performance
	Pattern recognition	MICA Assessment, Light-Bot Quizzes, Lesson plans & Video recording of lesson delivery, Computational-Thinking Quiz & Computing Attitude Questionnaire, Theoretical knowledge test of process modeling
	Information processing	PIAAC online survey, NRC's Framework Assessment, Theoretical knowledge test of process modeling
	Evaluation	NRC's Framework Assessment, Lesson plans & Video recording of lesson delivery
	Automation	Light-Bot Quizzes

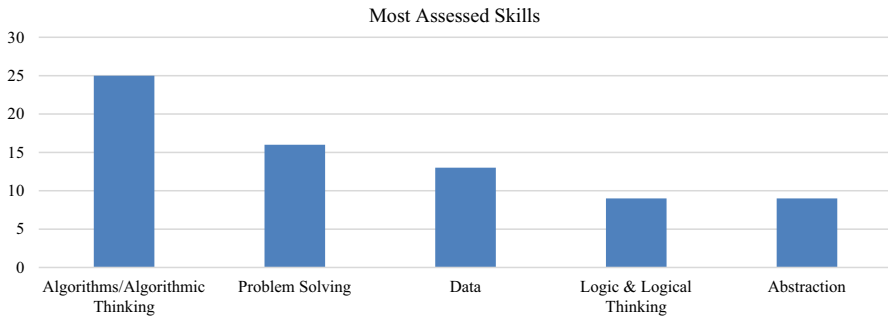
Table 7 (continued)

Dimension	CT skill	Assessment tool
<i>Practices</i>		
Abstraction		MICA Assessment, Small Basic, Light-Bot Quizzes, #5c21, CAS Online Survey, MCT rubric and Generic CT rubric, Computational-Thinking Quiz & Computing Attitude Questionnaire, FS2C framework assessment, Etoys and Scratch
Problem decomposition		MICA Assessment, Small Basic, Light-Bot Quizzes, #5c21, CAS Online Survey, Computational-Thinking Quiz & Computing Attitude Questionnaire, Etoys and Scratch, Visual programming projects
Reasoning		CT knowledge test, CT knowledge test & SPOCK test, GALT Test, FS2C framework assessment
Problem solving		MICA Assessment, CTS, PIAAC online survey, Online Moodle course with Scratch and quizzes, Light-Bot Quizzes, Quizzes on knowledge of CT, CT knowledge test, CAS Online Survey, CT knowledge test & SPOCK test, GALT, Attitude and Computational Thinking Quiz, CAAD Test, TAaAS project, Control-Value Theory of Achievement Emotions Self-belief measure & academic performance
Organization		NRC's Framework Assessment, #5c21
Planning		NRC's Framework Assessment
Testing and debugging		MICA Assessment, Robotics/Game-Based Assessment, #5c21, Lesson plans & Video recording of lesson delivery, CAAD Test, Computational-Thinking Quiz & Computing Attitude Questionnaire, TAaAS project test, Visual programming projects
Modularizing/Modeling		MICA Assessment, NRC's Framework Assessment, #5c21, Lesson plans & Video recording of lesson delivery, Control-Value Theory of Achievement Emotions Self-belief measure & academic performance, FS2C framework assessment
User interactivity		Small Basic, #5c21, MCT rubric and Generic CT rubric, Theoretical knowledge test of process modeling, Etoys and Scratch
Being incremental and iterative		#5c21, Visual programming projects, FS2C framework assessment

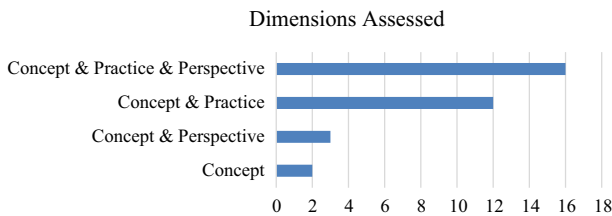
Table 7 (continued)

Dimension	CT skill	Assessment tool
<i>Perspectives</i>	Creativity and creation	MICA Assessment, CTS
	Communication	NRC's Framework Assessment, Lesson plans & Video recording of lesson delivery, GALT Test
	Collaboration and cooperation	MICA Assessment, CTS, Robotics/Game-Based Assessment, Attitude and Computational Thinking Quiz
	Self-efficacy, Self-competency, and confidence	Robotics/Game-Based Assessment, CAAD Test, Implicit Theories of Intelligence Scale, Structured interview, observations, Attitude and Computational Thinking Quiz
	Expressing and questioning	NRC's Framework Assessment, CT knowledge test & SPOCK test, Lesson plans & Video recording of lesson delivery
	Reflection	Lesson plans & Video recording of lesson delivery
	Generalization	FS2C framework assessment





