A systemic review of Problem-Based Learning (PBL) and Computational Thinking (CT) in teaching and learning

Salam*
Faculty of Languages and Literature, Universitas Negeri Makassar, Makassar, Indonesia

Abstract
Because of its educational benefits for both learners and teachers, Problem-Based Learning (PBL) and Computational Thinking (CT), teaching paradigms, and problem-solving methodologies have become widely used in teaching and learning. However, research on its integration into the teaching and learning process is still sparse, leaving educators with few guidelines for using constructive learning theory techniques. This study aims to thoroughly review and determine the country of application, learner level, PBL and CT dimensions, and learning methodologies that can lead to the successful implementation of PBL-CT. From 2017 to 2022, this study tracks relevant and published articles to identify gaps in the implementation of PBL-CT and maps their history into a taxonomy that can be used to guide future research. Limited study has yet been published that specifically describes the suitable PBL-CT framework or paradigm. Most articles deal with computing, followed by STEM (Science, Technology, Engineering, and Math) and non-STEM subjects. Some key implications can assist educators in developing effective PBL-CT lesson designs and provide some research directions for future studies. Effective PBL-CT integration improves learners' computational thinking skills while boosting teaching and learning. One of the 21st-century amplifications required for student success in CT.

© 2022 The Author(s). This is an open access article under Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license.

1. Introduction
Prior to the twentieth century, not all vocations required reading and writing skills, and some were only owned by a specific group or person, such as scholars' descendants. However, as science and technology evolve and the world civilization becomes more advanced, more people will learn this skill. In the twentieth century, the rapid adoption of computers needed the analogy of a revolution in which digital literacy has become a necessary ability for success in our complicated digital environment (Jacob & Warschauer, 2018). Shute et al. (2017). While we do not need to be software specialists, most of us use computers daily and must be able to speak with them to use our computing capabilities fully. Computational thinking is used to describe communication with computers in this situation (CT). CT has been a hot topic in education research and practice for the past decade. In searches for definitions, instructional interventions, and assessments, Google returns thousands of results. Banyak assumes that CT is related to coding or programming because of this entry, yet this assumption is incorrect.

According to the National Research Council (2010), everyone needs a CT, not just programmers. According to Cetin and Dubinsky (2017), in the data-driven era, CT skills involve managing information effectively and efficiently with technology. According to (National Research Council, 2010), a person or workforce with skilled CT skills can improve competitiveness in the global economy, such as in the US. Despite the fact that many programs say that coding abilities are uncommon, they are thoroughly understood in

* Corresponding author.
Email: salam@unm.ac.id
CT. CT is a set of behaviors entwined with a problem’s perspective that provides amplification and comprehension of CT, similar to an inquiry as a mode of thinking. However, in the sphere of Mathematics or Science, no curriculum establishes the foundation for comprehending CT for youngsters (beginners), particularly in Indonesia. As a result, various approaches are required to integrate CT information into the school’s appropriate curriculum. However, CT may be used to solve many of the difficulties presented in today’s curriculum.

Furthermore, Mehas certified CT skills could inspire kids to study computer science (Allan et al., 2010), and other STEM-related fields (Sneider et al., 2014). CT has also been linked to creativity and innovation (Selby & Woollard, 2013), and has applicability in other STEM domains (Czerkawski & Lyman, 2015). This study tries to explore the definition of CT that emerged from various disciplines and how learners can complete learning based on problems combined with CT itself (PBL-CT). To better understand ontology and our understanding of the concept, perspective, and practice of CT in PBL, this research was studied and carried out to answer the following questions:

- Why study computational thinking?
- How can that CT epistemology be implemented in PBL?
- Which dimension of CT dominates the teaching and education faculties?
- Educate prospective educators to integrate Computational Thinking into PBL

2. Literature Review

2.1. Computational Thinking (CT)

According to (Wing, 2006), Computational thinking is the conceptual foundation required to describe and solve real-world problems using algorithmic methods to obtain transferrable solutions; it is regarded as a necessary ability in a variety of scientific contexts. CT is a collection of mental methods and processes that can be used in a variety of situations, including recursive and procedural thinking, coding, abstraction, decomposition, heuristic reasoning, and parallelism. In 1980, Papert proposed the concept of CT for the first time. He believed that computational perceptions could influence how youngsters think about various topics (Papert, 1980). Then to date, the assumption is acceptable, and many researchers and agencies agree that CT is a universal skill that every devoted child must possess. Initially, only STEM (Science, Technology, Engineering, and Mathematics) fields were thought to require this skill. Nonetheless, CT is a valuable skill that may be used in subjects such as social studies, humanities, and the arts (Kalelioglu et al., 2016; Tang et al., 2020). Furthermore, the importance of growing computer science knowledge and understanding and its potential for developing more general problem-solving skills have been demonstrated.

