2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #32342

Computational Thinking Frameworks used in Computational Thinking Assessment in Higher Education. A Systematized Literature Review.

Ms. Laura Melissa Cruz Castro, Purdue University at West Lafayette (PPI)

Laura Melissa Cruz Castro is a M.S student in the Department of Electrical and Computer engineering under the advice of Dr. Mirelle Boutin and a Ph.D. student in the Department of Engineering Education under the guidance of Dr. Kerrie Douglas, both at Purdue University. She holds a B.S. in Statistics from Universidad Nacional de Colombia and a M.S. in Business Intelligence from Universität de Barcelona. Her research interests include educational data mining, ethical considerations regarding the use of data in education, assessment in engineering education, and the statistics curriculum for engineering. She is a member of Eta Kappa Nu (HKN), Purdue.

Ms. Huma Shoaib, Purdue University

Huma Shoaib is an engineering education graduate student at Purdue working with The Weldon School Biomedical Engineering. Her research interests are; identifying computational thinking patterns in engineering students and underrepresentation of women in engineering.

Dr. Kerrie A Douglas, Purdue University at West Lafayette (COE)

Dr. Douglas is an Associate Professor in the Purdue School of Engineering Education. Her research is focused on improving methods of assessment in engineering learning environments and supporting engineering students.

Computational Thinking frameworks used in Computational Thinking assessment in higher education. A systematized literature review.

Abstract

This research paper presents a literature review of Computational Thinking (CT) frameworks and assessment practices. CT is a 21st century way of solving a problem. It refers specifically to the effective methods when trying to solve a problem with a machine or other computational tools. In the past few years, CT researchers and educationists' significant movement started to look for a formal definition and composition of CT in K-12 and higher education. From this effort, over 20 different definitions and frameworks for CT have emerged. Although the availability of literature on CT has been increasing over the last decade, there is limited research synthesis available on assessing CT better. Besides, it is known that in higher education designing assessments for CT is challenging and one of the primary reasons is that the precise meaning of CT is still unknown. This research paper, therefore, presents a systematized literature review on CT frameworks and assessment practice. We search three different databases and review 19 journal articles that address CT assessment in higher education to answer the following two research questions: 1) What does the literature inform us about practices and types of assessments used to evaluate CT in higher education? 2) Which frameworks of CT are present in literature to support CT assessment in higher education? The critical components of this review focus on frameworks and assessment practices based on CT. We develop a synthesis of suggestions and explanations to answer the proposed questions based on literature from recent research in CT. Based on our initial synthesis, we found a disconnect between theory and practice. Specifically, neither the ideas within CT frameworks nor those from CT assessment research are utilized. Therefore, there is a dire need to connect the two for practical implementation and further research in CT in higher education.

1. Introduction and background

In 2006, Jeannette Wing, at that time, head of the computer science department at Carnegie Mellon, promoted the term computation thinking (CT). She defined computational thinking as "a range of mental tools that reflect the breadth of the field of computer science."[1] (p.33). In this same article, Wing invited the community to see CT not only as a set of skills concerning computer scientists but every professional.

After 2006 a significant movement of supporters of CT started to look for a formal definition and composition of CT. In the last 14 years, over 20 definitions and frameworks for CT have been proposed [2], [3]. Nevertheless, although there have been increasing efforts to compile a single definition, those were unsatisfactory[4]. It is the same case for the CT frameworks [2]. In general, frameworks have defined CT as at most a three-dimensional construct composed of concepts, practices, and perspectives. Some frameworks include three dimensions or an intersection of them, while others use one or two dimensions [5]–[7]. The diversity in theoretical literature has led to empirical research studies investigating CT use of various definitions and frameworks, generating difficulties to compare and use the empirical research produced.

One issue that arises from the divergence in definitions and frameworks is the assessment of CT. The assessment of CT has been cataloged as a "trouble spot related to the generation of understanding of CT"[4]. Although it is a more practical aspect, assessment is directly affected by the divergence in definitions and frameworks. The assessment of CT practices depends directly on the definition of the construct as well as the composition. Despite these problems and divergences, researchers across the globe have proposed tools, tests, and approaches to assess computational thinking [2]. Due to the high demand for CT research, in less than 15 years, a large number of papers have been produced, leading multiple studies analyzing the literature available on the assessment of CT using different methods as is literature-based perspectives, systemic mappings, and literature reviews.