**Fig. 11** CT skills that were assessed in most instruments



**Fig. 12** Number of studies that assessed different combinations of CT dimensions

in the reviewed studies (e.g., Brennan & Resnick, 2012; Voogt et al., 2015). However, over time, the concepts and practices of CT evolved from their CS origins to being perceived as more fundamental skills for both STEM and non-STEM professions. Thus, more effort is still needed to differentiate CT from CS and explore how CT can be involved in daily life in a wide range of domains (Tikva & Tambouris, 2021). Tang et al. (2020) also identified the trend that CT has been increasingly regarded as a set of generic competencies related with both domain-specific knowledge and problem-solving skills other than programming or computing. Hennessey et al. (2017) also examined the CT related terms used in the Ontario elementary school curriculum and found that most of the CS concepts stemming from Brennan and Resnick's (2012) framework were barely mentioned. Therefore, as CT is drawing more attention, CT assessments that are independent from programming tasks should be designed to broaden the scope of CT skills. Accordingly, scaffolding tools could also shift the focus from programming tasks to a wider range of tangible tasks to equip students to apply CT skills in both STEM and non-STEM contexts (Tang et al., 2020; Tikva & Tambouris, 2021).

Second, the review highlights the importance of participants' theoretical understanding of CT and related concepts through skill/knowledge tests. To more accurately assess undergraduates' or older participants' levels of CT competency, most reviewed studies developed their own tests to evaluate individuals' CT skill or knowledge. However, those assessments simply borrowed the theoretical frameworks and CT skills proposed and defined for K-12 education to implement their

**Table 8** CT skills assessed in each assessment instrument

CT assessment tool	CT Concepts	CT Practices	CT Perspectives
MICA assessment [1]	Algorithmic thinking, variables, loops, arrays, graphics, loops, critical thinking, pattern recognition	Modelling, abstraction, debugging, problem solving, problem decomposition, decomposition, evaluation, modules and modularization	Creativity, group collaboration
Small Basic [2]	Parallelism, logical thinking, data representation, synchronization	Abstraction, problem decomposition, user interactivity	Self-competency, confidence
CTS and GPA [3] [20]	Critical thinking, algorithmic thinking	Problem solving	Creativity, cooperativity
OECD's PIAAC online survey [4]	Information processing	Problem solving	NA
Robotics/Game-based assessment; pre- and post-surveys [5]	Algorithm creation, correspondence, sequencing, control flow	Debugging	Self-efficacy, interest in programming, attitudes towards collaboration and cooperativity
Online moodle course with Scratch and quizzes [6]	Algorithms, critical thinking	Problem solving	NA
Light-bot quizzes [7]	Procedures, loops, and conditional statements, pattern recognition, automation, algorithm solving abilities	Problem solving, abstraction	NA
Quizzes on knowledge of CT [8]	Data and data structures, sequence, loop, events, parallelism, conditionals, operators	Problem solving, artificial intelligence	NA
CT knowledge test [9]	Selection, looping, arrays, functions, algorithms, search, sorting	Reasoning, problem solving	NA
NRC's framework assessment [10]	Analyzing and interpreting data, using mathematics and computational thinking, information processing, obtaining and evaluating information	Planning and carrying out investigations, modeling, constructing explanations	Questioning, argument, communicating
#5c21 [11]	code literacy (algorithms, logic), technological systems literacy	Abstraction, organize and model the situation, create a computer program, evaluate and improve iteratively	NA
CAS online survey [12]	Algorithms, logic (algorithmic) thinking	Break down/decomposition, problem solving, abstraction	NA

Table 8 (continued)

CT assessment tool	CT Concepts	CT Practices	CT Perspectives
CT knowledge test; SPOCK test [13]	Selection, looping, arrays, functions, algorithms, search, sorting	Reasoning, problem solving	Self-regulation, motivation, engagement, questioning
Lesson plans and video recording of lesson delivery [14]	Interpreting data, evaluating, classifying	Investigating and using models, debugging	Questioning, expressing, communication, reflection, constructing explanations, arguing from evidence
CAAD test [15]	Algorithmic thinking	Debugging, problem solving	Self-competency
MCT rubric and Generic CT [16]	Events, variables, loops, data persistence, data sharing, conditionals, lists, accelerometer & orientation sensors, location awareness	Naming, screen Interface, procedural abstraction, component abstraction, public web services	NA
Programmatic assessment report [17]	Data	NA	NA
Genetic algorithms for robot control [18]	Functions, variables, loops, data, recursion, algorithms, computational complexity	NA	NA
Computational-thinking quiz and computing attitude questionnaire [19]	Logic, algorithms, pattern recognizing	Identification of problem, decomposition, abstraction, debugging	Attitudes towards CS: definition, comfort, interest, use in the classroom, and use in career
Multiple tools (Implicit theories of intelligence scale; profile measures, course achievement measures, and CT knowledge test) [21]	Selection, looping, arrays, functions, algorithms, search, sorting	Reasoning, problem solving	Implicit intelligence beliefs, motivation and affect, course affect, strategic self-regulation and engagement
Pre- and post-course tests [22]	Computing, data processing, data analysis and visualization	NA	Attitude on CS
Scratch and GPA [23]	Programming skills	NA	Programming motivation
ScratchJr. [24]	Production and comprehension of codes	Spatial talk, task-relevant talk, responsiveness	Question-asking