According to Hoppe and Werneburg (2019), CT can be interpreted as part of the process of solving a problem that can be categorized into several components, including: (1) problem formulation without involving the use of a computer, (2) logical data regulation and analysis, (3) modeling, simulation, and abstraction, (4) sequencing steps in making solutions, (5) the process of identifying, analyzing, and implementing solutions appropriately, (6) the process of generalization and combination of various alternatives to a solution to a problem.

Referring to some of the definitions mentioned above, CT is simplified into several strategies in the process in the human mentality and can help learners achieve PBL goals. In addition, the effectiveness in empowering CT capabilities can also help them increase expectations of learning achievement. The third is abstraction by identifying the general principles that underlie these patterns; and fourth, algorithm design: developing step-by-step instructions for solving a range of issues.

2.2. Problem-based learning (PBL) and Computational Thinking (CT)

One of the talents that kids must understand in today's 21st-century digital era, according to the author, is problem-solving (Larson & Miller, 2011). Teachers and their students are required to have these abilities since they directly impact the learning and teaching process both inside and outside the classroom. Teachers must also be able to integrate PBL into the learning curriculum, which includes reading, writing, arithmetic, science, and social sciences. PBL is, in essence, a continuation of the meaningful and experiential learning heritage. Learners in PBL solve challenges and reflect on their experiences to learn. Because it embeds learning in real-world challenges and holds students accountable for their progress, PBL is ideal for assisting students in becoming active learners. It focuses on developing tactics and gaining knowledge (cognition) in groups.

According to Hmelo-Silver (2004), there are two important issues in PBL. The first issue is that this approach implies that learners can more actively build knowledge when collaborating in groups. The second one is the role of teachers and students can synergize means that the teacher is no longer the only main source of learning, but the teacher acts as a facilitator or collaborator. In this case, teachers can facilitate students through questions to stimulate them to think more critically and creatively and be involved in every pre-learning process; this is an indicator of PBL success.

Concerning computational thinking (CT), Hoppe and Werneburg (2019) argue that CT is a process of solving the period which includes several indicators: CT can be interpreted as part of the process of solving a problem that can be categorized into several components, including (1) problem formulation without involving the use of a computer, (2) data management and analysis that logical, (3) modeling, simulation, and abstraction, (4) sequencing steps in solution making, (5) the process of identifying, analyzing, and applying solutions appropriately, (6) the process of generalization and combination of various alternatives to a solution. The process of solving problems is often described as seven stages of the cycle (Pretz et al., 2003): (a) the recognition or identification of the problem, (b) the definition
and repression of the mind of the problem, (c) the balance of strategies to resolve the animate period, (d) the organization of knowledge about the problem, (e) the allocation of resources mental and physical to solve the problem, (f) monitoring the progress achieved, and (g) evaluation of accurate solutions. Although there may be several ways to solve the problem, and the chosen one will depend on the nature of the problem, it is assumed that they will all have the same proses. However, what is particularly important in this PBL-CT process is solving the problem by first carrying out the process of recognizing and identifying to determine the best solution.

3. Method

A systematic literature review is a method of locating, analyzing, and interpreting research relevant to the study question, topic area, or phenomena of interest. Primer studies refer to personal research contributions to systematic reviews, while secondary studies refer to systematic reviews (Keele, 2007).

3.1 Search strategy

The method used in this systematic review of the literature refers to (Keele, 2007). The process in the guide consists of several stages, for example, the formulation of research questions, search strategies (inclusion and exclusion), conducting a primary study selection process, data extraction, and synthesis, and recording results and analysis. During the search of related articles in this study, six main sources or reputable databases were considered relevant, namely ProQuest, Scopus, ScienceDirect, ERIC, Taylor & Francis, ACM (Association for Computing Machinery), and IEEEExplore. But in addition to that, researchers also use the Google Scholar search engine as the basis for searching for related articles, and the scheme looks like in figure 1.