In 2019, researchers performed a scoping review on empirical research on CT assessments; this literature review addressed characteristics of the assessment, demographics of the empirical studies, psychometric evidence, and assessment tools used [8]. They revealed that CT assessments lacked psychometric reliability and were mainly focused on K-12 settings; in addition, they concluded that literature needed to build on CT definitions, frameworks, and models. Another study in 2020, published by [9], dealt with automated grading and assessment tools used to assess CT in K-12 settings. They found that Scratch ed [10] was the most used educational tool to assess CT and provided a state-of-the-art approach to assessment using automated program assessment. Tang et al., focused on assessing computational thinking and performed a systematic literature review of the empirical studies available[2]. They reviewed educational contexts, constructs measured, assessment tools, and validity evidence. They found that CT assessment research was devoted to K-8 settings and that only 45% of the studies provided validity evidence for the assessment used.

Although some of the literature in CT assessment has been reviewed, there are two missing pieces that we aim to contribute to the current research. On the one hand, research in CT has focused mainly on K-12 settings, with researchers calling for more research in higher education settings [11]. Additionally, in the case of literature reviews, there have not been yet CT assessment reviews focusing only on higher education settings, and attending the call from Cutumisu et al., our research focuses on frameworks used in computational thinking in higher education contexts. Notably, we address the following research questions:

1) What does the literature inform us about practices and types of assessments used to evaluate CT in higher education?
2) Which frameworks of CT are present in literature to support CT Assessment in higher education?

To answer these questions, we used a systematized literature review. The following section corresponds to the details concerning the methodology.

2. Methods

This systematized literature review possesses the most “easily identified elements of systematicity": (1) Searching one or more databases and then coding, and (2) systematically analyzing all retrieved results. In this section, the sources of information and decisions, followed by the inclusion/exclusion criteria and the description of the data analysis process for the articles, are described. In total, 19 articles were the result of the inclusion and exclusion criteria. The procedure used for including and excluding the papers was the four-phase flow diagram of the PRISMA Statement [12]. A recommended quality procedure is used widely in systematic literature reviews.

2.1 Data Sources

Academic and peer-reviewed papers published in 2006 – 2020 on computational thinking in the context of CT assessment in higher education were retrieved. The articles reviewed were published in peer-reviewed journals, and three databases were used for this purpose: ERIC, Education Source, and ProQuest (Compendex).

2.2 Inclusion and exclusion procedure

For the database search, two keywords were used (see Table 1). For both keywords, the search was framed in the databases in Abstract (AB), Keyword/identifier (IF), Title (TI), and Subject (SU). As a result, 15 items were found in ProQuest and 46 items in ERIC, 44 articles Education Source. From which 61 corresponded to unique articles.

The decision to include articles in the literature review was made using the following criteria. Empirical articles that were published in peer-reviewed journals (scholarly articles) were included. In addition, articles were written in the English language between the years 2006-2020. The restriction of time is related to an initial search in the database Web of Science, which generates a citation report about how some topics have been cited in academic documents. In this case, the topic began to be cited in 2006, and 2018 reached its highest peak. An additional argument for limiting the search to only those years was the publication of the seminal Jennings article. Jennings’ article is associated with the accounting of the term "Computational Thinking," which was published in 2006[1]. In that case, for limiting the scope, this literature review is limited to assessments in higher education, specifically in science and engineering classrooms. The search did not include terms related to higher education. This decision was made based on the literature review; there were few studies in higher education, so we preferred to manually exclude articles based on the context while reading the full abstract or text depending on the article.

Table 1. Keywords for the initial search

First level		Second level	
Keyword	Criteria	Keyword	Criteria
"Computational thinking"	AB or IF or TI or SU	Assessment OR test OR evaluation	AB or IF or TI or SU

Note: AB = Abstract, IF = Identifier/Keyword, TI = Title, SU = Subject. Source: Own elaboration.