**Table 8** (continued)

CT assessment tool	CT Concepts	CT Practices	CT Perspectives
The group assessment of logical thinking (GALT) Test [25]	Logical thinking, controlling variables	Proportional reasoning, probabilistic reasoning, correlational reasoning, combinatorial reasoning	Conservation
Structured interview, observations [26]	Computing	NA	Motivation, Self-efficacy, Self-expectations, Constraining beliefs
Attitude and computational thinking quiz [27]	Computational literacy, logical thinking	Problem solving	Technology confidence, collaboration
FS2C framework assessment [28]	Logic	Abstraction, discrimination, logical proportional reasoning, numerical reasoning, modeling	Generalization
Testing algorithmic and application skills (TAaAS) project [29]	Algorithmic skills (logical operators, Decoding pseudo codes, Programming in spreadsheet)	Testing and debugging	NA
Questionnaire: theoretical knowledge test of process modeling [30]	Control-flow pattern	Process modeling knowledge, element interactivity: process structure tree distance, element interactivity: cut-vertex, validity of conclusion	NA
Etoys and scratch [31]	Parallelism, logical thinking, data representation, synchronization	Abstraction, problem decomposition, user interactivity	NA
Visual programming projects [32]	Sequence, selection	Simple iteration, nested iteration, testing, problem decomposition, abutment composition, and nesting composition	Goal attainment and program size
Control-value theory of achievement emotions self-belief measure and academic performance [33]	Data, data analysis and data representation; algorithm, mathematical and numerical skills, conditions, analyze and interpret results	Modeling, simulations	NA

assessment tools. Few studies have expanded existing CT frameworks or taxonomies to accommodate older participants with potentially higher levels of CT competency, nor have they attempted to construct new ones. The present review shows that, among the scarce publications on CT assessments, most articles have targeted K-12 education, with few addressing higher education. As CT has resurfaced relatively recently, research on CT assessments and specifically on CT theory has lagged behind. This issue is most salient for pre-service and in-service teachers. Angeli and Giannakos (2020) discussed four challenges in CT education and highlighted the importance of teacher CT professional development. They advocated that teachers should be systematically prepared to teach and assess CT concepts (Angeli & Giannakos, 2020). Hsu et al. (2018) and Tikva and Tambouris (2021) also suggested informing instructors about CT thoroughly to better support their front-line teaching of CT concepts and design of CT learning activities. However, educators generally reported that they lack a sufficient understandings of CT concepts and training to integrate CT in the existing curriculum (e.g., Denning, 2017; Tikva & Tambouris, 2021; Yadav et al., 2017). Specifically, Yadav et al. (2017) argued that pre-service teachers should be provided with more CT training opportunities across domains, which will enable them to integrate CT in their future teaching practice. The present review reveals that efforts have been made to promote CT research among pre-service and in-service teachers in the last few years. Pre-service and in-service teachers could potentially improve their performance and self-efficacy on CT concepts, practices, and perspectives through interventions and assessments (e.g., Jaipal-Jamani & Angeli, 2017; Lazarinis et al., 2019; Tuttle et al., 2016; Yadav et al., 2014).

Third, the CT assessments reviewed have focused on problem-solving skills, including abilities in troubleshooting algorithms or in solving real-life problems, in addition to theoretical questions that examined participants' understanding of CT terminology and concepts. The respective studies were not usually tied to a specific CT principle. Instead, the adequate understanding of CT concepts together with problem-solving practices can be applied in a wide range of subjects including science, mathematics, engineering, social sciences, and liberal arts. As the results show, most studies have adopted a combination of multiple tools to comprehensively assess individuals CT competency from different perspectives. The concept of *systems of assessments* may be a better fit for a complicated competency such as CT. This is largely because theoretical questions may assess only academic knowledge of CT terminology, without addressing the ability to apply this knowledge. Conversely, an individual might master certain aspects of programming which are encompassed by CT but remain unaware of the CT skills that are being used. This finding is consistent with the results of other existing reviews. For example, Grover, Pea, and Cooper (2015) proposed *Systems of Assessments* for deeper learning of computational thinking in K-12 by including formative multiple-choice quizzes, open-ended programming assignments, and summative assessments of CT skill acquisition and transfer. Tikva and Tambouris (2021) proposed that researchers designed and validated standardized CT assessment tools that support teaching strategies and CT acquisition in a wide range of settings. Hsu et al. (2018) and Tang et al. (2020) also suggested that adopting assessment tools for students of different grade levels, cognitive abilities, and study of programs will not only help assess more fine-grained

CT dimensions, but also reshape students' CT learning strategies and foster their interest in applying CT in various domains.

