

# Characterizing Computational Thinking for the Learning of Tertiary Educational Programs

Salman Firdaus Sidek<sup>1</sup>, Che Soh Said<sup>2</sup>, Maizatul Hayati Mohamad Yatim<sup>3\*</sup>

<sup>1</sup>Computing Department, Faculty of Art, Computing and Creative Industry, Sultan Idris Education University;  
salmanfirdaus@fskik.upsi.edu.my

<sup>2</sup>Computing Department, Faculty of Art, Computing and Creative Industry, Sultan Idris Education University;  
chesoh@fskik.upsi.edu.my

<sup>3</sup>Computing Department, Faculty of Art, Computing and Creative Industry, Sultan Idris Education University;  
maizatul@fskik.upsi.edu.my

\* corresponding author

DOI: <https://doi.org/10.37134/jictie.vol7.1.8.2020>

---

## Abstract

Computational thinking is a new concept that is hardly known by most ordinary Malaysians. On a positive note, in 2017, the Ministry of Education of Malaysia implemented a new curriculum by introducing two new secondary school subjects, namely *Fundamental of Computer Science* and *Computer Science*. These subjects contain five components of computational thinking called 'techniques, which are taught to Form 1 and Form 4 students, respectively. Likewise, the Department of Teachers Education of Malaysia also introduced six components of computational thinking in its training programs. In contrast, there is a lack of attention to such a concept given to tertiary education. A preliminary survey was conducted in early 2019, which involved 50 students majoring in 21 educational programs. The findings showed 92% of the respondents had no knowledge regarding the computational thinking skill, signifying an urgent need to determine relevant components that characterize such a skill needed in educational programs, which according to Wing (2006) has numerous components. In this study, the researchers aimed to identify relevant components that characterize the computational thinking skill required for students in learning educational programs. An extensive review of documents published in leading online databases, such as IEEEExplore, Science Direct, and Web of Science, was carried out that yielded 116 articles. Further analysis reduced this number to 29 articles that were related to the characterization of the computational thinking skill with 66 components. Among these components, algorithm, abstraction, and decomposition were the top three components with the highest frequency of being cited in the selected articles, registering percentage points of 9.78%, 7.41%, and 5.35%, respectively. As such, the design and development of new instructional approaches for the teaching of educational programs should emphasize three components to help students develop strong computational thinking skills.

**Keywords:** computational thinking, educational programs, online repositories, tertiary education.

---

## INTRODUCTION

In 1980, an American mathematician of South African descent named Seymour Papert attempted to embed computational characteristics into human life. One of his approaches was based on a computational idea and technology as learning and thinking tools that emotionally and cognitively evolve (Papert, 1980). The term 'computational thinking' was coined by Papert, which first appeared in his influential work called *Mindstorms*. Thus, most scholars, such as Kong and Wong (2017), Rico Lugo, Olabe, and Niño (2018) and Shute, Sun, and Asbell-Clarke (2017) attributed such a computational concept to this renowned mathematician.

Over recent years, many researchers have begun to focus their studies on such a new concept, which have been spurred after the publication of Wing's (2006) seminal article that emphasizes computational thinking as an important skill on par with as reading, writing, and arithmetic skills that every student should acquire (Li, 2016; Wing, 2006). Such an emphasis is hardly surprising, as computer technology has become an integral part of people's lives in the 21<sup>st</sup> century to support them in problem-solving (Wing, 2006). Equally significant, Wing's argument has made a profound impact on education across the world. For example, in many developed nations, Australia, England, Estonia, Finland, New Zealand, Norway, Sweden, South Korea, Poland, and United States, many initiatives have been carried out to embed computational thinking skill in their educational curricula since 2011 (Heintz, Mannila, & Färnqvist, 2016). Essentially, such initiatives aimed at cultivating computational thinking among students and teachers at various educational levels, ranging from K-12 education curriculum models to in-service teacher programs.

In Malaysia, the MOE of Malaysia has also emphasized computational thinking as a fundamental thinking skill based on technology that is needed by all to support problem-solving in the future (Curriculum Development Division [BPK], 2015). Hence, an innovative initiative called 'Computational Thinking for Everyone' (National Research Council [NRC], 2010, p. viii) was launched by the MOE of Malaysia. In early 2017, the ministry implemented several new school curricula, namely *Secondary School Standard Curriculum* (KSSM), *Integrated Curriculum* (KBD), and *Tahfiz Integrated Curriculum* (KBT), to help students acquire such a skill. These new school curricula introduced new school subjects, namely *Fundamental of Computer Science* (ASK) and *Computer Science* (SK), to Form 1 students (13 years-old students) and Form 4 students (16 years-old students), respectively.

In these subjects, they will learn five techniques, considered as the fundamental of the computational thinking skill, namely decomposition, pattern recognition, abstraction, generalization, and algorithm. From a practical standpoint, Scratch and HTML programming software will be used as learning tools to practice the knowledge and skill of computational thinking through various activities, such as the development of algorithms and computer programs based on the ASK textbook and ASK Teaching and Learning module. Admittedly, the number of students acquiring such an important skill will be very small as not all secondary students have the chance to pursue the two school subjects (ASK and

SK), as the enrolment for these two subjects is subjected to the approval of respective school administrators. Unfortunately, such a practice continues to exist despite these two subjects being made compulsory subjects, as stated in the Ministry of Education's 2016 Circular (MOE, 2016).

In the training context, the Teacher Education Division (BPG) has organized the Form 1 Computer Science Teacher Training Program for selected schoolteachers. Furthermore, in 2016, the MOE in collaboration with Malaysia Digital Economy Corporation Sdn. Bhd. (MDEC) launched the *My Digital Maker Program* to train in-service teachers who had been assigned to teach ASK and SK subjects, which had been recently introduced in schools at that time. In this program, they learned the six components or main concepts of computational thinking skills, namely decomposition, pattern, abstraction, algorithm, logical reasoning, and evaluation.