After obtaining the unique results from the search, a title screening and filtering were followed by an abstract screen and filtering procedure. All abstracts from the articles obtained in the search were read individually by the two researchers. The process was followed by the inclusion of the articles in which there was agreement and the revision of those in which the inclusion conclusion was different. Each paper having a different inclusion conclusion was discussed until agreement on a decision was reached. Of the 42 articles excluded in the screening process, 37 were excluded because of their context; most of these contexts were K-12 settings. Three were excluded because the assessment was related to other constructs: intentions, pedagogical knowledge, and mathematical maturation. The remaining two articles were excluded because they were theoretical and did not pertain to the scope of the literature review. After the abstract decision process was completed, 24 articles were selected for the full read. Out of those 19-peer reviewed articles were included in the final synthesis (see Figure 1).

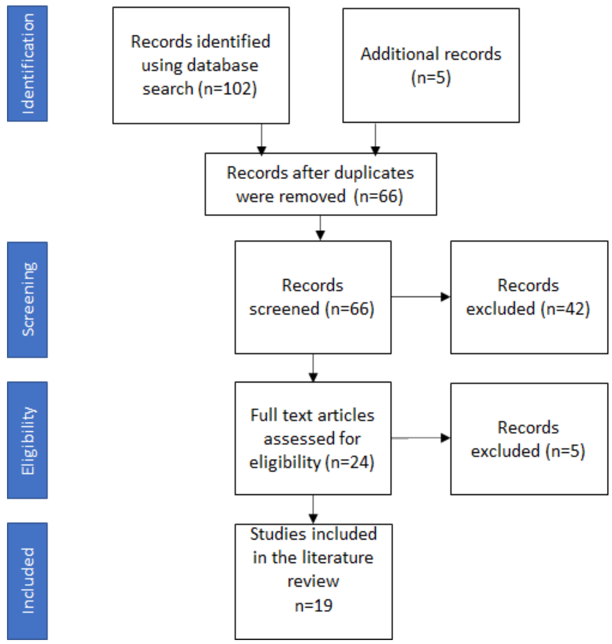

Figure 1: Prisma diagram including the information of the number of excluded and included articles in each step of the reviewing process [12]

3. Findings

Different strategies for assessing the development of computational thinking in higher education were found. These summaries of findings will be discussed in the following themes (a) tests, instruments, and portfolios, (b) makeshift environments and online games, and (c) formal undergraduate courses.

(a) Computational thinking assessment through tests, instruments, and portfolios

Korkmaz, Çakir, & Özden [13] used the ISTE (2015) framework for computational thinking [14], which covers the skills of creativity, algorithmic thinking, critical thinking, problem-solving, establishing communication, and establishing cooperation. A scale has been developed, presented, and validated to determine the computational thinking skill levels of the students called CTS. CTS is a five-point Likert scale and consists of 29 items that could be collected under five factors. Similarly, Gouws, Bradshaw, & Wentworth [15], designed and administered a test to investigate the role that computational thinking plays in the experience of introductory computer science students at a South African university. They use bloom's taxonomy to ensure that teaching and assessments are targeted at the appropriate cognitive level. Question papers were analyzed and classified according to the six CT classes, with some questions falling into multiple classes. Some examples of CT assessment through tests also include selected-response tests (e.g., [5], [16]) Walden, Doyle, Garns, & Hart [17], developed a test to assess students' computational thinking and critical thinking skills to note the improvement in skills after students participated in a principle of informatics course. A pre and post-test was conducted where the test was a combination of multiple-choice and short-answer questions. The test evaluated students' responses to simple algorithms, efficient sorting, quality of digital information storage, and file structures; however, the article did not contain instrument validity evidence. Similarly, Arastoopour [18] developed pre and post-test to assess CT where they noted an increase from pre to post course in (1) exploring a model and explaining how interactions produce system behaviors, (2) identifying simplifications, and (3) understanding a model's range of applications in a biology CT STEM Unit. Pulimood, Pearson, and Bates [19], also measured student outcomes in Computational Skills through self-assessments in pre-tests and post-tests over four semesters. They analyzed statistically significant differences in self-assessment between Computer science and engineering and journalism students in pre-test measures; they reported journalism majors consistently and significantly rated their computational thinking skills lower than their peers.