Fourth, CT assessments that have focused on the motivation, awareness, attitudes, or self-efficacy of CT are mostly targeted to pre-service or in-service teachers. It is important to ensure that teachers are well-versed in recognizing and incorporating CT concepts, practices, and perspectives into their practice, so that they can impart CT knowledge and practices to their students. However, in studies assessing pre-service teachers' motivation, self-efficacy, or confidence in teaching CT, participants' responses displayed vague, uninformed, or inconsistent conceptions of CT (Yadav et al., 2014). This again echoes the second theme we identified in this review, indicating that research on theory of CT should be pushed forward to better inform empirical studies on teaching and assessing CT.

Overall, we found that lack of standardization of CT assessment criteria makes it problematic to compare some CT assessment methods, establish the predictive validity of a study's results in other settings, or draw conclusions without information concerning the educational contexts. Moreover, the purposes and contexts of the studies determine the optimal selection of the assessment tools. However, one phenomenon that calls for attention is that relating academic performance to CT ability proved problematic in the literature. Some researchers found a positive relationship between CT abilities and grades (Lee & Cho, 2020). Other researchers found that students obtained higher marks in undergraduate mathematics when using CT tools (Berkaliev et al., 2014). In contrast, more studies reported a weak or even negative correlation between CT ability and academic achievement (Basnet et al., 2018; Csernoch et al., 2015). The findings suggest that fostering CT as a problem-solving competency does not ensure knowledge gains in a specific domain subject. Rather, CT is a complicated higher-order competency that promotes learning, planning, and applying problem-solving skills.

## Challenges revealed from the review

Some studies included in our review reported sample sizes which are too small to be considered statistically relevant. As well, in other studies, results may only apply to the specific populations being examined and not be generalizable to other populations. Of the studies we reviewed, 8 had sample sizes under 30. Beyond this, extremely localized data sets were also present even in those studies with larger sample sizes, such as studies focusing on only undergraduate pre-service teachers from a specific university or students in a certain introductory CS class.

Furthermore, the reliability and validity of most of the assessment tools have not been investigated except for *CT knowledge test*, *CTS*, *OECD's PIAAC*, and the *Group Assessment of Logical Thinking (GALT) Test*. It is difficult to validate CT assessments, due to the many existing operational definitions and theoretical frameworks of CT competency, which make it difficult to define the CT constructs. Also, CT is a broad term that applies to diverse audiences and educational settings; thus, it is strenuous to establish validity between instruments or accumulate validity evidence for a certain tool. The last challenge lies in the implementation of effective

and efficient CT interventions and assessments to ensure the fulfilment of major subjects as well as students' CT development.

The three challenges mentioned above have been discussed in previous literature (Grover & Pea, 2018). Some additional distinctive issues and challenges for CT assessments have been appearing in the context of higher education. The most salient issue we have identified is the vague boundary between CT skills and CS skills in higher education. Among the 33 articles reviewed, 11 of the empirical studies integrated CT interventions and assessments into CS courses. Some of the studies even directly used course academic achievement to assess CT, which implicitly equates CT to CS (Doleck et al., 2017; Topalli & Cagiltay, 2018). Linking CT directly to CS is problematic, as CT is a comprehensive problem-solving competency that involves multidimensional abilities. Some of the CS skills are only borrowed to tackle complicated problems, but not to replace the entire embodiment of CT skills. Definition and constructs are the key to successful implementation of assessment tools, therefore the primary mission of popularizing CT in higher education is to discriminate CT from CS.

The second challenge deals with identifying CT theoretical frameworks for higher education. Currently most CT frameworks, including Brennan and Resnick's three-dimension framework, Grover and Pea's CT competency framework, or Weintrop et al.'s CT framework for Mathematics and Science, were all originally created for K-12 STEM education, while few were designed for post-secondary contexts. However, the present review found that some of the CT concepts and practices assessed in the selected studies have gone beyond the taxonomy of existing frameworks. Moreover, more CT perspectives are assessed for students from post-secondary institutions, which is to be expected, as this population has more developed inter- and intra-personal skills compared with younger children.

Working with a hybrid framework, an inventory of items could be extended and tested for each concept, practice, and perspective. These items could be examined for reliability, construct, concurrent, and predictive validity. By assembling such an inventory, it could create a core of assessment tools for researchers to use for each specific component of CT. Inclusion of an array of both applied and theoretical items would allow the inventory to convey a more complete picture of CT for higher grade levels and mitigate the likelihood of studies which overemphasize any one dimension of CT.