![Figure 1. Article browsing scheme](image)

3. Results and Discussion

From the results of the search and feasibility study, the results of the literature review are presented as follows:

3.1. Why is it Important to Teach Computational Thinking to learners?

To respond to research question number 1 above, the study results noted that CT is crucially taught to learners and not only that but also in prospective teachers (pre-service) (Chang & Peterson, 2018). There is some urgency because teachers crucially teach CT to learners in the 21st-century era. Given the prevalence of computational artifacts in society and our daily lives, learners are increasingly expected to solve problems using a computational approach. Meanwhile, analysts project that by 2030, up to 800 million jobs will be mechanized due to advances in automation and artificial intelligence. Learners must think computationally and navigate numerous levels of abstraction to develop new answers to perplexing situations to succeed in this dynamic workforce. Beyond workforce readiness, starting with learners to develop CT skills supports civic involvement, allowing students to contribute as academics who increasingly use computing to benefit humanity's well-being.

CT abilities, according to experts, have such a broad impact on communication and social interaction that they are considered essential literacy. Despite these beliefs, educational institutions continue to emphasize the development of reading, writing, and mathematics abilities rather than CT skills. By explicitly integrating the aims of computational thinking into the core field of study curriculum, educational policy initiatives can update present pedagogical practice and stress these skills and dispositions. While initiatives to teach computational thinking in high school have shown promise, computational thinking has been muddled in tackling context problems in recent years. A transformation from an unsatisfactory initial condition to a desired final one by overcoming a barrier can be described as solving the problem. As a result, they are diverted from the challenge of teaching additional competencies such as sophisticated thinking and flexible problem solving to place learners in the greatest possible starting position for engagement in work, life, and society.

Regarding language instruction, Jacob et al. (2018) claim that CT can be taught to students using a scaffolding linguistic method. Jacob and Warschauer (2018) investigate the importance of computational thinking in the English Language Arts curriculum and the linguistic and sociocultural processes that underpin English learners' success in mastering computational thinking. The establishment of a first-year curriculum based on the ideas of language scaffolding and culturally responsive pedagogy stated below has been effectively implemented by teachers. Explicit vocabulary lessons are scaffolded in an agile manner. Guru employs a range of tactics to expose learners to the concept of computational thinking, including performing explicit instructions using vocabulary cards in the introduction lesson when the learner is experimenting with basic concepts for the first time. Before restating the target vocabulary, the teacher models it in common language and engages learners in exploratory activities in the programming environment. Create a vocabulary card medium that may be used as the primary source of information and reference for ongoing learning activities or cover a variety of topics.

Teachers can use growing literacy trends in collaborative PBL to instill computer science and language goals into every academic topic and establish a linguistic framework for learners' computational comprehension. When learning
computational thinking, students employ a variety of colloquial language to describe topics, negotiate code interpretations, and propose alternate solutions. The framework was created to accommodate three different degrees of language proficiency: emerging (low), developing (mid), and bridging (high) (high). The linguistic framework is theoretically founded on the concept of systemic functional linguistics, which asserts that language is intimately linked to its social context and that the process of language meaning-making is continually responding to changes in human interaction (Halliday, 1973). Furthermore, the relationship between language and programming implies that both rely on syntactic sequences and social interactions among speakers/programmers to produce meaningful constructs (Grover & Lyytinen, 2015). In practice, teachers might model sentence frames for students and use props such as flashcards to inspire students to pursue higher-level academic goals. Teachers can use CT to monitor and promote literacy abilities that surface in individual reflections and group conversations about learner programs, making the linguistic framework ideal for formative assessments.

3.2. How can that CT epistemology be implemented in PBL?

Unlike traditional learning methods, Problem-Based Learning (PBL) is learner-centered. By overcoming some open problems, learners understand and master the basic content of the learning material. The purpose of learning is not to focus on solving a specific problem but on the skills and attributes that learners acquire by solving these problems. PBL allows learners to solve problems by observing and understanding real-world experiences through an active learning process. The benefit of solving this problem is to make learners acquire the skills and attributes they need. Skills and attributes include knowledge acquisition, teamwork, and communication. It is important to note that the goal of PBL is not knowledge itself but rather a universal learning model. The findings of this literature review suggest that problems in PBL essentially have the feasibility of meeting the following conditions: (a) Generalizations: learners can obtain a common approach by completing tasks, (b) Moderate difficulties: too simple questions cannot improve learners' learning ability. On the contrary, overly complicated questions beyond the learner's ability to learn, (c) Decomposability: the learner can reduce the difficulty of the problem, and this decomposition is convenient for the learner to work together to solve the problem, (d) Independence: The problem should not depend too much on a certain background, so that the learner cannot pay attention to the key part.