Unlike other CT tests, Tang, Yin, Lin, & Hadad [20] proposed that Bebras tasks do not rely on any software or content subjects. They consist of primarily multiple-choice items and measure "pure" CT skills. Bebras has been widely used in more than six countries, such as Germany and the UK. A computer engineering doctoral student and three educational psychologists selected the six Bebras items using the following criteria: (a) the items were designed for students in high school; (b) the items measured multiple CT components so that we can well cover all the CT components; (c) the items vary in difficulty levels. The finalized Bebras items are in Appendix A. CT components evaluated abstraction, decomposition, algorithmic thinking, generalization, and evaluation. Román-González, Moreno-León, & Robles [21], introduce computational thinking in the context of Higher Education creative programming activities. The study engaged undergraduate students in a creative programming activity using Scratch computational thinking scores of an automatic analysis tool and the human assessment of the creative programming projects. They follow the framework of Selby and Woollard[22]. Dr. Scratch's test results were computed based on seven criteria: abstraction, parallelism, logic, synchronization, flow control, user interactivity, and data representation. Sondakh, Osman, & Zainudin [23], proposed their Holistic Assessment of Computational Thinking Framework, an instrument to assess CT holistically for Indonesian undergraduate students. The instrument recommends eleven CT concepts: abstraction, algorithmic thinking, automation, decomposition, debugging, evaluation, generalization, problem-solving, teamwork, communication, and spiritual intelligence.

Another artifact to assess computational thinking presented by [6] is portfolios. Portfolios recently gained traction within computer science education to assess students' computational thinking and practices. Whereas traditional assessments such as exams tend to capture learning within artificial settings at a single point in time, portfolios provide more authentic opportunities to document a trajectory of students' learning and practices in everyday contexts. Giving students opportunities to describe their process can provide deeper insights into their computational understanding and practices [6].

(b) Computational thinking assessment in makeshift environments and online games

Hadad et. Al, [24] claim that makerspace helps acquire computational thinking skills in an informal public library environment by using formative assessment in physics and engineering integrated into making activities in a two-week-long summer activity. Hadad et al. [24] used the Computational Thinking framework given by Csizmadia, Standl, & Waite [25], using elements such as: decomposition, pattern recognition, abstraction, algorithm design, and evaluation. They reported approaches centered around the knowledge, the learner, and the community, allowing for broad integration in effective learning environments. Similarly, Lee and Recker [26] assess their paper circuits program implemented through a makeshift space by having participants share their projects with the group or through informal competitions and exhibits.

Kanika & Chakraborty [27], did a literature review, and their findings support visual and game-based programming as a successful way to introduce programming and computational thinking. Similarly, Spieler & Kemeny [28] used a game development workshop to assess the computational thinking development in 84 students who developed games. They use the theoretical framing as a game-based Learning approach, an inherent constructionist approach [29]. Regarding coding, the constructionist theory states that it is much more effective when students program personalized games instead of just learning about computing [30].

In order to reach a comprehensive assessment of CT skills, a combination of complementary assessment tools is recommended by [31]. The proposed tools such as Scrape and Dr. Scratch [10] can automatically assign a CT score in terms of basic CT concepts such as abstraction and problem decomposition, parallelism, logical thinking, synchronization, flow control, user interactivity, and data representation by analyzing the game source code. A case study was performed with the participation of undergraduate computer science students in the last year of their studies to validate the effectiveness of the CT assessment framework.

(c) Computational thinking assessment in formal undergraduate courses

Libeskind-Hadas and Bush [32], assessed student success by scoring the student solutions on their final projects from a Biocomp course at Harvey Mudd College for correct functionality, quality of design, ability to communicate their computational approach in writing, and ability to use their program for scientific exploration as assessed however the framework followed for assessment was not clearly described. Mishra and Iyer [33], used a grounded theory-based assessment framework in an AI course to assess the quality of questions students pose after a lecture on reflex agents in AI. The assessment was done on one of the components of Computational thinking proposed as logical reasoning ability.

Cruz Castro, Magana, Douglas, & Boutin [34], used formative assessment, conformed by multiple artifacts such as exams and homework graded at the learning objective level, to evaluate the progression of students in CT practices. In this study, the researchers used the framework proposed by Weintrop [5], arguing its closeness to engineering needs. They concluded that some practices have a high impact on student performance, such as troubleshooting and debugging, which need to be mastered to acquire more complex skills.

