**INTERNATIONAL JOURNAL OF
EDUCATION, PSYCHOLOGY
AND COUNSELLING
(IJEPC)**www.ijeipc.com**REVIEW OF COMPUTATIONAL THINKING MODELS IN
VARIOUS LEARNING FIELDS**Subashini Annamalai^{1*}, Azizah Che Omar², Sobihatun Nur Abdul Salam³¹ School of Multimedia Technology and Communication, Universiti Utara Malaysia (UUM), Malaysia
Email: subashini@uum.edu.my² School of Multimedia Technology and Communication, Universiti Utara Malaysia (UUM), Malaysia
Email: cheomar@uum.edu.my³ School of Multimedia Technology and Communication, Universiti Utara Malaysia (UUM), Malaysia
Email: sobihatun@uum.edu.my

* Corresponding Author

Article Info:**Article history:**

Received date: 16.10.2022

Revised date: 07.11.2022

Accepted date: 22.12.2022

Published date: 31.12.2022

To cite this document:

Annamalai, S., Che Omar, A., & Abdul Salam, S. N. (2022). Review of Computational Thinking Models in Various Learning Fields, *International Journal of Education, Psychology and Counseling*, 7 (48), 562-574.

DOI: 10.35631/IJEPC.748042This work is licensed under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)**Abstract:**

CT has recently sparked a lot of research in a variety of disciplines. Many researchers have attempted to define CT and published articles on the subject and its model. The developed models show a dependency of dimensions. As a result, a review of 14 computational thinking models has been completed for this article. The goal is to combine the many elements and therefore contribute to a common understanding of words. The authors identified the most often used terms in CT definitions and scope, culminating it in the CT dimensions category. The results of this study may be beneficial not only in the investigation of CT research subjects and the identification of CT in the literature but also in present and future attempts to apply CT in diverse settings and aims.

Keywords:

Computational Thinking, Model, Framework, Review, Dimension

Introduction

In response to the rapidly evolving digital age in which we live, schools are placing a renewed emphasis on preparing students to engage successfully in a 21st-century society dominated by digital technology. Computational Thinking (CT) is an emerging concept in the Industrial Revolution 4.0 (IR4.0) and has been introduced to school curriculum across the world as a necessary ability, equivalent to reading and writing (Aranda & Ferguson, 2018). Academic scholars and technology researchers have recently advocated for the integration of CT abilities into formal education settings. This interest has led to the development of new curriculum and technologies that allow students to engage in computational thinking in the classroom. (Weintrop, Holbert, Horn, & Wilensky, 2016).

CT is a thought process that is used to formulate a problem and identify a solution so that the answer may be translated into a form that an information processing agent can successfully implement (Selby, 2015; Wing, 2011). However, the majority of the research has described CT as being related to computer science education and how it should be integrated into computer science education (Lee et al., 2011). Many research has been conducted to determine what skills require CT and what tools and approaches may be utilized to assist students in computer science education (Berland & Lee, 2011).

Nevertheless, CT has primarily grown out of the computer science education community and has been implemented in broad applicability (Weintrop, Holbert, Horn, & Wilensky, 2016). The implementation should emphasize its potential benefits as a support tool to foster student motivation and abilities in problem-solving (Kazimoglu, Kiernan, Bacon & MacKinnon, 2012). Due to the multiple benefits, CT offers to provide at all educational levels, for teaching and learning processes across various disciplines, the notion of CT is gaining greater acceptance. Many studies have linked computational thinking to a fantastic tool for enhancing students' interest in Science, Technology, and Engineering (STEM) courses at all levels of education.

The National Research Council (2010) underlined the need of teaching CT to students as early as possible, as well as aiding them in understanding how to use these critical abilities. CT's increasing prevalence and potential are apparent since it pervades nearly every subject, including science and the humanities (Mohamed Zaki, Wong & Yaakob, 2019). Thus, the purpose of this article is to highlight the existing frameworks, or models, of CT and its dimensions that have been adopted in diverse educational contexts as a valuable reference for present and future attempts to apply CT in various settings and purposes.