## **PROBLEM STATEMENT**

In Malaysia, the principle of 'computational thinking for everyone' has thus far not been fully adopted in the higher learning context (NRC, 2010, p. viii). To date, efforts that focus on the development of the computational thinking skill have been seriously lacking, which pales in comparison with those relating to the development of higher-order thinking skills (HOTS). A case in point is best exemplified by the finding of a survey conducted in Sultan Idris Education University in 2019 in which 50 respondents from 21 educational programs run by nine faculties were asked whether they knew the concept of computational thinking skill. The findings were surprising and quite controversial as an overwhelming majority (92%) of the respondents had no knowledge regarding such a concept. Surely, such findings may have serious repercussions because they, as future teachers, must not only know such an important concept but also cultivate it among their students. Thus, more efforts are entailed to raise the level of awareness of the importance of the computational thinking skill among future teachers. On a positive note, however, 94% of respondents agreed that they had to develop such a skill to help them deal with the teaching challenges in the twenty-first century.

To date, the Teachers Education Division (BPG) of MOE, in collaboration with MDEC, have organized several training programs in computational thinking. However, such programs can only train a few in-service teachers at a time, which makes the aim of 'Computational Thinking for Everyone' initiative untenable. Certainly, the relatively low awareness of such a concept among teachers and a lack of training in computational thinking may necessitate policymakers to seek a new approach to deal with such predicaments. In terms of practicality, the concept of computational thinking skill can be embedded into the educational programs pursued by undergraduates. Against such a backdrop, the researchers carried out a systematic literature review to identify the appropriate components that characterize the computational thinking skill needed for students pursuing educational programs at the tertiary level of education.

## **RELATED WORKS**

According to a seminal article written by Wing (2006), computational thinking involves problem-solving, designing systems, and understanding human behaviors based on the basic concepts of computer science. In today's technology-driven realm, such a concept is essential to help people solve a myriad of problems. In terms of computer hardware, computational thinking is also important for computer scientists to think mathematically to enable computer systems to interact with the real world. From the perspective of computational thinking too, computer science learning involves learning of programming concept which requires programmer to think at various levels of abstraction, which is complementary to mathematical and engineering thinking.

The features of computational thinking benchmarked by Wing (2006) was refined when the scope and nature of such an ability had been reviewed from a broader perspective, including its concepts, applications, tools, skill sets, languages, abstraction automation, and cognitive tools (NRC, 2010). Likewise, computational thinking has also been refined in terms of pedagogy and its context in everyday life, games and gaming, science, engineering, and journalism (NRC, 2011). Also, the findings of studies carried out between 2008 and 2010 by the National Academy of Sciences, the Computer Science Teachers Association (CSTA), and the International Society for Technology in Education (ISTE) (which were funded by three research grants from the National Science Foundation) had led to a new operational definition of computational thinking for K-12 Education (Toedte & Aydeniz, 2015).

Based on this operational definition, computational thinking is defined as a problem-solving process characterized by, but not limited, to 1) making a problem as a formula that allows computers and other tools to be used to help humans solve problems, 2) organizing and analyzing data logically, 3) representing data through abstraction using models and simulations, 4) automating solutions through algorithmic thinking, 5) identifying, analyzing, and implementing possible solutions based on the most efficient and effective combination of steps and resources, and 6) generalizing and transferring the problem-solving process to other various types of problems. These six characteristics should be supported by 1) confidence in dealing with something complex, 2) perseverance in facing difficult problems, 3) tolerance to something vague, 4) ability in dealing with problems with open solutions, and 5) ability in communicating and collaborating to achieve goals or solutions (CSTA & ISTE, 2011). In addition, Vallance and Towndrow (2016) suggest that the operational definition could be refined by including feedback as another characteristic.

Moreover, Google for Education has adopted the operational definition of computational thinking skill proposed by CSTA and ISTE (2011), which is reflected on its website highlighting 11 concepts of computational thinking, including mental processes and tangible outcomes, as follows: 1) abstraction, 2) algorithmic design, 3) automation, 4) data analysis, 5) data collection, 6) data representation, 7) decomposition, 8) parallelism, 9) pattern generalization, 10) pattern recognition, and 11) simulation. Furthermore, many science and computer teachers argue that the concept of computational thinking

stated in the operational definition is relevant to the fifth practice of the eight science and engineering education practices outlined by the NRC (Park & Jeon, 2015).

Barr and Stephenson (2011) proposed 12 concepts or capabilities of computational thinking for K-12 with a focus on algorithmic problem solving as follows: 1) data collection, 2) data analysis, 3) data representation, 4) abstraction, 5) model analysis and validation, 6) automation, 7) testing and verification, 8) algorithms and procedures, 9) problem decomposition, 10) control structure, 11) parallelism, and 12) simulation. Later, Grover and Pea (2013) (who viewed abstraction as the backbone of computational thinking) proposed nine elements of computational thinking for K-12 education as follows: 1) abstraction and pattern generalization (including models and simulations), 2) systematic information processing, 3) symbol and representation systems, 4) algorithmic notions of flow control, 5) structured or modular problem decomposition, 6) iterative, recursive, and parallel thinking, 7) conditional logic, 8) performances and efficiency constraints, and 9) systematic debugging and error detection.

Jenson and Droumeva (2016) designed an entry-level instruction based on seven of these elements, which were found to be challenging, especially when designing instructions with specific objectives. Such a revelation signifies that the process of designing efficacious teaching and learning activities involving computational thinking skill will be extremely difficult.

Earlier, Brennan and Resnick (2012) developed a framework that elaborates computational thinking based on three dimensions, namely concept, practice, and perspective. Seven 'computational concepts' used by designers when designing a computer program or when performing non-programming activities are as follows: 1) sequence, 2) loops, 3) events, 4) parallelism, 5) conditions, 6) operators, and 7) data. Four 'computational practices' executed during programming are incrementalization and iteration, testing and debugging, reusing and remixing, and abstraction and modularization. Three 'computational perspectives' that relate to oneself and his/her surrounding are expression, connection, and questioning (Zhong, Wang, Chen, and Li, 2016). Such perspectives helped guide other endeavors, such as the design of six types of tasks for assessing computational thinking skills of pre-service teachers (Bean, Weese, Feldhausen, & Bell, 2015), English language learning (Weng & Wong, 2017), the verification of the criteria of Computational Thinking Test (Román-González, Pérez-González, & Jiménez-Fernández, 2017), and the production of board games for children (Tsarava, Moeller, & Ninaus, 2018).