In the implementation of PBL, the process of the learning experience is obtained through investigation, explanation, and solving of problems appropriately. In this case, students collaborate to learn things that need to be known, including looking for information to solve problems mutually agreed upon. Meanwhile, the teacher acts as a facilitator to guide a student through the learning cycle, as depicted in figure 1. In this cycle, a PBL tutorial process, learners are presented with problem scenarios. They formulate and analyze the problem by identifying the relevant facts of the scenario. This number of fact identification helps learners present the problem. Students are expected to have a good understanding of the problem to produce hypotheses about possible solutions. An important part of this cycle is identifying the lack of knowledge relative to the problem. This knowledge and quantification become what is known as a learning problem that learner’s study during their self-directed learning (SDL). After SDL, learners apply their new knowledge and evaluate their hypotheses based on what they learned. After each problem, the learner reflects on the abstract knowledge gained. In this process of abstraction, the computational ability to think arises within the learner's cognition. Simultaneously teachers can intervene in them cognitively because the solution to solving the problem is expected to be varied. In short, it can be seen through the scheme in figure 2 below.

![Figure 2. CT cycles in PBL](image_url)

When administering pre-and post-tests to learners aged 14 and 15 in Israel, researchers such as Abdu and Schwarz (2020) focused on integrating CT in PBL at various stages of education spanning from elementary school to college. They were invited to work together using argumentation framework strategies to solve related mathematical billiard model problems, identifying entire claims and arguments stated by group members. After the post-test, they were able to investigate six argumentative variables (claims, arguments, valid claims, valid arguments, collaborative claims, and valid
collaborative arguments) throughout ten tests, with valid claims produced being significantly correlated with cooperation (Chen, 2017; de Jesus & Silveira, 2021).

3.3. How to educate educators to integrate Computational Thinking in PBL

Based on the research results that we summarized in this literature study, CT was also effectively taught by educational practitioners, instructors, and teachers, both beginners, and professionals. This is because students who have established CT skills in their simultaneous time can also solve problems in their learning process. As stated by Chang and Peterson (2018), Falcão and de França (2021), and Sands et al. (2018) that there are nine concepts in core CT namely data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation. They also made it clear that CT has the potential to significantly advance learners' problem-solving skills and abilities as they begin to think in new ways. Zapata-Cáceres et al. (2020) argue that learners need to learn CT early and often, emphasizing computational understanding processes and not on their manifestation in specific programming languages and the skills to abstract and represent information. To achieve this goal, they propose the use of computational thinking language to combine computational concepts in the core content area. CT languages are not programming languages but vocabulary and symbols that can help computational thinking penetrate other content areas at the K-12 level. It is important to equip all teachers with adequate computational knowledge thinking and how to incorporate computational thinking into their disciplines (Grover & Pea, 2013; Kong & Lai, 2021; Yadav et al., 2014).

One way to approach this task is to start with prospective teachers (pre-service), novice teachers, and senior teachers (professional teachers). Train novice teachers in computational thinking concepts at the beginning of their teacher preparation will allow them to see the relevance of CT in their discipline. The interesting thing is Yadav et al. (2014) the synthesis of the concept CT has been used in other disciplines through the process of solving problems, and computational thinking ability is not important for any discipline, for example, biology, journalism, finance, and archaeology. Furthermore, teachers in education must integrate CT with reading skills, writing, and arithmetic. But before that, aspiring teachers must internalize how to think computationally first so that they can transfer and engage their learners in a more meaningful understanding of CT. This becomes very crucial in relation to the subjects they study based on the applicable curriculum (González & Muñoz-Repiso, 2017). For example, one aspect of CT is abstraction. The teacher can apply this skill in giving instructions or instructions in class discussions or assigning project assignments to his learners. In addition, to get a more in-depth, meaningful, and effective understanding of CT, teachers also need to be included in various training or workshops on integrating technology pedagogically with CT (Technological Pedagogical Content Knowledge / TPACK) (Butarbutar, 2021a; Butarbutar et al., 2021b; Mouza et al. 2017). These two important aspects contribute to producing learners with a high level of thinking (High Order Thinking/HOT) (Zaharin et al., 2018). Thus, the teacher will be even more qualified to set the right learning methods, approaches, and strategies by the character of their students in the classroom. From this research, the author summarizes that TPCK, CT, and HOT are three potential essences that students must have in the 21st century and help them improve PBL skills in the learning process.