Mendoza Diaz, Meier, Trytten, & Yoon [35], developed their computational thinking assessment framework for engineering undergraduates based on ABET student learning outcomes called ECTD, which incorporates five aspects of computational thinking: (a) Abstraction, (b) Algorithmic Thinking and Programming, (c) Data Representation, Organization, and Analysis, (d) Decomposition, and (e) Impact of Computing. The most recent ABET student outcomes from 2019-2020 are aligned with the ECTD framework. They reported that based on assessment, the first-year engineering course increases the first-year engineering students' computational thinking skills statistically significantly.

In the context of non-computer science undergraduates, an influential paper and pencil programming strategy is presented by [36], to improve the understanding and use of computational thinking for non-computer majors during a formal course. The strategy was created for 110 participating non-computer science majors. This method allows students to create diagrams, cues, codes, symbols, tables, among others, for a logical idea or solution. Framework adapted was components of CT, i.e., Steps of the process such as collecting data, analyzing data, breaking problems into smaller parts, pattern recognition, abstracting, developing algorithms, and build, simulating/testing, and debugging models.

4. Discussion and Future Directions

Defining computational thinking appropriate to an engineering context is difficult because of varying computational thinking definitions across CT literature, which are very generic and rise from the K-12 context [3]. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) have well-designed definitions and frameworks that can be opted from CT assessment[14]. However, they were developed for K-12 education, are somewhat broad, and do not specifically target undergraduate engineering. In addition to the difficulty of having very generic definitions of Computational Thinking coming from a K-12 context, the frameworks based on which CT is being assessed even in undergraduate engineering space are also coming from K-12 space. The need for a specialized and consensual framework to assess engineering undergraduates' CT skills and learning outcomes is evident [35].

Students' computational concepts are being assessed through assessment instruments like pre-post-tests, surveys, and artifacts for which frameworks are also present. Some work is also being done in assessing the computational practices of engineering students through portfolios, learner documentation, and interviews. However, the space of assessing computational perspectives in engineering students still needs considerable work and the development of new frameworks that assess the computational practices and perspectives across multiple time frames of computational skill development.

Most of the work done in computational thinking assessment is being done in programming, i.e., [6] computational thinking framework and assessment utilizing Scratch tool are also based on programming. Most of the pre-existing assessment frameworks are based on assessing programming competence development while using tools like Dr. scratch [10]. While programming is an essential tool for engineers, computational thought involves far more than just the ability to program and develop games online.

CT assessment space should focus on developing new specialized CT frameworks for engineering students and connecting them with innovative forms of assessment that target computational skills development and the computational practices and perspectives of engineering students. In addition, assessment developers on CT need to provide a framework to understand the construct evaluated.

5. Limitations

The current study aimed to review the literature corresponding to assessing the CT frameworks in higher education. Nevertheless, the search used only focused on three databases and the English language, limiting the scope of the review. Future research is expected to include various terms that for different disciplines imply assessment, as it is the term scale found in related literature.

6. Acknowledgments

The research reported in this paper was supported in part by the US National Science Foundation under the awards EEC #1826099 and EEC #144923. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation.

7. Bibliography

[1] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 68-1-68–18, 2006.

[2] X. Tang, Y. Yin, Q. Lin, R. Hadad, and X. Zhai, "Assessing computational thinking: A

systematic review of empirical studies," *Comput. Educ.*, vol. 148, no. January, p. 103798, 2020.

[3] H. Shoaib and S. P. Brophy, "A systematic literature-based perspective towards learning and pedagogy of computational thinking," *ASEE Annu. Conf. Expo. Conf. Proc.*, vol. 2020-June, 2020.

[4] P. J. Denning, "Computational Thinking in Science," *Best Writ. Math. 2018*, pp. 67–77, 2019.

[5] D. Weintrop *et al.*, "Defining computational thinking for mathematics and science classrooms," *J. Sci. Educ. Technol.*, vol. 25, no. 1, pp. 127–147, 2016.

[6] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, 2012, vol. 1, p. 25.

[7] T. Lowe and S. Brophy, "An operationalized model for defining computational thinking," *Proc. - Front. Educ. Conf. FIE*, pp. 1–8, 2017.