### **Implications and recommendations for practitioners**

This scoping review examined the existing CT assessments in higher education and addressed several gaps in the literature to inform stakeholders, including researchers and practitioners, regarding future research and policy. First, CT interventions and assessments can be embedded into a wide range of domains including CS, mathematics, engineering, physics, and education across K-12 and beyond. For students attending post-secondary institutions, practitioners can choose appropriate assessment tools to evaluate students' according to different dimensions of CT. For example, teachers can use selected-response or constructed-response items to assess CT

concepts, use surveys to measure CT perspectives, and employ block-based or text-based programming artefacts to evaluate CT practices. Students in higher education are more cognitively and affectively developed compared with K-12 students. Thus, it is plausible to use a combination of tools including generic tests, surveys, and interviews to probe their development of CT self-efficacy and motivation, which are underexplored in the context of K-12.

Second, more attention needs to be focused on the implementation of CT training programs for both STEM majors and pre-service teachers to prepare them for using, teaching, and assessing CT in their future careers. Specifically, efforts should be made to address the gaps between interventions and outcomes to better scaffold student CT learning activities. Another important aspect of research is to understand the relationships among CT skills, intervention activities, teaching practices, and student attitudes to align CT training with programs of studies, gender, and learning outcomes.

Third, although much work has been done to develop CT assessments, there is a paucity of validated assessment tools to accommodate the diverse population in higher education. Future research could be conducted to construct a database of CT concepts, practices, and perspectives aiming for different grade levels and backgrounds. Practitioners could gain insights from mapping CT skills to specific target populations when they need to choose the optimal assessment tools in different scenarios.

## Limitations and future research

Although we followed a systematic approach in this scoping review, the resulting 33 articles may not be considered an exhaustive search. To make the review manageable, we have focused only on journal articles in the selected six databases published from January 2013 to June 2019. As CT stems from CS, where conference proceedings papers account for a greater proportion of research dissemination, in the future, we plan to include CT assessment tools featured in conference proceedings. Also, we find it difficult to recommend specific instruments to post-secondary instructors, due to the lack of validity evidence provided by the reviewed studies. Additionally, 88% ( $n=29$ ) of the studies in this literature review were conducted at Western institutions. Therefore, cultural, geographical, or socioeconomic limitations may make it difficult to draw broad conclusions about the suitability of specific CT assessments or methods in a region or country. This can be regarded as an important consideration for future research, as the world has become increasingly interconnected, and standardization of CT assessment methods will become more important to allow consistency in international comparisons. For instance, CT has been embedded into some K-12 curricula around the world and it is being included in global large-scale assessment frameworks, such as PISA.

To determine the most effective and efficient CT assessments tackling all the challenges discussed in this review, future research can be conducted to examine the various existing CT assessments in terms of reliability and validity evidence; developing new CT theoretical frameworks and underlying constructs targeting higher



education contexts; designing subject-independent interventions and assessment tools for improving long-term learning outcomes instead of simply using CS courses and assessments; and exploring valid and effective methods to assess non-cognitive skills under CT perspectives. Standardization within CT assessment is needed in the long run, which must be based on a consistent theoretical framework. Specific emphasis should be included within assessments to address key theoretical concepts, the flexible application of theory in diverse situations, and assessing essential social skills that foster creative problem solving.

Another aspect which is necessary for future research to address is the social component of CT, which is important in mastering CT skills, but it is rarely emphasized in CT instruction. Therefore, delineating the benefits of social skills in the dimensions of CT would have great potential benefits and would help elucidate potential methods of strengthening these skills during CT interventions.

## Conclusion

The present study reviewed 33 empirical studies on CT assessments in post-secondary education published in academic journals between 2013 and 2019. Several aspects of the studies were analyzed and summarized including the characteristics of the targeted grade levels and programs of study, key features of the assessment tools, and CT dimensions. First, most studies were conducted in the US and sampled undergraduate students from STEM-related majors, such as CS, engineering, and mathematics. Next, pre-service and in-service teachers were also sampled regarding their awareness and abilities of teaching CT, in addition to their own CT skills. Second, most CT assessments reviewed were accompanied by interventions to promote CT in formal or informal educational settings. Among the reviewed studies, 48% favored introductory formal CS courses, followed by elementary education, engineering, mathematics, and data science courses. Only three studies administered CT interventions and assessments via informal settings like workshops or online surveys. Third, commonly-used forms of assessments in the studies reviewed included block-based programming tasks (e.g., Scratch or Alice), text-based programming projects, CT knowledge tests, self-reported scales, interviews, and observations. Fourth, most assessments were computer-based non-automatic assessments that measured a combination of cognitive and non-cognitive skills with a focus on cognitive skills. Fifth, all studies assessed CT concepts, 85% assessed CT practices, and 58% assessed CT perspectives. Results show that CT concepts are still the primary pedagogical goals for related interventions on CT, followed by CT practices, and then by CT perspectives. However, it is a positive sign that more studies have started to include CT perspectives. Among the collection of CT skills, algorithms and algorithmic thinking, problem solving, data, logic and logical thinking, and abstraction are most assessed in the 33 studies. The innate nature of problem solving and computing of CT competency is reflected in the emphasis of skills that are most assessed in the empirical studies.