3.4. What is the very dominating dimension of CT in the teaching faculty?

The findings of the researchers that were successfully noted in this literature study that CT is categorized into several dimensions, namely: (1) Formulation of a problem or redesigning a problem into a problem that can be solved appropriately, (2) recursion, namely building a system gradually based on previous information, (3) decomposition of problems means deciphering the problem to be in manageable units, (4) abstraction, i.e., model core aspects of a wrong major complex system, (5) systematic testing is taking of actions aimed at obtaining solutions. People take useful input (and discard unnecessary data) from complex systems to build patterns and uncover commonalities among diverse representations, which is the key ingredient underlying CT (Wing, 2008). Wing (2011) revealed that abstraction includes layers, so each layer must be defined, and their interactions must be clarified. It entails three steps: (a) layer-by-layer abstraction, (b) overall abstraction, and (c) layer-by-layer interconnection. Define an algorithm, for example, is a type of abstraction and abstraction of a step-by-step technique for receiving inputs and producing the desired output (Wing, 2008). According to Barr and Stephenson (2011), CT includes data organization and analysis, automation, efficiency, generalization, and the abstraction and reformulation of the problem. Automating a process or system, optimizing a solution, and generalizing a CT method to address a new challenge are all examples of automatization. Certain characteristics critical for CT, such as self-confidence, tenacity in accomplishing complex problems, and the ability to work well in a team, were also included in Barr and colleagues' study. Abstraction, generalization, and trial and error behaviors are part of the CT outlined by Bers et al. (2014). They stress the significance of debugging (identifying and correcting errors when the solution does not work as expected). According to the National Research Council's thorough research, CT consists of five important and universal parts across domains: hypothesis testing, data management, parallelism, abstraction, and debugging. In order to understand how a system works, one must build and test hypotheses methodically while solving difficult problems in any discipline. Because it is impossible to test every possible scenario, picking the correct criteria to test is crucial. Data management is gathering information from a variety of sources, analyzing trends in the data, and understandably presenting the information. The simultaneous processing of data from multiple sources or dimensions is referred to as parallelism. Abstraction is concerned with simulating the operation of a complicated problem or system. Finally, debugging is the process of locating and correcting flaws in a
model after it has been built. Anderson (2016) has described the five components of CT: (1) issue decomposition, (2) pattern recognition, (3) abstraction (i.e., iterative pattern generalization), (4) solution algorithm design, and (5) solution evaluation (debug). According to the findings of this study’s article, search, decomposition, abstraction, algorithms, and debugging are the most prevalent dimensions in the faculty of teacher training and educational sciences (Kalelioglu et al., 2016).

4. Conclusion

Based on the findings and discussions above, the researcher summarized that supporting the computation of thought processes and competencies is considered a part of yang oriented towards the Future of school education and adding it to learning. However, developing an understanding of computational thinking that leads to facilitating teaching in schools will be very challenging. When various perspectives on computational thinking and its relevance to school learning exist, the limitations of empirical knowledge are the obstacles. Furthermore, over the past three years, there has been a transformation in integrating technology into schools’ curricula ranging from elementary to tertiary levels. In other words, this transformation implies that teachers and learners are obliged to have computational thinking competencies because their existence is in the era of the 21st century.

Ultimately, when learners want to transform the abstraction of information and the learning experience with new knowledge, they simultaneously need CT, collaborative PBL, and TPACK. This skill serves to decipher the exercises, making the algorithmic pattern a sequence of learning design procedures simpler and more systematic. When teachers integrate PBL-CT, in this case, the role of the teacher has the potential to adjust the type of approach or strategy that is relevant to the level of learners. This study suggests several effective strategies such as collaborative or cooperative, scaffolding, storytelling, game-based PBL, group work, simulation, blended learning, trials, discussions, and so on, but the most important is the game-based collaborative approach to solving certain problems.

CT has been introduced in the national standards for schools; hence consequently, teachers must learn about CT to know how to integrate it into the teaching of their field of study. Furthermore, the development of CT in prospective teachers, beginners, and senior teachers is also very important to provide extra training to increase their insights so that they can also teach students in this 21st-century era where they are very adaptive and familiar with technological literacy (ButarButar & Simatupang, 2020a; Butarbutar et al., 2020b). However, teachers should also not be left behind. Therefore, students, teachers in schools, curriculum makers, and office holders must synergize. Two techniques need to be carried out in adapting to skills in this century, namely self-regulated learning and lifelong learning for both students and teachers. Teacher education programs should accelerate the inclusion of CT in the curriculum, considering not knowledge for subjects but also pedagogical knowledge dan how to apply it in schools.