[8] M. Cutumisu, C. Adams, and C. Lu, "A Scoping Review of Empirical Research on Recent Computational Thinking Assessments," *J. Sci. Educ. Technol.*, vol. 28, no. 6, pp. 651–676, 2019.

[9] N. Da Cruz Alves, C. Gresse Von Wangenheim, and J. C. R. Hauck, "Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study," *Informatics Educ.*, vol. 18, no. 1, pp. 17–39, 2019.

[10] J. Moreno-Leon, G. Robles, and M. Roman-Gonzalez, "Comparing computational thinking development assessment scores with software complexity metrics," *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. 10-13-April-2016, no. April, pp. 1040–1045, 2016.

[11] J. A. Lyon and A. J. Magana, "Computational thinking in higher education: A review of the literature," *Comput. Appl. Eng. Educ.*, vol. 28, no. 5, pp. 1174–1189, 2020.

[12] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and T. P. Group, "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement," *PLoS Med.*, vol. 6, no. 7, p. e1000097, Jul. 2009.

[13] Ö. Korkmaz, R. Çakir, and M. Y. Özden, "A validity and reliability study of the computational thinking scales (CTS)," *Comput. Human Behav.*, vol. 72, pp. 558–569, 2017.

[14] ISTE and CSTA, "Leadership Toolkit," 2011. [Online]. Available: https://id.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4.

[15] L. Gouws, K. Bradshaw, and P. Wentworth, "First year student performance in a test for computational thinking," *ACM Int. Conf. Proceeding Ser.*, pp. 271–277, 2013.

[16] S. Grover and R. Pea, “Computational Thinking in K-12: A Review of the State of the Field,” *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013.

[17] M. Walden, R. Garns, and Z. Hart, “An informatics perspective on Computational Thinking,” *ACM*, no. 581, pp. 1635–1646, 2013.

[18] G. Arastoopour Irgens *et al.*, “Modeling and Measuring High School Students’ Computational Thinking Practices in Science,” *J. Sci. Educ. Technol.*, vol. 29, no. 1, pp. 137–161, 2020.

[19] S. M. Pulimood, K. Pearson, and D. C. Bates, “A study on the impact of multidisciplinary collaboration on computational thinking,” *SIGCSE 2016 - Proc. 47th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 30–35, 2016.

[20] Y. Yin, R. Hadad, X. Tang, and Q. Lin, “Correction to: Improving and Assessing Computational Thinking in Maker Activities: the Integration with Physics and Engineering Learning (Journal of Science Education and Technology, (2020), 29, 2, (189-214), 10.1007/s10956-019-09794-8),” *J. Sci. Educ. Technol.*, vol. 29, no. 2, p. 215, 2020.

[21] M. Román-González, J. Moreno-León, and G. Robles, “Complementary tools for computational thinking assessment,” *Proc. Int. Conf. Comput. Think. Educ.*, pp. 154–159, 2017.

[22] C. Selby, “Computational Thinking : The Developing Definition,” *ITiCSE Conf. 2013*, pp. 5–8, 2013.

[23] D. E. Sondakh, K. Osman, and S. Zainudin, “A proposal for holistic assessment of computational thinking for undergraduate: Content validity,” *Eur. J. Educ. Res.*, vol. 9, no. 1, pp. 33–50, 2020.

[24] R. Hadad, K. Thomas, M. Kachovska, and Y. Yin, “Practicing Formative Assessment for Computational Thinking in Making Environments,” *J. Sci. Educ. Technol.*, vol. 29, no. 1, pp. 162–173, 2020.

[25] A. Csizmadia, B. Standl, and J. Waite, “Integrating the constructionist learning theory with computational thinking classroom activities,” *Informatics Educ.*, vol. 18, no. 1, pp. 41–67, 2019.

[26] I. Lee *et al.*, “Computation thinking for youth in practice,” *ACM Inroads*, vol. 2, no. 1, pp. 32–37, 2011.

[27] Kanika, S. Chakraverty, and P. Chakraborty, “Tools and Techniques for Teaching Computer Programming: A Review,” *J. Educ. Technol. Syst.*, vol. 49, no. 2, pp. 170–198, 2020.

[28] B. Spieler and F. Kemeny, “Design, complexity, and coding: A framework to evaluate games,” *Proc. 14th Int. Conf. Game Based Learn. ECGBL 2020*, pp. 558–566, 2020.