In contrast to Brennan and Resnick's (2012) conceptualization, Altanis, Retalis, and Petropoulou (2018) proposed several computational concepts as follows: 1) flow control, 2) abstraction, 3) user interactivity, 4) synchronization, 5) parallelism, and 6) logic. Likewise, Seiter and Foreman (2013), and Repenning, Basawapatna, and Escherle (2016) proposed a number of concepts as follows: 1) abstraction, 2) procedures and algorithms, 3) data representation, 4) decomposition, 5) parallelism, 6) synchronization, 7) abstraction, 8) analysis, and 9) automation. On the other hand, the definition of computational thinking by CS Principles combines the concepts of '7 Big Ideas' and '6 Computational

Practices' (Arraki et al., 2014). The former comprises seven concepts as follows: 1) creativity, 2) abstraction, 3) data and information, 4) algorithms, 5) programming, 6) internet, and 7) global impact while the latter consists of six concepts as follows: 1) connecting computing, 2) producing computational artifacts, 3) abstracting, 4) analyzing problems and artifacts, 5) communicating, and 6) collaborating.

As highlighted, the definitions of the concept of computational thinking vary considerably depending on the tools or methods used in the learning activities. For example, the computational thinking learning modules for the K-12 science curriculum developed by teachers, computer sciences students, computer engineering students, and applied computing students use the definition prescribed by the CSPrinciples (Arraki et al., 2014). Furthermore, the educational robot used in K-12 computational thinking learning intervention is based on a computational model consisting of five skills, namely abstraction, generalization, algorithms, modularization, and decomposition (Atmatzidou & Demetriadis, 2016). Clearly, tangible programming tools used to cultivate the concept of computational thinking among children have a slight distinction because the learning activities focus on the following concepts: 1) abstraction, 2) automation, 3) problem analysis and decomposition, and 4) creativity (Wang, Wang, & Liu, 2014).

Meanwhile, Ioannidou, Bennett, Repenning, Koh, and Basawapatna (2011) developed the Computational Thinking Pattern (CTP) for game design activities and science simulations involving several components as follows: 1) collision, 2) push, 3) pull, 4) transport, 5) generation, 6) absorption, 7) choreography, 8) diffusion, 9) path-finding/seeking, 10) collaborative diffusion, and 11) multiple needs. Interestingly, the gamification-based framework has 11 features of learning activities to develop computational thinking skills as follows: 1) formulating problems to be solved using software and hardware, 2) modeling the problems, 3) organizing and analyzing data logically, 4) representing data through abstraction, model, and simulation, 5) automating solutions through algorithmic thinking, 6) identifying, analyzing, and implementing efficient, effective solutions, 7) synthesizing individual solutions and finding the most optimal solution, 8) generalizing and transferring the problem-solving process to various types of problems, 9) building abstraction and pattern recognition capabilities, 10) testing and debugging processes and conducting self-assessment (Kotini & Tzelepi, 2015).

In terms of educational programs for pre-service teachers, Yadav, Zhou, Mayfield, Hambrusch, and Korb (2011), and Yadav, Mayfield, Zhou, Hambrusch, and Korb (2014) proposed five basic concepts of computational thinking as follows: problem introductory and decomposition, 2) abstraction, 3) logical thinking, 4) algorithms, and 5) debugging. Likewise, non-technical and non-scientific fields, such as humanity, music, and sociology at the collegiate level, uses the definitions proposed by Wing (2006) and comprehensive operational definition used by CSTA and ISTE (Toedte & Aydeniz, 2015). On a broader learning perspective, computational thinking skill can also be assessed through several university courses involving the following concepts: 1) abstraction, 2) algorithmic thinking, 3) decomposition, 4) evaluation, and 5) generalization (Araujo, Santos, Andrade, Guerrero, & Dagiene, 2017).

In Malaysia's secondary school curricula, two school subjects, namely *Fundamental of Computer Science* (Kassan, Fatt, & Meng, 2016) and *Computer Science* (Chua, Yew, Mohamad, & Ismail, 2016) have been introduced to help Form 1 and Form 4 pupils, respectively, to learn five components of computational thinking called 'techniques' as follows: 1) decomposition, 2) pattern recognition, 3) abstraction, 4) generalization, and 5) algorithms. By contrast, in-service teachers use the 'Computational Thinking and Computer Science Teaching Certificate' module to learn six components of computational thinking as follows: 1) decomposition, 2) pattern, 3) abstraction, 4) algorithm, 5) logical reasoning, and 6) evaluation. Additionally, they can learn other related components as follows: 1) parallelism, 2) simulation, 3) data representation, 4) data analysis, 5) data collection, and 6) automation.

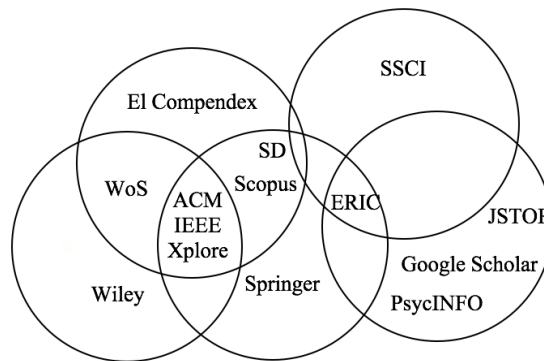
## **METHODOLOGY AND DATA COLLECTION**

Over recent years, seeking research resources has become more efficient by the emergence of several online repositories. For example, the Web of Science (WoS) and Wiley Online Library (Wiley) are two common repositories for academic writing. On the other hand, the Association for Computing Machinery's (ACM) and the Institute for Electrical and Electronics Engineers' (IEEEExplore) digital libraries are repositories for computing and technology (Toedte & Aydeniz, 2015). A case in point, Lye and Koh's (2014) review on studies of the development of computational thinking through programming and its implication on research and teaching on ERIC database and the Social Sciences Citation Index (SSCI) yielded 27 articles related to interventions. Later, Toedte and Aydeniz (2015) conducted a similar review of studies that focused on computational thinking and its impact on K-12 science education that yielded 17 articles from WoS, four (4) articles from IEEEExplore, and 20 articles from ACM. Currently, exploring the IEEEExplore database through Wiley has been significantly simplified.