References

Abdu, R., & Schwarz, B. (2020). Split up, but stay together: Collaboration and cooperation in mathematical problem solving. Instructional Science, 48(3), 313–336.
Allan, V., Barr, V., Brylow, D., & Hambrusch, S. (2010). Computational thinking in high school courses. Proceedings of the 41st ACM Technical Symposium on Computer Science Education, 390–391.
Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48–54.
Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. Computers & Education, 72, 145–157.
Butarbutar, R. (2021). Learner’s perception of task difficulties in technology-mediated task-based language teaching. English: Journal of Language, Education, and Humanities, 9(1), 129–144.
Butarbutar, R., et al., (2021). Challenges and Opportunities of Accelerated Digital Literacy during the COVID-19 Pandemic. Hong Kong Journal of Social Sciences.
ButarButar, R., & Simatupang, E. (2020). The Impact of Technology Hello English Application in EPL Classroom. Lingual: Journal of Language and Culture, 8(2), 11.
Butarbutar, R., Uspayanti, R., Bawawa, M., & Leba, S. M. R. (2020). Mobile Assisted Language Learning. 3rd International Conference on Social Sciences (ICSS 2020), 390–392.
Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. The Journal of Mathematical Behavior, 47, 70–80. https://doi.org/10.1016/j.jmathb.2017.06.004
Chang, Y., & Peterson, L. (2018). Pre-service teachers’ perceptions of computational thinking. Journal of Technology and Teacher Education, 26(3), 353–374.
Chen, G. (2017). Programming language teaching model based on computational thinking and problem-based learning. Proceedings of the 2017 2nd International Seminar on Education Innovation and Economic Management (SEIEM 2017).
Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. TechTrends, 59(2), 57–65.
Falcão, T. P., & de França, R. S. (2021). Computational Thinking Goes to School: Implications for Teacher Education in Brazil. Revista Brasileira de Informática Na Educação, 29, 1158–1177.
González, Y. A. C., & Muñoz-Reposo, A. G.-V. (2017). Development of computational thinking and collaborative learning in kindergarten using programmable educational robots: A teacher training experience. Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, 1–6.
Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38–43.
Grover, S., & Lyytinen, K. (2015). New state of play in information systems research. MIS Quarterly, 39(2), 271–296.
Halliday, M. A. K. (1973). Explorations in the functions of language.
Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? Educational Psychology Review, 16(3), 235–266.
Hoppe, H. U., & Werneburg, S. (2019). Computational thinking—More than a variant of scientific inquiry. Computational Thinking Education, 13–30.
Jacob, S., Nguyen, H., Tofel-Grehl, C., Richardson, D., & Warschauer, M. (2018). Teaching computational thinking to English learners. NYS TESOL Journal, 5(2).
Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. Journal of Computer Science Integration, 1(1).
Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. Baltic Journal of Modern Computing, 4(3), 583.
Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Ver. 2.3 EBSE Technical Report Report. EBSE.
Kong, S.-C., & Lai, M. (2021). A proposed computational thinking teacher development framework for K-12 guided by the TPACK model. Journal of Computers in Education, 1–24.
Laron, L. C., & Miller, T. N. (2011). 21st Century Skills: Prepare Students for the Future. kappa Delta Pi Record, 47(3), 121–123. https://doi.org/10.1080/00228958.2011.10516575
Magno de Jesus, Â., & Silveira, I. F. (2021). Game-based collaborative learning framework for computational thinking development. Revista Facultad de Ingeniería Universidad de Antioquia, 99, 113–123.

Mouza, C., Yang, H., Pan, Y.-C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). Australasian Journal of Educational Technology, 33(3).

Papert, S. (1980). "Mindstorms" Children. Computers and Powerful Ideas. Pretz, J. E., Naples, A. J., & Sternberg, R. J. (2003). Recognizing, defining, and representing problems. The Psychology of Problem Solving, 30(3).

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In Computational thinking in the STEM disciplines (pp. 151–164). Springer.

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition.

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. The Science Teacher, 81(5), 53.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. Computers & Education, 148, 103798.

Wing, J. (2011). Research notebook: Computational thinking—What and why. The Link Magazine, 6, 20–23.

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

Zaharin, N. L., Sharif, S., & Mariappan, M. (2018). Computational thinking: A strategy for developing problem solving skills and Higher Order Thinking Skills (HOTs). Int. J. Acad. Res. Bus. Soc. Sci, 8(10).

Zapata-Cáceres, M., Martin-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. 2020 IEEE Global Engineering Education Conference (EDUCON), 1905–1914.