[29] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, “A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming,” *Procedia - Soc. Behav. Sci.*, vol. 47, pp. 1991–1999, 2012.

[30] Y. B. Kafai, “Playing and making games for learning: Instructionist and constructionist perspectives for game studies,” *Games Cult.*, vol. 1, no. 1, pp. 36–40, 2006.

[31] I. Altanis and S. Retalis, “A multifaceted students’ performance assessment framework for motion-based game-making projects with Scratch,” *EMI. Educ. Media Int.*, vol. 56, no. 3, pp. 201–217, 2019.

[32] R. Libeskind-Hadas and E. Bush, “A first course in computing with applications to biology,” *Brief. Bioinform.*, vol. 14, no. 5, pp. 610–617, 2013.

[33] S. Mishra and S. Iyer, “An exploration of problem posing-based activities as an assessment tool and as an instructional strategy,” *Res. Pract. Technol. Enhanc. Learn.*, vol. 10, no. 1, 2015.

[34] L. M. Cruz Castro, A. Magana, K. A. Douglas, and M. Boutin, “Analyzing Students ’ Computational Thinking Practices in a First-Year Engineering Course,” *IEEE Access*, pp. 1–11, 2020.

[35] N. V. Mendoza Diaz, R. Meier, D. A. Trytten, and S. Yoon Yoon, “Computational Thinking Growth during a First-Year Engineering Course,” *Proc. - Front. Educ. Conf. FIE*, vol. 2020-October, 2020.

[36] B. Kim, T. Kim, and J. Kim, “Paper-and-pencil programming strategy toward computational thinking for non-majors: Design your solution,” *J. Educ. Comput. Res.*, vol. 49, no. 4, pp. 437–459, 2014.

Appendix A

Article Title	Publication	Authors	Year	Context	Objective/Hypothesis	Framework	Type of assessment
A multifaceted students' performance assessment framework for motion-based game-making projects with Scratch	*Educational Media International*	Altanis, I., & Retalis, S.	2019	Computer science undergraduate students	Proposes the multifaceted assessment framework	Not CT framework mentioned	Game code analysis, in addition to an interview or journal analysis
Modeling and Measuring Students' Computational Thinking Practices in Science. Journal of Science Education and Technology.	*Journal of Science Education and Technology.*	Arastoopour Irgens, G., Dabolkar, S., Bain, C., Woods, P., Hall, K., Swanson, H., ... & Wilensky, U.	2019	A ten-day biology unit	Relationship between students' assessment scores and their responses to embedded assessment questions in the unit using discourse analytics	CT-STEM taxonomy	Test
Infusing Computer Science in Engineering and Technology Education.	*The Journal of Technology Studies*	Asunda, P. A.	2018	Four engineering and technology education teachers	(1) How do engineering and technology education teachers infuse CSP and CT into engineering and technology education? (2) How do engineering and technology education teachers assess students' CSP and CT projects that are integrated with engineering and technology education?	Not framework mentioned	Semi-structured interviews

Computational Thinking Growth During a First-Year Engineering Course.	*2020 IEEE Frontiers in Education Conference*	Diaz, N. V. M., Meier, R., Trytten, D. A., & Yoon, S. Y.	2020	Engineering students completing their first course in engineering at a large Southwestern university in the United States.	Demonstrates growth in computational thinking	They have proposed their framework	Test
First year student performance in a test for computational thinking.	*South African Institute for Computer Scientists and Information Technologists Conference*	Gouws, L., Bradshaw, K., & Wentworth, P.	2013	Introductory computer science students at a South African university.	Investigate the role that computational thinking plays in the experience of students	Gouws, K., and P. Wentworth 2013	Test
Practicing formative assessment for computational thinking in making environments.	Journal of Science Education and Technology	Hadad, R., Thomas, K., Kachovska, M., & Yin, Y.	2020	High school students interested in physics and engineering. Library in a large midwestern city in 2016.	What kinds of informal formative assessment approaches can facilitate student understanding of CT as applied to physics/engineering-based making?	Csizmadia et al 2015	Engineering notebooks, field observations, photographs, video and audio recordings Qualitative assessment.
Developing a game model for computational thinking and learning traditional programming through game-play.	*World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*	Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L	2010	Does not apply	Introduce a model for a Game-Based Learning environment	Not CT framework mentioned	Game project