Computational Thinking (CT)

Computational Thinking (CT) is defined in a variety of ways in the literature, but it all stems from a phrase coined by Papert (1996) and popularised by Wing (2006) in a landmark work. It has gained a lot of momentum in recent years. Although there is no agreed-upon definition of computational thinking (Voogt, Fisser, Good, Mishra, & Yadav, 2015), it builds on the thought processes involved in formulating problems and expressing their solution(s) in a way that a computer-human or machine – can effectively carry out (Voogt, Fisser, Good, Mishra, & Yadav, 2015; Wing, 2011).

Computational thinking can be characterized as the process of breaking down a complex problem into numerous smaller problems that we can solve. According to Wing (2006), it is a process that involves using core computer science ideas to solve problems, build systems, and understand human behavior. It is also chosen as a thinking process, and it is used to outline a problem and discover a solution so that the answer may be translated into a form that an information processing agent can execute successfully (Selby, 2015; Wing, 2011). The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) (2011) defined CT as a problem-solving process involving the use of a computer and other equipment that includes (but is not restricted to) the skills of logically managing and analyzing data; representing data through abstractions such as models and simulations; and implementing solution through algorithmic thinking (a series of ordered steps); identifying, analyzing, implementing viable solutions to obtain the most efficient and effective combination of actions and resources; generalizing and translating this problem-solving practice.

Barr and Stephenson (2011), in their research, stated that CT is a method of addressing issues that can be accomplished using a computer. CT is a problem-solving approach that may be automated and used in a variety of disciplines. This was supported by Google (2016) which expound CT as the capacity to confidently cope with complexity and open-ended issues as a part of a problem-solving process that incorporates a variety of qualities, such as logically arranging and analyzing data and generating solutions using a sequence of organized stages (or algorithms). According to Chao (2016), CT entails a visual problem-solving environment that allows for the development of programming abilities while also providing technical help for solving computational problems. In Malaysia, CT is defined as a problem-solving method that is carried out either by humans or machines, or both and is based on a fundamental idea in computer science (Romainor, Abdul Talib & Abd Hakim, 2018). Malaysia's Ministry of Education (MOE) has partnered with CT is a problem-solving method that involves applying logic to the nature of the problem, gathering data, and following a consistent procedure to get the desired outcome. As a result, CT is a collection of cognitive abilities and a method for creating and solving problems, with or without the use of a technical instrument, and presented in an understandable manner.

Computational Thinking (CT) in Various Learning Field

As students learn to think in new ways, computational thinking has the potential to significantly improve their problem-solving skills and talents. Students should learn computational thinking early and often, with an emphasis on comprehending computational processes and abilities for abstracting and expressing information rather than programming or computer science (Lu and Fletcher, 2009). In addition, Hemmendinger (2010) highlights that the purpose of teaching computational thinking is to help educate students on how to think like a business person, a technologist, or an artist, and to fully comprehend how to use computation to resolve issues, generate, and discover new questions that can be constructively explored, rather than to teach everyone to think like a computer programmer.

Therefore, CT can be referred to as a method for generalizing and transferring the problem-solving process to a wide range of situations, such as confidence in dealing with uncertainty, the capacity to interact with open-ended problems, and the ability to communicate and collaborate with others to achieve a common goal or solution (Barr, Harrison, and Conery,

2011; Shute, Sun & Clarke, 2017). Computational thinking has an impact on how students approach and solve issues when it is used across multiple topic areas. Computational thinking aids the success of problem-solving by giving a variety of approaches to issues (McClelland & Instructional, 2018).

Although it is assumed that computing, science, and mathematics are the most prominent domains for computational thinking, it can be found in any curriculum. This is owing to the recognition that computational thinking is merely a problem-solving skill applicable to many disciplines, and that it may be taught as part of a subject area or as a separate skill (Czerkowski, 2016). According to Barr, Harrison, and Conery (2011), incorporating computational thinking into the curriculum across subject areas, allows the students to learn these essential skills in a non-traditional manner, allowing them to internalize them and make it normal for them to relate the knowledge across subject matter and apply skills in different contexts.