A year later, de Araujo, Andrade, and Guerrero (2016) conducted a similar review on works relating to computational thinking ability assessment on ACM, ERIC, IEEEExplore, Science Direct (SD), Springer, and Scopus databases, which helped yield 27 related articles. Specifically, they chose such repositories because of their relevancy to computer science and education (de Araujo et al., 2016). Then, Shute et al. (2017) carried out a review on works published in ERIC, PsycINFO, JSTOR, and Google Scholar databases, in which they found 45 out of 70 articles reviewed were related to the definitions, interventions, assessments, and models of computational thinking. Subsequently, Silva, da Silva, Toda, and Isotani (2018) performed an extensive review of similar works involving 712 articles on ACM, El Compendex, IEEEExplore, WoS, SD, and Scopus databases. They found only 15 articles were relevant to the impact of teaching approaches on computational thinking of upper secondary students. Both teams of researchers, however, did not provide any justifications for the selection of such repositories.

As highlighted in Figure 1, ERIC, ACM, and IEEEExplore databases are the top three online repositories referred to by researchers, one of which was chosen by the present researchers to seek

relevant articles in this study. Additionally, the researchers referred to WoS and SD databases, which are also widely used by many researchers for academic writing (Toedte & Aydeniz, 2015), to seek relevant articles. Specifically, the researchers relied on such online repositories because of their relevancy to the fields of Computer Science and Education (de Araujo et al., 2016).



**Figure 1:** A Venn's Diagram showing 13 repositories used by the researchers

In the Malaysian educational landscape, computational thinking is a new concept that not many Malaysian researchers and scholars are familiar with, which explains why most of them have limited knowledge of such a concept. On the bright side, however, relevant governmental agencies and organizations have begun to emphasize the importance of the development of computational thinking among students. For example, in 2016, the MOE of Malaysia released a new textbook and a learning module for the *Fundamental of Computer Science* subject that contain computing thinking elements. Likewise, in the same year, the Division of Teachers Education (BPG) under the same ministry published a learning module containing the same elements of computational thinking for the Teacher Training Program. Similarly, a collaboration between the MOE of Malaysia and MDEC in 2016 helped develop the My Digital Maker Program, which aimed at preparing in-service teachers with strong knowledge in computational thinking that would help them teach the ASK and SK subjects with greater efficacy.

The document search with the keyword 'computational thinking' in abstracts, document titles, and publication titles on IEEE Xplore and Wiley databases published from 2014 to March 2018 yielded 39 journal articles and magazine-type documents. The analysis of these documents showed only 12 were related to computational thinking in education, curriculum, pedagogy, learning, and assessment or related to the definition and concept of computational thinking. In contrast, document search with the same keyword with the conjunction 'and' yielded only 12 journal articles, out of which 11 met the required criteria. Furthermore, the contents of these articles overlapped with those of 11 documents obtained from the previous analysis.



The latter document search using the keyword ‘computational thinking’ together with the conjunction ‘and’ was more accurate than the former document search. Therefore, the next document search on IEEEExplore and Wiley databases was carried out with the same keyword and conjunction that revealed 167 conference papers relating to computational thinking. Out of these, 52 had to be discarded because their contents were not written in English and overlapped with those obtained from previous document search, and some were only abstracts that had little information on computational thinking. From these, only 58 papers met the selection criteria, with the remaining 57 papers being used for additional reading. Overall, out of 70 papers that had met the selection criteria, only 47 papers were selected in this study.

Document search for titles and abstracts on SD using the same keyword and conjunction found 58 relevant documents, including review articles, research articles, and editorials, published from 2014 to March 2018. These documents were then analyzed that showed only 17 documents had met the selection criteria relating to computational thinking in education, curriculum, pedagogy, learning, and assessment or to the definition and concept of computational thinking. Therefore, these documents were used in this study. Articles containing the term ‘computational science’ or other aspects not related to computational thinking had to be discarded from the selection. Document search with the keyword ‘computational thinking’ for topics or titles on the WoS database yielded 61 relevant articles published from 2014 to March 2018 on several online resources, including ACM, Taylor & Francis, and Springer databases. Out of these, only 22 fulfilled the selection criteria and, therefore, were used in this study. Additionally, 30 relevant articles selected from Internet webpages, books, and reports were included in this study. Table 1 shows the number of articles selected from various online resources.

**Table 1:** The number of articles selected from online resources.

Resource	Keyword	Searching Method	Year	Document Type	No. of Articles	No. of Articles Met the Criteria	No. of Ref.
IEEEExplore and Wiley	computational thinking	abstract, document title, and publication title	2014 - March 2018	Journal and magazine	39	12	
IEEEExplore and Wiley	"computational thinking"	abstract, document title, and publication title	2014 - March 2018	Journal and magazine	12	11	47
IEEEExplore and Wiley	"computational thinking"	abstract, document title, and publication title	2014 - March 2018	Conference paper	167	58	
Science Direct	"computational thinking"	title or abstract, or keyword	2014 - March 2018	Literature article, research article	58	17	17
WoS	"computational thinking"	topic or title	2014 - March 2018	Article	61	22	22
Others	-	-	-	Internet webpage, books, workshop's report	-	>30	>30
Total						>150	>116

## FINDINGS AND DISCUSSION

Detailed analysis of 116 selected articles and documents revealed only 29 articles were related to the characterization of the computational thinking skill, the components of which are shown in Table 2. As highlighted, 66 components characterizing the computational thinking skill were referenced more than one. In particular, abstraction and algorithm had the highest frequency of being cited in the articles, registering percentage points of 9.87% and 7.41%, respectively. Even though these percentages were based on documents derived from the combination of K-12 education and tertiary education, these findings are consistent with Sondakh's (2018) assertion that algorithm and abstraction are two components of computational thinking skill that are often assessed at the tertiary education level. Table 2 summarizes the number of individual components that characterize the computational thinking skill, as reported in the current literature.