Paper-and-Pencil Programming Strategy toward Computational Thinking for Non-Majors: Design Your Solution	*Journal of Educational Computing Research*	Kim, B., Kim, T., & Kim, J.	2013	Pre-service elementary school teachers, none of whom were CS majors	Does PPS improve students' logical thinking? Does PPS help students' understanding of CT? Does PPS increase students' interest in learning CS?	Wing (2008) analytical thinking	Test
A validity and reliability study of the computational thinking scales (CTS)	*Computers in human behavior*	Korkmaz, Ö., Çakir, R., & Özden, M. Y.	2017	Associate degree and an undergraduate degree with formal education in Amasya University	Develop computational thinking scale.	ISTE (2015)	Scale
Analyzing Students' Computational Thinking Practices in a First-Year Engineering Course	IEEE Access	L. M. C. Castro., A. J. Magana., K. A. Douglas., M. Boutin	2021	1559 students in a First-Year engineering course	Understand the correlation between practices related to data and computational problem-solving	Weintrop,2016	Assignments, tests, and rubrics
Paper circuits: A tangible, low threshold, low-cost entry to computational thinking	*TechTrends, 62(2), 197-203.*	Lee, V. R., & Recker, M.	2018	Library settings for teenage youth	Propose that paper circuitry provides a productive space for exploring aspects of computational thinking	Grover and Pea 2013; Kafai and Burke 2014b; Weintrop et al. 2016	Project
A first course in computing with applications to biology.	*Briefings in bioinformatics*	Libeskind-Hadas, R., & Bush, E	2013	Undergraduate biology students	Describe the organization and content of the BioComp course and some data on learning outcomes and student perceptions of the course.	Not framework mentioned	Project

An exploration of problem posing-based activities as an assessment tool and as an instructional strategy.	*Research and practice in technology enhanced learning*	Mishra, S., & Iyer, S. (2015)	2015	Two CS application courses (Data Structures (DS) and Artificial Intelligence (AI)		Brennan and Resnick (2012)	Problem posing, generation of questions by learners
A study on the impact of multidisciplinary collaboration on computational thinking.	*Proceedings of the 47th ACM technical symposium on computing science education*	Pulimood, S. M., Pearson, K., & Bates, D. C.	2016	An undergraduate software engineering course	Study the impact of collaboration in the students' perceptions of their CT abilities	Not CT framework mentioned	Self-assessment
Computational thinking development through creative programming in higher education	*International Journal of Educational Technology in Higher Education*	Romero, M., Lepage, A., & Lille, B.	2017	A total of 120 undergraduate students at Université Laval in Canada (N = 120)	This paper introduces computational thinking in the context of Higher Education creative programming activities.	#5c21 framework	Project
A Proposal for Holistic Assessment of Computational Thinking for Undergraduate: Content Validity.	*European Journal of Educational Research*	Sondakh, D. E., Osman, K., & Zainudin, S.	2020	Experts in computational thinking	Identify indicators for a holistic CT assessment instrument for undergraduate students.	Selby & Woollard, 2013 and Bocconi et al., 2016	Delphi study
Design, Complexity, and Coding: A Framework to Evaluate Games. In *European Conference on Games Based Learning*	*European Conference on Games Based Learning*	Spieler, B., & Kemeny, F.	2020	84 participants between 10 and 15 years old.	Provide an assessment template to assess game projects	Not framework mentioned	Game project

Making Computational Thinking Evident: A Validation Study of a Computational Thinking Test	*AERA Online Paper Repository*	Tang, X., Yin, Y., Lin, Q., & Hadad, R.	2018	Summer academy students	Explore the psychometric features of sample items of the Bebras test, which has been developed and used internationally since 2003	Grover & Pea, 2013	Test
An informatics perspective on computational thinking.	*ACM Conference on Innovation and technology in computer science education*	Walden, J., Doyle, M., Garns, R., & Hart, Z	2013	Majors ranging from journalism to computer science.	Developed an introductory course for students in our College of Informatics.	Researchers argue informatics is the intersection between critical thinking and problem-solving	Test