Deschryver and Yadav (2015) go even further, arguing that “new literacies and computational thinking are needed to encourage creative thinking” across disciplines to bridge the gap between traditionally creative content areas (music, art, and literature) and scientific fields (math, science, engineering). Foundational abilities may be created through integrating learning activities utilizing jointly defined literacies and the inclusion of computational thinking skills to assist scaffold learning and encourage creative thinking among learners, avoiding limited interpretations and methods of learning (McClelland & Instructional, 2018). Computational Thinking (CT) abilities should be included in formal education, according to education academics and computer scientists (McClelland & Instructional, 2018). This excitement has led to the development of new curriculum and technologies that allow the students to participate in the implementation of computational thinking in the classroom (Weintrop, Holbert, Horn, & Wilensky, 2016). Incorporating CT in the teaching and learning process will:

- i. help students better to find new solutions to problems that seem impossible to solve
- ii. help the teachers to improve the teaching and learning activity
- iii. enrich the teaching process and the student's exploration of the subject
- iv. enhance the students' confidence, especially for dealing with the unclear, complex, or open/wide (open-ended) problems

(Weintrop, Holbert, Horn, & Wilensky, 2016)

This integration is divided into two approaches, namely, the transfer of skills CT to students (Basawapatna, Koh, Repenning, Webb, & Marshall, 2011; Chang, 2011) and the synthesize of CT in teaching (Angeli et al., 2016; Sengupta, Dickes & Farris, 2018). Therefore, in order to produce the intended outcome of CT, the students must be exposed and allowed to master the CT skills through a variety of methods applied in various subjects. Hence, these can be done by including CT in the curriculum and it is effectively reached the students. One of the best approaches to integrate CT into the instructional process in various learning fields is by using the Game-Based Learning (GBL) tools (Werneburg, Manske, & Hoppe, 2018).

Review of Models

There have been a few models and frameworks related to the computational thinking model in education. To shed a little light on the models/frameworks and their components, this article describes the summary of the review done on the dimensions and its definition that forms each model/framework. The primary concept of the review is to take textual information and analyze, simplify, and summarise it based on pre-determined or emergent topics.

This article describes the findings of reviewing relevant literature for fourteen (14) models related to CT in education. A literature review model by Kitchenham (2004), was adapted as the guideline to do this review. It was chosen because it allows the researcher authors to analyze and interpret all published evidence relevant to a particular research question, subject matter, or object of interest, and it does not consider the impact of the review procedures or specify the mechanisms needed to conduct a meta-analysis in depth. The guideline covers the three steps of a systematic review: planning, conducting, and reporting on the review.

For this purpose, the electronic databases of Scopus, Science Direct, EBSCOhost, ERIC, and Web of Science were searched for literature and articles regarding computational thinking models. The articles selected were from the areas of Science, Technology, and Mathematics (STEM). The keywords such as "computational thinking," "definition," "model," "review," "STEM," "Mathematics," and "dimensions" were used to search the electronic database indexes. In addition, for the database search, other combinations of these keywords were used, as well as the "AND" and "OR" commands. For the review, articles that were within the scope of the systematic scanning criteria were chosen. The authors concentrated mostly on the relevant research papers written by the model developer. The authors consulted other works of literature that described the models for those models where no formal criteria were provided. The authors reviewed articles that emphasize the model from the year 2006-2020. The authors only reviewed articles that are available in English. When reviewing the publications, the authors chose the following criteria of interest (where relevant) to provide an overview of models and their dimensions.:

- Definition of Computational Thinking (CT)
- Dimensions involved
- Subject: Computers Science or STEM

The articles and models that don't fit into the following criteria of the reviews were excluded:

- i) Articles published in a peer-reviewed journal
- ii) Written in the English language
- iii) Accessible in full
- iv) Studies focused on computational thinking and using any educational technology approaches such as mobile technologies or online classrooms

The findings of this study are significant because they give evidence of the strengths and drawbacks of various models, as well as an additional foundation for discussions in the literature. Furthermore, it is anticipated that the outcomes of this study will lead to actual recommendations for promoting computational thinking. The review procedure is depicted in Figure 1.