In an increasingly digitized world, CT has become one of the essential elements of 21st century competencies. This review aimed to investigate the status quo of

research on CT assessments in higher education in the last few years, proposing to make recommendations regarding the appropriate selection of assessment tools for different contexts, and to further identify gaps from the existing literature that would aid future efforts in CT research.

**Acknowledgements** We would like to thank the Social Sciences and Humanities Research Council of Canada—Insight Development Grant (SSHRC IDG) RES0034954 and Insight Grant (SSHRC IG) RES0048110, the Natural Sciences and Engineering Research Council (NSERC DG) RES0043209, and the Killam Cornerstone Operating Grant RES0043207 for supporting this research.

## References

- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*, 106185. <https://doi.org/10.1016/j.chb.2019.106185>
- Arksey, H., & O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology*, *8*(1), 19–32. <https://doi.org/10.1080/1364557032000119616>
- Basnet, R. B., Doleck, T., Lemay, D. J., & Bazalais, P. (2018). Exploring computer science students' continuance intentions to use kattis. *Education and Information Technologies*, *23*(3), 1145–1158. <https://doi.org/10.1007/s10639-017-9658-2>
- Berkaliev, Z., Devi, S., Fasshauer, G. E., Hickernell, F. J., Kartal, O., Li, X., ... & Zawojewski, J. S. (2014). Initiating a programmatic assessment report. *Primus*, *24*(5), 403–420. doi:<https://doi.org/10.1080/10511970.2014.893939>
- Biró, P., Csernoch, M., Máth, J., & Abari, K. (2015). Measuring the level of algorithmic skills at the end of secondary education in hungary. *Procedia - Social and Behavioral Sciences*, *176*, 876–883. <https://doi.org/10.1016/j.sbspro.2015.01.553>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Paper presented at the proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (pp. 1–25).
- Bussaban, K., & Waraporn, P. (2015). Preparing undergraduate students majoring in computer science and mathematics with data science perspectives and awareness in the age of big data. *Procedia - Social and Behavioral Sciences*, *197*, 1443–1446. <https://doi.org/10.1016/j.sbspro.2015.07.092>
- Buteau, C., & Muller, E. (2017). Assessment in undergraduate programming-based mathematics courses. *Digital Experiences in Mathematics Education*, *3*(2), 97–114. <https://doi.org/10.1007/s40751-016-0026-4>
- Chao, P. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, *95*, 202–215. <https://doi.org/10.1016/j.compedu.2016.01.010>
- Csernoch, M., Biró, P., Máth, J., & Abari, K. (2015). Testing algorithmic skills in traditional and non-traditional programming environments. *Informatics in Education*, *14*(2), 175–197.
- Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. <https://doi.org/10.1007/s10956-019-09799-3>
- Dağ, F. (2019). Prepare pre-service teachers to teach computer programming skills at K-12 level: Experiences in a course. *Journal of Computers in Education*, *6*(2), 277–313. <https://doi.org/10.1007/s40692-019-00137-5>
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39.
- Dichev, C., & Dicheva, D. (2017). Towards data science literacy. *Procedia Computer Science*, *108*, 2151–2160. <https://doi.org/10.1016/j.procs.2017.05.240>
- Doleck, T., Bazalais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, *4*(4), 355–369. <https://doi.org/10.1007/s40692-017-0090-9>