**Table 2:** The number of individual components that characterize the computational thinking skill

No Elements	References	
1 Abstraction	Wing (2006) ISTE and CSTA (2011) Google for Education (n.d.) Barr and Stephenson (2011) Grover and Pea (2013) Li (2016) Arraki et al. (2014) Wang et al. (2014) Atmatzidou and Demetriadis (2016) Ioannidou et al. (2011) Worrell, Brand, and Repenning (2015) Yadav et al. (2011) Kotini and Tzelepi (2015) Brennan and Resnick (2012) Tsarava et al. (2018) Altanis et al. (2018) Seiter and Foreman (2013) Kalelioglu Gülbahar, and Kukul (2016) Angeli et al. (2016) Repenning et al. (2016) Basogain et al. (2016) Weintrop et al. (2015) Moreno-León, Robles, and Román-González Vallance and Towndrow (2016) Korkmaz, Çakir, and Özden (2017) Cross, Hammer, Zito, and Nourbakhsh (2016) Araujo et al. (2017) Lowe and Brophy (2017) Kassan et al. (2016)	24
2 Algorithm		18
3 Automation		9
4 Data Analysis		8
5 Data Collection		6
6 Data Representation		10
7 Decomposition		13
8 Parallelism		9
9 Generalization		11
10 Recognition		5
11 Simulation		9
12 Formulation		3
13 Combination of the most efficient and effective steps and sources to produce possible solution		2
14 Model analysis and validation		1
	<b>Total</b>	<b>75</b>

---

	Wing (2006)	ISTE and CSTA (2011)	Google for Education (n.d.)	Barr and Stephenson (2011)	Grover and Pea (2013)	Li (2016)	Arraki et al. (2014)	Wang et al. (2014)	Atmatzidou and Demetriadis (2016)	Ioannidou et al. (2011)	Worrell, Brand, and Repenning (2015)	Yadav et al. (2011)	Kotini and Tzelepi (2015)	Brennan and Resnick (2012)	Tsarava et al. (2018)	Altanis et al. (2018)	Seiter and Foreman (2013)	Kalefioglu Gülbahar, and Kukul (2016)	Angeli et al. (2016)	Repenning et al. (2016)	Basogain et al. (2016)	Weintrop et al. (2015)	Moreno-León, Robles, and Román-González	Vallance and Towndrow (2016)	Korkmaz, Çakir, and Özden (2017)	Cross, Hammer, Zito, and Nourbakhsh (2016)	Araujo et al. (2017)	Lowe and Brophy (2017)	Kassan et al. (2016)	<b>Total</b>	
<i>...cont.</i>																															
15 Testing and verification		/																													<b>1</b>
16 Control structure		/	/										/	/	/				/				/								<b>7</b>
17 Systematic information processing			/																											<b>1</b>	
18 Iterative		/										/														/				<b>3</b>	
19 Recursive		/																												<b>1</b>	
20 Performance constraint and effectiveness		/																												<b>1</b>	
21 Debugging		/								/	/	/				/										/				<b>6</b>	
22 Connecting computing							/																							<b>1</b>	
23 Producing computational artifacts							/																							<b>1</b>	
24 Problem analysis and artifacts							/			/										/										<b>3</b>	
25 Communication	/						/																		/					<b>3</b>	
26 Interactive	/														/				/			/								<b>4</b>	
27 Cooperative																									/					<b>1</b>	
28 Collaboration							/		/	/																				<b>3</b>	
29 Creativity							/																		/					<b>2</b>	

	Wing (2006)	ISTE and CSTA (2011)	Google for Education (n.d.)	Barr and Stephenson (2011)	Grover and Pea (2013)	Li (2016)	Arraki et al. (2014)	Wang et al. (2014)	Atmatzidou and Demetriadis (2016)	Ioannidou et al. (2011)	Worrell, Brand, and Repenning (2015)	Yadav et al. (2011)	Kotini and Tzelepi (2015)	Brennan and Resnick (2012)	Tsarava et al. (2018)	Altanis et al. (2018)	Seiter and Foreman (2013)	Kalefioglu Gülbahar, and Kukul (2016)	Angeli et al. (2016)	Repenning et al. (2016)	Basogain et al. (2016)	Weintrop et al. (2015)	Moreno-León, Robles, and Román-González	Vallance and Towndrow (2016)	Korkmaz, Çakir, and Özden (2017)	Cross, Hammer, Zito, and Nourbakhsh (2016)	Araujo et al. (2017)	Lowe and Brophy (2017)	Kassan et al. (2016)	<b>Total</b>	
...	<i>cont.</i>																														
30 Modularization		/		/				/				/														/					<b>4</b>
31 Collision								/	/																						<b>2</b>
32 Push								/	/																						<b>2</b>
33 Pull								/	/																						<b>2</b>
34 Transport								/	/																						<b>2</b>
35 Generation								/	/																						<b>2</b>
36 Absorption								/	/																						<b>2</b>
37 Choreography								/	/																						<b>2</b>
38 Diffusion								/	/																						<b>2</b>
39 Path finding/seeking								/	/																						<b>2</b>
40 Multiple needs								/	/																						<b>2</b>
41 Logic thinking										/			/										/		/						<b>4</b>
42 Critical thinking																									/						<b>1</b>
43 Modelling	/		/									/		/				/				/			/	/					<b>7</b>
44 Synthesizing												/																			<b>1</b>
45 Assessment / evaluation												/	/													/	/				<b>4</b>
46 Sequence												/	/					/													<b>3</b>
47 Loop												/	/																		<b>2</b>
48 Event												/	/																		<b>2</b>

	Wing (2006)	ISTE and CSTA (2011)	Google for Education (n.d.)	Barr and Stephenson (2011)	Grover and Pea (2013)	Li (2016)	Arraki et al. (2014)	Wang et al. (2014)	Atmatzidou and Demetriadis (2016)	Ioannidou et al. (2011)	Worrell, Brand, and Repenning (2015)	Yadav et al. (2011)	Kotini and Tzelepi (2015)	Brennan and Resnick (2012)	Tsarava et al. (2018)	Altanis et al. (2018)	Seiter and Foreman (2013)	Kalefioglu Gülbahar, and Kukul (2016)	Angeli et al. (2016)	Repenning et al. (2016)	Basogain et al. (2016)	Weintrop et al. (2015)	Moreno-León, Robles, and Román-González	Vallance and Towndrow (2016)	Korkmaz, Çakir, and Özden (2017)	Cross, Hammer, Zito, and Nourbakhsh (2016)	Araujo et al. (2017)	Lowe and Brophy (2017)	Kassan et al. (2016)	<b>Total</b>		
...cont.																																
49 Operator													/	/																		2
50 Data													/																			1
51 Reusing and remixing													/																			1
52 Expressing													/							/												2
53 Connecting													/																			1
54 Questioning													/																			1
55 Constant and variable														/																		1
56 Pattern														/																		1
57 Synchronization															/	/							/									3
58 Procedure			/														/															2
59 Mathematical reasoning	/																/															2
60 Conception	/																/				/											3
61 Multimedia content integration																					/											1
62 Development of object block and function block																					/											1
63 Computational problem solution	/																				/		/	/								4