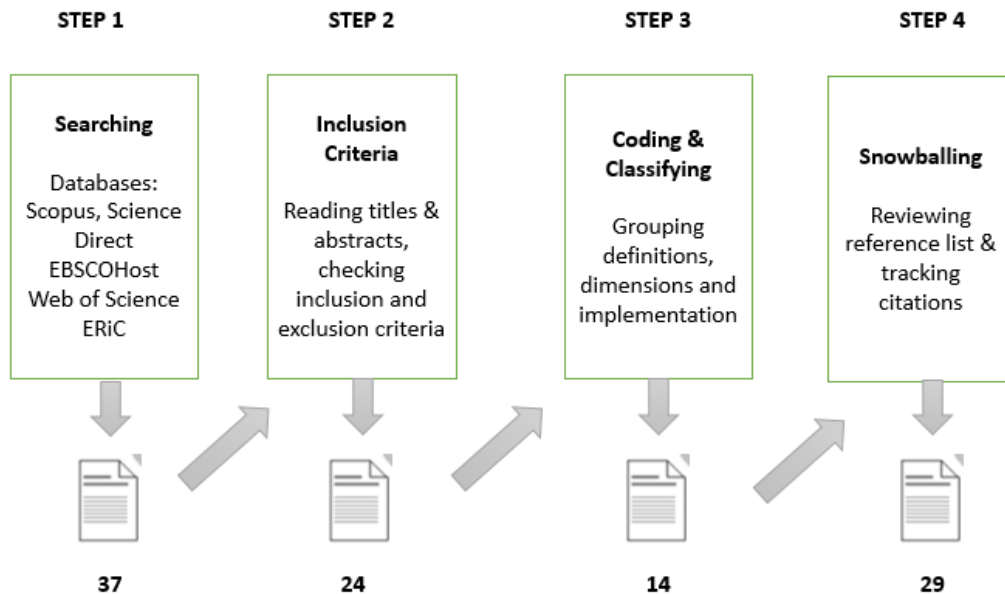


Figure 1: The Review Process of CT Models

Findings and Discussion

In this review, we analyze scholarly articles on the models and frameworks of computational thinking. By doing so, we investigated the definition of CT by the researchers, model design, dimensions, and the constructs of the models. This section discusses the findings regarding the summary of the CT model review. Table 1 describes the summary of the review done on 14 models of computational thinking from all domains.

In analyzing the 14 models found in the literature review, it appeared that they are often based on each other. Therefore, it is easier to categorize them based on the definition and dimensions of CT used in each article. The results of the review analysis of the models show a dependency of CT the dimensions between the models from the Wing (2006), National Research Council (NRC) (2010), Barr and Stephenson (2011), CSTA and ISTE (2011), Brennan and Resnick (2012), Kazimoglu, Kiernan, Bacon, & MacKinnon (2012), Selby and Woollard (2013), Csizmadia, Curzon, Dorling, Humphreys, Ng, Selby, & Woollard (2015), Anderson (2016), Atmatzidou & Demetriadis (2016), Kalelioglu, Gulbahar, & Kukul (2016), Shunte, Sun & Clarke (2017), Dong, Cateté, Jocius, Lytle, Barnes, Albert, Joshi, Robinson, & Andrews (2019), and Palts & Pedaste (2020).

Based on the comparison, the common CT dimensions are abstraction, decomposition, debugging/evaluation, algorithms, and generalization. The following is the general description of the CT dimensions derived from the models.

- i. Abstraction is the process of making an item more understandable by removing superfluous information and lowering the number of variables, resulting in simpler solutions. It is the ability to determine what information on a person, thing, or item should be kept and what should be ignored (Wing, 2006).