- Figl, K., & Laue, R. (2015). Influence factors for local comprehensibility of process models. *International Journal of Human-Computer Studies*, 82, 96–110. <https://doi.org/10.1016/j.ijhcs.2015.05.007>
- Flanigan, A. E., Peteranetz, M. S., Shell, D. F., & Soh, L. (2017). Implicit intelligence beliefs of computer science students: Exploring change across the semester. *Contemporary Educational Psychology*, 48, 179–196. <https://doi.org/10.1016/j.cedpsych.2016.10.003>
- Flores, C. (2018). Problem-based science, a constructionist approach to science literacy in middle school. *International Journal of Child-Computer Interaction*, 16, 25–30. <https://doi.org/10.1016/j.ijcci.2017.11.001>
- Grover, S., Pea, R., & Cooper, S. (2015). Systems of assessments” for deeper learning of computational thinking in K-12. In *Proceedings of the 2015 annual meeting of the American educational research association* (pp. 15–20).
- Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentance, S. Carsten, & E. Barendsen (Eds.), *Computer science education: Perspectives on teaching and learning*. Bloomsbury.
- Hennessey, E. J., Mueller, J., Beckett, D., & Fisher, P. A. (2017). Hiding in plain sight: Identifying computational thinking in the ontario elementary school curriculum. *Journal of Curriculum and Teaching*, 6(1), 79–96.
- Hepppner, P. P., & Petersen, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology*, 29(1), 66–75. <https://doi.org/10.1037/0022-0167.29.1.66>
- Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Iniñiguez-Berrozpe, T., & Boeren, E. (2019). Twenty-first century skills for all: Adults and problem solving in technology rich environments. *Technology, Knowledge and Learning*, <https://doi.org/10.1007/s10758-019-09403-y>
- 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. <https://doi.org/10.1016/j.compedu.2014.11.022>
- Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers’ self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, 26(2), 175–192. <https://doi.org/10.1007/s10956-016-9663-z>
- Kim, B., Kim, T., & Kim, J. (2013). Paper-and-pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, 49(4), 437–459. <https://doi.org/10.2190/EC.49.4.b>
- Knochel, A. D., & Patton, R. M. (2015). If art education then critical digital making: Computational thinking and creative code. *Studies in Art Education*, 57(1), 21–38. <https://doi.org/10.1080/00393541.2015.11666280>
- Kordaki, M. (2013). High school computing teachers’ beliefs and practices: A case study. *Computers & Education*, 68, 141–152. <https://doi.org/10.1016/j.compedu.2013.04.020>
- Korkmaz, Ö. (2012). A validity and reliability study of the online cooperative learning attitude scale (OCLAS). *Computers & Education*, 59(4), 1162–1169. <https://doi.org/10.1016/j.compedu.2012.05.021>
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Lazarinis, F., Karachristos, C. V., Stavropoulos, E. C., & Verykios, V. S. (2019). A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies*, 24(2), 1237–1249. <https://doi.org/10.1007/s10639-018-9823-2>
- Lee, Y., & Cho, J. (2020). Knowledge representation for computational thinking using knowledge discovery computing. *Information Technology and Management*, 21(1), 15–28. <https://doi.org/10.1007/s10799-019-00299-9>
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research*. <https://doi.org/10.1007/s41979-020-00030-2>
- Libeskind-Hadas, R., & Bush, E. (2013). A first course in computing with applications to biology. *Briefings in Bioinformatics*, 14(5), 610–617. <https://doi.org/10.1093/bib/bbt005>

- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Magana, A. J. (2014). Learning strategies and multimedia techniques for scaffolding size and scale cognition. *Computers & Education*, *72*, 367–377. <https://doi.org/10.1016/j.compedu.2013.11.012>
- Magana, A. J., Falk, M. L., Vieira, C., & Reese, M. J. (2016). A case study of undergraduate engineering students' computational literacy and self-beliefs about computing in the context of authentic practices. *Computers in Human Behavior*, *61*, 427–442. <https://doi.org/10.1016/j.chb.2016.03.025>
- Miller, L. D., Soh, L., Chiriacescu, V., Ingraham, E., Shell, D. F., Ramsay, S., & Hazley, M. P. (2013). Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses. In *Paper presented at the* (pp. 1426–1432). doi:<https://doi.org/10.1109/FIE.2013.6685067>
- Mishra, S., & Iyer, S. (2015). An exploration of problem posing-based activities as an assessment tool and as an instructional strategy. *Research and Practice in Technology Enhanced Learning*, *10*(1), 5. <https://doi.org/10.1007/s41039-015-0006-0>
- Organisation for Economic Cooperation and Development (OECD) (2018). *PISA 2021 mathematics framework (draft)*. Retrieved February 9, 2021, from <http://www.oecd.org/pisa/publications/>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books Inc.
- Peteranetz, M. S., Flanigan, A. E., Shell, D. F., & Soh, L. (2017). Computational creativity exercises: An avenue for promoting learning in computer science. *IEEE Transactions on Education*, *60*(4), 305–313. <https://doi.org/10.1109/TE.2017.2705152>
- Peteranetz, M. S., Flanigan, A. E., Shell, D. F., & Soh, L. (2018). Helping engineering students learn in introductory computer science (CS1) using computational creativity exercises (CCEs). *IEEE Transactions on Education*, *61*(3), 195–203. <https://doi.org/10.1109/TE.2018.2804350>
- Ricketts, A. (2014). Preservice elementary teachers' ideas about scientific practices. *Science & Education*, *23*(10), 2119–2135. <https://doi.org/10.1007/s11191-014-9709-7>
- Robertson, J., Macvean, A., & Howland, K. (2013). Robust evaluation for a maturing field: The train the teacher method. *International Journal of Child-Computer Interaction*, *1*(2), 50–60. <https://doi.org/10.1016/j.ijcci.2013.05.001>
- Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1), 42. <https://doi.org/10.1186/s41239-017-0080-z>
- Sentance, S., & Cszimadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. <https://doi.org/10.1007/s10639-016-9482-0>
- Sheehan, K. J., Pila, S., Lauricella, A. R., & Wartella, E. A. (2019). Parent-child interaction and children's learning from a coding application. *Computers & Education*, *140*, 103601. <https://doi.org/10.1016/j.compedu.2019.103601>
- Shell, D. F., & Soh, L. (2013). Profiles of motivated self-regulation in college computer science courses: Differences in major versus required non-major courses. *Journal of Science Education and Technology*, *22*(6), 899–913. <https://doi.org/10.1007/s10956-013-9437-9>
- Sherman, M., & Martin, F. (2015). The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, *30*(6), 53–59.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational thinking in high school science classrooms: Exploring the science “framework” and “NGSS.” *Science Teacher*, *81*(5), 53–59.
- Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, *22*(3), 443–463. <https://doi.org/10.1007/s10758-017-9328-x>
- Tang, K., Chou, T., & Tsai, C. (2020). A content analysis of computational thinking research: An international publication trends and research typology. *The Asia-Pacific Education Researcher*, *29*(1), 9–19. <https://doi.org/10.1007/s40299-019-00442-8>
- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education*, *162*, 104083.

- Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with scratch. *Computers & Education*, 120, 64–74. <https://doi.org/10.1016/j.compedu.2018.01.011>
- Tuttle, N., Kaderavek, J. N., Molitor, S., Czerniak, C. M., Johnson-Whitt, E., Bloomquist, D., ... & Wilson, G. (2016). Investigating the impact of NGSS-aligned professional development on PreK-3 teachers' science content knowledge and pedagogy. *Journal of Science Teacher Education*, 27(7), 717–745. doi:<https://doi.org/10.1007/s10972-016-9484-1>
- 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.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Whetton, D. A., & Cameron, K. S. (2002). *Answers to exercises taken from developing management skills*. Northwestern University.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society a: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725.
- Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. *Emerging research, practice, and policy on computational thinking* (pp. 205–220). Springer.
- Yadav, A., Mayfield, C., Zhou, N., Hambruch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*. <https://doi.org/10.1145/2576872>
- Yesil, R., & Korkmaz, O. (2010). Reliability and validity analysis of the multiple intelligence perception scale. *Education*, 131(1), 8–33. Retrieved from <https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=00131172&v=2.1&it=r&id=GALE%7CA239813821&sid=googleScholar&linkaccess=abs>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Chang Lu** is a doctoral student in Education at the University of Alberta. She is a member of the Measurement, Evaluation, and Data Science area in the Department of Educational Psychology, Faculty of Education, University of Alberta. Her research focuses on applications of machine learning in education and measurement.

**Rob Macdonald** is an Education student at the University of Alberta. He is a recipient of the Roger S. Smith Undergraduate Research Award that facilitated his research in the Department of Educational Psychology, Faculty of Education, University of Alberta. His research focuses on the role of computational thinking in education.

**Bryce Odell** is a Master's student in Education at the University of Alberta. He is a member of the Measurement, Evaluation, and Data Science area in the Department of Educational Psychology, Faculty of Education, University of Alberta. His research focuses on applications of measurement techniques in educational technology.

**Vasyl Kokhan** is a MITACS B.Sc. in Computing Science Summer student at the University of Alberta. He is a recipient of the Mitacs Globalink Undergraduate Research Award that facilitated his research in the Department of Educational Psychology, Faculty of Education, University of Alberta. His research focuses on computational thinking and machine learning.

**Carrie Demmans Epp** is an assistant professor of Educational Technology and Human-Computer Interaction at the Department of Computing Science, Faculty of Science, University of Alberta. Her research interests include using technology to assess student knowledge and skills with a focus on enabling the

adaptation of learning activities and environments. She also uses modelling and analytic techniques that enable adaptation and the reporting mechanisms that supply feedback to students based on those models.

**Maria Cutumisu** is an associate professor at the Department of Educational Psychology, Faculty of Education, University of Alberta. She is a faculty member in the Measurement, Evaluation, and Data Science area at the Faculty of Education and an adjunct professor in the Department of Computing Science at the University of Alberta. Her research draws on computing science and education, including computational thinking, human and machine learning, feedback processing and generation, and game-based assessments.

## Authors and Affiliations

Chang Lu<sup>1</sup> · Rob Macdonald<sup>1</sup> · Bryce Odell<sup>1</sup> · Vasyl Kokhan<sup>1</sup> ·  
Carrie Demmans Epp<sup>1</sup> · Maria Cutumisu<sup>1</sup> 

Chang Lu  
clu4@ualberta.ca

Rob Macdonald  
rimacdon@ualberta.ca

Bryce Odell  
btodell@ualberta.ca

Vasyl Kokhan  
kokhan@ualberta.ca

Carrie Demmans Epp  
demmanse@ualberta.ca

<sup>1</sup> University of Alberta, Edmonton, Canada