	Wing (2006)		
	ISTE and CSTA (2011)		
	Google for Education (n.d.)		
	Barr and Stephenson (2011)		
	Grover and Pea (2013)		
	Li (2016)		
	Arraki et al. (2014)		
	Wang et al. (2014)		
	Atmatzidou and Demetriadis (2016)		
	Ioannidou et al. (2011)		
	Worrell, Brand, and Repenning (2015)		
	Yadav et al. (2011)		
	Kotini and Tzelepi (2015)		
	Brennan and Resnick (2012)		
	Tsarava et al. (2018)		
	Altanis et al. (2018)		
	Seiter and Foreman (2013)		
	Kalefioglu Gülbahar, and Kukul (2016)		
	Angeli et al. (2016)		
	Repenning et al. (2016)		
	Basogain et al. (2016)		
	Weintrop et al. (2015)		
	Moreno-León, Robles, and Román-González		
	Vallance and Towndrow (2016)		
	Korkmaz, Çakir, and Özden (2017)		
	Cross, Hammer, Zito, and Nourbakhsh (2016)		
	Araujo et al. (2017)		
	Lowe and Brophy (2017)		
	Kassan et al. (2016)		
	<b>Total</b>		
...cont.			
64 System thinking		/	1
65 System designing	/		1
66 Understanding human behavior	/		1

Meanwhile, decomposition, which stood at a percentage point of 5.35%, was the component with the third highest frequency. Such a finding supports the contention that abstraction and decomposition have become a focus of studies among researchers (Turchi & Malizia, 2016). Furthermore, both are regarded as critical components of the computational thinking skill that help convert real-world problems into computer models (Ota, Morimoto, & Kato, 2016). The above findings are consistent with earlier findings that indicate abstraction, algorithm, decomposition, and generalization are among 20 components of computational thinking that enable an individual to understand and apply problem-solving in multiple contexts (Burke, Bailey, & Ruiz, 2019). In higher learning, Yadav et al. (2011, 2014) argue that education courses for students of educational programs and pre-service teachers should contain the components abstraction, algorithm, and decomposition, in addition to logical thinking and debugging. Additionally, the same findings give credence to Araujo et al.'s (2017) argument that abstraction, algorithmic thinking, and decomposition are the three critical components (out of five components) of computational thinking skills that could, and should, be assessed through certain university courses, in addition to evaluation and generalization.

Admittedly, the concepts of computational thinking vary quite considerably given diverse academic backgrounds and expertise of scholars and researchers and the types of learning activities. Nevertheless, the fundamental components of such concepts remain similar, such as abstraction, generalization, algorithm, decomposition, and automation. Such components have featured prominently in the learning activities of the educational robot for K-12 (Atmatzidou & Demetriadis, 2016), tangible programming tools for children (Wang et al., 2014), and learning activities created

based on the gamification framework (Kotini & Tzelepi, 2015). Among these components, abstraction, algorithm, and decomposition have been extensively implemented in higher learning (Yadav et al., 2011, 2014; Araujo et al., 2017), such as in-service teacher's programs in Malaysia.

After having studied the concepts for more than 10 years, many scholars and researchers have reached a consensus that indicates abstraction and decomposition are the only two components that accurately characterize the computational thinking skill (Turchi & Malizia, 2016). Despite such a consensus, some scholars have proposed several components of computational thinking, the number of which keeps on changing. For example, Lowe and Brophy (2017) proposed 25 different computational thinking concepts after reviewing studies reported in various resources. Such concepts can be divided into nine categories to form an operational model of computational thinking as follows: 1) abstraction, 2) decomposition, 3) pattern recognition and generalization, 4) algorithms, 5) data collection, analysis, and representation, 6) parallelism, 7) iteration, 8) simulation and automation, and 9) testing and debugging.

Additionally, after having analyzed several research findings, de Araujo et al. (2016) found four characteristics of computational thinking that have been widely emphasized by most researchers, namely problem solving or trouble-shooting, algorithms or algorithmic thinking, abstraction, and decomposition. Likewise, Shute et al. (2017) reviewed 45 articles relating to computational thinking that helped identify four similar characteristics but with different terms used, namely debugging, algorithms, abstraction, and decomposition. By contrast, after reviewing the findings of seven previous studies, Rose, Habgood, and Jay (2017) found seven concepts that have been widely used to define computational thinking as follows: 1) abstraction and generalization, 2) algorithms and procedure, 3) data collection, analysis, and representation, 4) decomposition, 5) parallelism, 6) debugging, testing, and analysis, and 7) control structure.

Surprisingly, only one out of 41 articles delved into the implementation of comprehensive computational thinking features as defined in the operational definition of the Computational Thinking for K-12 Education (Toedte & Aydeniz, 2015). Thus, on average, researchers have referred to only three out of 15 features cited by Wing (2006) and to two out of six features of the operational definition (Toedte & Aydeniz, 2015). Overall, the findings of this study showed that abstraction, algorithm, and decomposition were the most cited components that characterize computational thinking (as highlighted in Table 2), which are consistent with the findings of previous studies (Sondakh, 2018, Araujo et al., 2017; Yadav et al., 2011, 2014). Therefore, such components should be given a strong emphasis not only on secondary education but also on tertiary education.

## **CONCLUSION AND RECOMMENDATION**

In general, most university students majoring in education in Malaysia do not have strong computational thinking skills. As such, relevant stakeholders must put in more effort to help them acquire such a skill that is highly needed in learning in the twenty-first century. One of the efforts may



involve redesigning existing instructional approaches and employing novel learning aids. As acknowledged, many educational practitioners in Malaysia have little awareness of such a concept let alone its importance in problem-solving. Thus, it becomes highly imperative for researchers to identify critical components that characterize critical thinking that can help design effective instructional approaches consisting of appropriate learning activities and novel learning aids. As revealed in this study, abstraction, algorithm, and decomposition were the three fundamental components out of 66 cited by scholars and researchers that could accurately characterize the computational thinking skill. Such findings can surely pave a new direction of research to guide future studies by focusing on these three critical components.