- ii. Decomposition is a technique for dismantling issues and breaking them down into smaller, more manageable components. "Divide and Conquer" is another name for this strategy. It is the ability to break down a difficult problem into smaller, easier-to-understand, and solve components (NRC, 2010).
- iii. Algorithms formation is the process of putting together a set of steps that may be followed to solve all of the difficulties that make up the original problem. It is the ability to create a step-by-step series of operations/actions to solve an issue (Selby & Woollard, 2015).
- iv. Debugging/Evaluation is a method of analyzing (and identifying) processes and outputs in terms of efficiency and resource usage. It is the ability to recognize, eliminate, and correct mistakes (Selby & Woollard, 2015).
- v. Generalization is the act of extending stated solutions or algorithms to a variety of problem situations, even when the variables are different. It's the ability to construct a solution in broad terms that may be applied to a variety of issues (Selby & Woollard, 2015).

This study also managed to categorize the dimension terms from all the reviewed models according to their function. Figure 2 describes the categories of the terms.

Problem detection	Data collection, representation, and analysis	Solution generation, selection, and implementation	Examine the outcome	Induction
<ul style="list-style-type: none"> • Problem formulation • Identifying problem • Defining Problem • Abstraction • Incremental & iterative • Hypothesis testing • Problem solvin 	<ul style="list-style-type: none"> • Recursion • Decomposition • Analyzing problem • Logical reasoning • Data management • Pattern recognition • Data gatering, representing & analysing 	<ul style="list-style-type: none"> • Abstraction • Solving problem • Algorithms • Automation • Parallelization • Simulation • Automation • Reusing & remixing • Generate, select & plan solution 	<ul style="list-style-type: none"> • Systemic testing • Evaluation • Debugging & iteration • Implement solution 	<ul style="list-style-type: none"> • Generalization • Transferring • Modularity • Assessing solution & continue for improvement • Socializing

Figure 2: Categories of the CT Dimensions.

Conclusion

The findings of this review are still in the early stages of development, and the researchers are still looking for additional CT models and frameworks in many areas. Furthermore, no clear difference exists between the dimensions and constructions. CT, according to the experts, may be utilized in a variety of areas, not only computer science. Students will learn to apply and integrate their abilities inside the diverse subject matter, to broaden their thinking beyond the apparent solution, irrespective of the subject, by incorporating CT in a range of disciplines. The integration of CT in teaching and learning process will help the teachers to improve the teaching and learning activity and enrich the teaching process and the students' exploration of the subject. Students will become better in findings new solutions to problem and enhance their confidence, especially for dealing with the unclear, complex or open / wide (open-ended) problems. Hence, these can be done by including CT in the curriculum and it is effectively reached the students. One of the best approach to integrate CT in the teaching and learning process is by using the Game Based Learning (GBL) tools.

Table 1: Summary of CT Model review

Authors	Year	Definition of Computational Thinking	Constructs/ Dimensions/ elements in the model/ framework																						
			Problem reformulation/ Identifying problem	Analysing data	Recursion	Decomposition	Abstraction	System testing/Evaluation/ Debugging/ Assessing	Defining Problem	Solving problem	Analysing the solution	Algorithms/ procedures	Generalization	Iteration	Automation	Parallelization	Simulation	Reusing & Remixing	Hypothesis testing	Pattern recognition	Logical reasoning	Modularity	Generate, select & Plan solution	Implement Solution	Socializing
Wing	2006	Way of solving problems, designing systems, and understanding human behavior by drawing on the concepts of computer science.	√		√	√	√																		
NRC	2010	A broad range of mental tools and concepts from computer science that help people solve problems, design systems, understand human behavior, and engage computers to assist in automating a wide range of intellectual processes		√			√	√							√			√							
Barr & Stephenson	2011	An approach to solving problems in a way that can be implemented with a computer.				√	√				√			√	√	√									
ISTE & CSTA	2011	A problem-solving process that includes (but is not limited to) six concepts for describing CT: formulating problems, organizing and analyzing data, abstractions, automation through algorithmic thinking, evaluation for efficiency and correctness, and generalizing.	√	√			√	√				√		√											
Brennan & Resnick	2012	Three key dimensions: computational concepts (the concepts designers employ as they program), computational practices (the practices designers develop as the program), and computational perspectives (the perspectives designers form about the world around them and themselves).					√	√					√	√			√				√				
Kazimoglu, Kiernan, Bacon, & MacKinnon	2012	Solving problems with logical thinking through using various computational models						√		√		√				√								√	