## REFERENCES

- Altanis, I., Retalis, S., & Petropoulou, O. (2018). Systematic design and rapid development of motion-based touchless games for enhancing students' thinking skills. *Education Sciences*, 8(1), 18. <https://doi.org/10.3390/educsci8010018>
- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, 19(3), 47-57. Retrieved from <https://www.jstor.org/stable/jeductechsoci.19.3.47>
- Araujo, A. L. S. O., Santos, J. S., Andrade, W. L., Guerrero, D. D. S., & Dagiene, V. (2017). Exploring computational thinking assessment in introductory programming courses. In C. Dyer (Ed.), *2017 IEEE Frontiers in Education Conference* (pp. 1–9). <https://doi.org/10.1109/FIE.2017.8190652>
- Arraki, K., Blair, K., Bürgert, T., Greenling, J., Haebe, J., Lee, G., ... Hug, S. (2014). DISSECT: An experiment in infusing computational thinking in K-12 science curricula. In *2014 IEEE Frontiers in Education Conference* (pp. 1–9). <https://doi.org/10.1109/FIE.2014.7044262>
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670. <https://doi.org/10.1016/j.robot.2015.10.008>
- 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. <https://doi.org/10.1145/1929887.1929905>
- Basogain, X., Olabe, M. A., Olabe, J. C., Ramírez, R., Del Rosario, M., & Garcia, J. (2016). PC-01: Introduction to computational thinking: Educational technology in primary and secondary education. In F. J. García-Peñalvo, & A. J. Mendes (Eds.), *2016 International Symposium On Computers in Education* (pp. 1–5). <https://doi.org/10.1109/SIE.2016.7751816>
- Basu, C., Mustafara, E., & Rich, K. (2016). Computational thinking. Retrieved February 10, 2018, from <https://circlcenter.org/computational-thinking/>
- Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015). Starting from scratch: Developing a pre-service teacher training program in computational thinking. In *2015 IEEE Frontiers in Education Conference* (pp. 1–8). <https://doi.org/10.1109/FIE.2015.7344237>
- 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 Education Research Association* (pp. 1–25). Retrieved from <https://www.media.mit.edu/publications/new-frameworks-for-studying-and-assessing-the-development-of-computational-thinking/>
- Burke, C., Bailey, C. S., & Ruiz, P. (2019, January 17). Assessing computational thinking. Retrieved May 23, 2019, from <https://circlcenter.org/assessing-computational-thinking/>
- Chua, S. G., Yew, K. H., Mohamad, Z., & Ismail, F. (2016). *Sains Komputer Tingkatan 4*. Selangor, Malaysia: Oxford Fajar Sdn. Bhd.
- Cross, J., Hamner, E., Zito, L., & Nourbakhsh, I. (2016). Engineering and computational thinking talent in middle school students: a framework for defining and recognizing student affinities. In *2016 IEEE Frontiers in Education Conference* (pp. 1–9). <https://doi.org/10.1109/FIE.2016.7757720>
- Curriculum Development Division. (2015). *Kurikulum Standard Sekolah Menengah Asas Sains Komputer Tingkatan 1*. Retrieved from <https://gurubesar.my/wp-content/uploads/2018/11/10-DSKP-KSSM-Tingkatan-1-Asas-Sains-Komputer.pdf>
- de Araujo, A. L. S. O., Andrade, W. L., & Guerrero, D. D. S. (2016). A systematic mapping study on assessing computational thinking abilities. In *2016 IEEE Frontiers in Education Conference* (pp. 1–9). <https://doi.org/10.1109/FIE.2016.7757678>
- Google for Education. (n.d.). Exploring computational thinking. Retrieved March 15, 2018, from