Authors	Year	Definition of Computational Thinking	Problem reformulation/ Identifying problem	Analyzing data	Recursion	Decomposition	Abstraction	System testing/Evaluation/ Debugging/ Assessing	Defining Problem	Solving problem	Analyzing the solution	Algorithms/ procedures	Generalization	Iteration	Automation	Parallelization	Simulation	Reusing & Remixing	Hypothesis testing	Pattern recognition	Logical reasoning	Modularity	Generate, select & Plan solution	Implement Solution	Socializing
Selby & Woollard	2013	CT as a cognitive or thought process that reflects the ability to think in abstractions, to think in terms of decomposition, to think algorithmically, to think in terms of evaluations, and to think in generalizations				√	√	√				√	√												
Csizmadia, Curzon, Dorling, Humphreys, Ng, Selby, & Woollard	2015	Process of recognizing aspects of computation in the world that surrounds us and applying tools and techniques from computing to understand and reason about natural, social and artificial systems and processes.				√	√	√				√									√				
Anderson	2016	Geared towards developing solutions for open-ended problems following a series of formalized steps				√	√	√				√								√					
Atmatzidou & Demetriadis	2016	Promotes new ways of thinking to students across all science disciplines. T				√	√					√	√									√			
Kalelioglu, Gulbahar, & Kukul	2016	Complex higher-order thinking, skills may require to use of the power of human cognitive ability and embrace the support of machines to think and solve problems.	√	√			√																√	√	
Shute, Sun & Clarke	2017	the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts				√	√	√				√	√	√											

Authors	Year	Definition of Computational Thinking	Problem reformulation/ Identifying problem	Analyzing data	Recursion	Decomposition	Abstraction	System testing/Evaluation/ Debugging/ Assessing	Defining Problem	Solving problem	Analyzing the solution	Algorithms/ procedures	Generalization	Iteration	Automation	Parallelization	Simulation	Reusing & Remixing	Hypothesis testing	Pattern recognition	Logical reasoning	Modularity	Generate, select & Plan solution	Implement Solution	Socializing
Dong, Cateté, Jocius, Lytle, Barnes, Albert, Joshi, Robinson, & Andrews	2019	A thinking process, not bounded by content area or tools, to help people solve problems in a systematic and generalizable way				√	√					√								√					
Palts & Pedaste	2020	As a tool for solving problems using algorithms							√	√	√														

Acknowledgements

The authors would like to acknowledge the Ministry of Higher Education Malaysia (MOHE) for the financial support of this research. This research is supported by MOHE through the FUNDAMENTAL RESEARCH GRANT SCHEME FOR RESEARCH ACCULTURATION OF EARLY CAREER RESEARCHERS (FRGS - RACER) with project code: RACER/1/2019/ICT01/UUM//4 (Project ID: 15296) (S/O Code: 14439)."

References

- Anderson, N.D., (2016). A call for computational thinking in undergraduate psychology. *Psychology Learning & Teaching*, 15, 226–234.
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., et al., (2016). A K-6 Computational Thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19(3), 47–57.
- Aranda, G. & Ferguson, J. P. (2018). Unplugged Programming: The future of teaching computational thinking?. *Pedagogika*, 68(3), 279-292.
- Atmatzidou, S. & Demetriadis, S. N. (2012). Evaluating The Role of Collaboration Scripts As Group Guiding Tools in Activities of Educational Robotics, In *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT 2012)*, Rome, Italy (pp. 298-302).
- 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.
- Barr, D., Harrison, J., & Conery, L. (2011). *Computational Thinking: A Digital Age*. Learning & Leading with Technology, March/April, 20-23.
- Basawapatna, A., Repenning, A., Koh, K.H., & Savignano, M. (2014). The Consume - Create Spectrum: Balancing Convenience and Computational Thinking in STEM Learning. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, Atlanta, USA (pp. 659-664).
- Berland, M. & Lee, V.R. (2011). Collaborative strategic board games as a site for distributed computational thinking, *International Journal of Game-Based Learning*, 1(2), 65–81.
- Brennan, K. & Resnick, M. (2012). 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 (pp.1-25).
- Chao, P.Y. (2016). Exploring students' computational practice, design, and performance of problem-solving through a visual programming environment. *Computers and Education*, 95, 202-215.
- Chang, C.K. (2011). Integrate social simulation content with game designing curriculum to foster computational thinking. In *Proceedings of the 7th International Conference on Digital Content, Multimedia Technology and its Applications*, Busan, South Korea (pp.115-118).
- Csizmadia, A., Curzon, P., Dorling M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking: A guide for teachers*. <http://community.Computingatschool.org.uk/resources/2324>
- CSTA and ISTE (2011). *Computational Thinking in K–12 Education leadership toolkit*. <http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>.
- Czerkawski, B.C., Lyman III, E.W., 2015. Exploring issues about computational thinking in higher education. *TechTrends* 59(2), 57–65.

- DeSchryver, M.D. & Yadav, A. (2015). Creative and Computational Thinking in the Context of New Literacies: Working with Teachers to Scaffold Complex Technology-Mediated Approaches to Teaching and Learning. *Journal of Technology and Teacher Education*, 23(3), 411-431.
- Dong, Y., CatetĂl, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*, Minneapolis (pp. 906-912).
- Google. (2018). *Google for education: Exploring computational thinking*. <https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>.
- Hemmendinger, D. (2010). A plea for modesty. *ACM Inroads*, 1(2), 4-7.
- Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Modern Computing*, 4(3), 583-596.
- Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play, *Procedia Computer Science*, 9, 522-531.
- Kitchenham, B., (2004). *Procedures for Undertaking Systematic Reviews*. Joint Technical Report, Computer Science Department, Keele University and National ICT Australia Ltd.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L. (2011). Computational Thinking for Youth in Practice. *ACM Inroads*, 2(1), 32-38.
- Lu, J. J. & Fletcher, G. H. L (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, 41(1), 260-264.
- McClelland, K. & Instructional, L. G. (2020). A Review of the Importance of Computational Thinking in K-12. In *Proceedings of The Tenth International Conference on Mobile, Hybrid, and On-line Learning, Rome, Italy* (pp. 32-34).
- Mohamed Zaki, F.Z., Wong, S. L. & Yaakub, M. R. (2019). A review of common features in computational thinking frameworks in K-12 education. *IOP Conference Series: Materials Science and Engineering*, 551, 1-5.
- National Research Council (NRC) (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. Washington, D.C.: The National Academies Press.
- Palts. T., & Pedaste, M. (2020) A Model for Developing Computational Thinking Skills. *Informatics in Education*, 19(1), 113-128.
- Papert, S. (1996). An Exploration in the Space of Mathematics Educations. *International Journal of Computers for Mathematical Learning*, 95-123.
- Romainor, N., Abdul Talib, C., & Abd Hakim, N. W., (2018). The Necessity of Computational Thinking in STEM Education: An Analysis with Recommended Research, *Learning Science and Mathematics*, 3, 1-12.
- Selby, C. (2015). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the Workshop in Primary and Secondary Computing Education, London, United Kingdom* (pp. 80-87).
- Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. In *Proceeding of Special Interest Group on Computer Science Education (SIGCSE) 2014, Atlanta* (pp. 1-6).
- Sengupta, P., Dickes, A., & Farris, A.V. (2018). Toward a Phenomenology of Computational Thinking in STEM. In: Khine, M.S. (Ed.): *Computational Thinking in STEM: Research Highlights*. Springer: New York, pp 49 – 72.

- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research Review*, 22, 142-158.
- 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., et al., (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.
- Weintrop, D., Holbert, N., Horn, M., & Wilensky, U. (2016). Computational thinking in constructionist video games. *International Journal of Game-based Learning*, 6(1), 1-17.
- Werneburg, S., Manske, S., & Hoppe, H. U. (2018). ctGameStudio—A game-based learning environment to foster computational thinking. In *Proceedings of the 26th International Conference on Computers in Education. Philippines* (pp. 676-685).
- Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. (2011). *Research Notebook: Computational Thinking - What and Why? The Link*. Pittsburgh, PA: Carneige Mellon.