- <https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189x12463051>
- Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *2016 IEEE Frontiers in Education Conference* (pp. 1-9). <https://doi.org/10.1109/FIE.2016.7757410>
- Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., & Basawapatna, A. (2011). *Computational thinking patterns*. Presented at the 2011 Annual Meeting of the American Educational Research Association (AERA) in the Division C - Learning and Instruction / Section 7: Technology Research Symposium, Louisiana. Retrieved from <https://files.eric.ed.gov/fulltext/ED520742.pdf>
- Jenson, J., & Droumeva, M. (2016). Exploring Media Literacy and Computational Thinking: A Game Maker Curriculum Study. *Electronic Journal of e-Learning*, 14(2), 111-121. Retrieved from <http://www.ejel.org/volume14/issue2>
- 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), 583-596. Retrieved from: [https://www.bjmc.lv/fileadmin/user\\_upload/lu\\_portal/projekti/bjmc/Contents/4\\_3\\_15\\_Kalelioglu.pdf](https://www.bjmc.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf)
- Kassan, S., Fatt, L. K., & Meng, T. Y. (2016). *Asas Sains Komputer Tingkatan 1*. Retrieved from <https://drive.google.com/file/d/1yWAlI8cEzz9xTRXh1S-QHuHGmvBMiVry/view>
- Kong, R., & Wong, G. K. W. (2017). Teachers' perception of professional development in coding education. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering* (pp.377-380). <https://doi.org/10.1109/TALE.2017.8252365>
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Kotini, I., & Tzelepi, S. (2015). A gamification-based framework for developing learning activities of computational thinking. In T. Reiners & L. Wood (Eds.), *Gamification in Education and Business* (pp. 219-252). Cham, Switzerland: Springer. [https://doi.org/10.1007/978-3-319-10208-5\\_12](https://doi.org/10.1007/978-3-319-10208-5_12)
- Li, Y. (2016). Teaching programming based on computational thinking. In *2016 IEEE Frontiers in Education Conference* (pp. 1-7). <https://doi.org/10.1109/FIE.2016.7757408>
- Lowe, T., & Brophy, S. (2017). An operationalized model for defining computational thinking. In *2017 IEEE Frontiers in Education Conference* (pp. 1-8). <https://doi.org/10.1109/FIE.2017.8190682>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Ministry of Education Malaysia. (2016, November 2). Ministry of Education's Ikhtisas Circular Letter Number 9 Year 2016: The Implementation of Secondary School Standard Curriculum in Stages Starting in 2017. Retrieved 4, 2018, from [https://www.moe.gov.my/images/pekeliling/2016/circularfile\\_file\\_001420.pdf](https://www.moe.gov.my/images/pekeliling/2016/circularfile_file_001420.pdf)
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED-Revista de Educación a Distancia (RED)*, 46(10), 1-23 <https://doi.org/10.6018/red/46/10>
- National Research Council. (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. Washington, DC: National Academies Press. <https://doi.org/10.17226/12840>
- National Research Council. (2011). *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*. Washington, DC: National Academies Press. <https://doi.org/10.17226/13170>
- National Science Foundation. (2008). Computational Thinking for Everyone: A Workshop Series. Retrieved December 20, 2017, from [http://www.nsf.gov/awardsearch/showAward?AWD\\_ID=0831827](http://www.nsf.gov/awardsearch/showAward?AWD_ID=0831827)
- National Science Foundation. (2009). Leveraging Thought Leadership for Computational Thinking in the PK-12 Curriculum. Retrieved December 20, 2017, from [http://www.nsf.gov/awardsearch/showAward?AWD\\_ID=0964217&HistoricalAwards=false](http://www.nsf.gov/awardsearch/showAward?AWD_ID=0964217&HistoricalAwards=false)
- National Science Foundation. (2010). Leveraging Thought Leadership for Computational Thinking in the K-12 Curriculum - Phase II. Retrieved December 20, 2017, from [http://www.nsf.gov/awardsearch/showAward?AWD\\_ID=1030054&HistoricalAwards=false](http://www.nsf.gov/awardsearch/showAward?AWD_ID=1030054&HistoricalAwards=false)
- Ota, G., Morimoto, Y., & Kato, H. (2016). Ninja code village for scratch: Function samples/function analyzer and automatic assessment of computational thinking concepts. In A. Blackwell, B. Plimmer, & G. Stapleton (Eds.), *2016 Proceedings of The IEEE Symposium On Visual Languages and Human-Centric Computing* (pp. 238-239). <https://doi.org/10.1109/VLHCC.2016.7739695>
- Park, S., & Jeon, Y. (2015). Teachers' perception on computational thinking in science practices. *International Journal of Education and Information Technologies*, 9, 180-186. Retrieved from <https://www.naun.org/main/NAUN/educationinformation/2015/a462008-075.pdf>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books Inc.
- Repenning, A., Basawapatna, A., & Escherle, N. (2016). Computational thinking tools. In A. Blackwell, B. Plimmer, & G. Stapleton (Eds.), *2016 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 218-222).

- <https://doi.org/10.1109/VLHCC.2016.7739688>
- Rico Lugo, M. J., Olabe, X. B., & Niño, N. M. (2018). "Evolution": Design and Implementation of Digital Educational Material to Strengthen Computational Thinking Skills. *IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje*, 13(1), 37–45. <https://doi.org/10.1109/rita.2018.2809943>
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017) Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72(C), 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Rose, S., Habgood, J., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of e-Learning*, 15(4), 297-309. Retrieved from <http://www.ejel.org/volume15/issue4>
- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (pp. 59–66). <https://doi.org/10.1145/2493394.2493403>
- 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>
- Silva, L. R., da Silva, A. P., Toda, A., & Isotani, S. (2018). Impact of teaching approaches to computational thinking on high school students: A systematic mapping. In M. Chang et al. (Eds.), *IEEE 18th International Conference on Advanced Learning Technologies* (pp. 285–289). <https://doi.org/10.1109/ICALT.2018.00072>
- Sondakh, D. E. (2018, May). Review of Computational Thinking Assessment in Higher Education. Retrieved February 18, 2019, from <http://doi.org/10.13140/RG.2.2.16120.55042>
- The Computer Science Teachers Association and International Society for Technology in Education. (2011). Operational Definition of Computational Thinking for K–12 Education. Retrieved January 15, 2018, from <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>
- Toedte, R. J., & Aydeniz, M. (2015). Computational thinking and impacts on K-12 science education. In *2015 IEEE Frontiers in Education Conference* (pp. 1–7). <https://doi.org/10.1109/FIE.2015.7344239>
- Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training Computational Thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, 5(2), 25-44. <https://doi.org/10.17083/ijsg.v5i2.248>
- Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. In M. Pivec et al. (Eds.), *11th European Conference on Games Based Learning* (pp.687-695). Retrieved from [https://www.researchgate.net/publication/320491120\\_Training\\_Computational\\_Thinking\\_Game-Based\\_Unplugged\\_and\\_Plugged-in\\_Activities\\_in\\_Primary\\_School](https://www.researchgate.net/publication/320491120_Training_Computational_Thinking_Game-Based_Unplugged_and_Plugged-in_Activities_in_Primary_School)
- Turchi, T., & Malizia, A. (2016). Fostering computational thinking skills with a tangible blocks programming environment. In A. Blackwell, B. Plimmer and G. Stapleton (Eds.), *2016 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 232–233). <https://doi.org/10.1109/VLHCC.2016.7739692>
- Vallance, M., & Towndrow, P. A. (2016). Pedagogic transformation, student-directed design and computational thinking. *Pedagogies: An International Journal*, 11(3), 218-234. <https://doi.org/10.1080/1554480x.2016.1182437>
- Wang, D., Wang, T., & Liu, Z. (2014). A Tangible Programming Tool for Children to Cultivate Computational Thinking. *The Scientific World Journal*, 2014, 1–10. <https://doi.org/10.1155/2014/428080>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Weng, X., & Wong, G. K. (2017). Integrating computational thinking into English dialogue learning through graphical programming tool. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pp. 320–325). <https://doi.org/10.1109/TALE.2017.8252356>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Worrell, B., Brand, C., & Repenning, A. (2015). Collaboration and Computational Thinking: A classroom structure. In Z. Li, C. Ermel and S. D. Fleming (Eds.), *2015 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 183–187). <https://doi.org/10.1109/VLHCC.2015.7357215>
- 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*, 14(1), 1–16. <https://doi.org/10.1145/2576872>
- Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In *42nd ACM Technical Symposium On Computer Science Education* (pp. 465–470). <https://doi.org/10.1145/1953163.1953297>
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53(4), 562-590. <https://doi.org/10.1177/0735633115608444>