Informatics in Education, 2020, Vol. 19, No. 4, 701–719

DOI: 10.15388/infedu.2020.30

How to Develop Computational Thinking: A Systematic Review of Empirical Studies

Elif TASLIBEYAZ[1], Engin KURSUN[2], Selcuk KARAMAN[2]
[1]*Erzincan Binali Yildirim University, Education Faculty, Computer Education and Instructional Technologies Department, Erzincan, Turkey*
[2]*Ataturk University, Education Faculty, Computer Education and Instructional Technologies Department, Erzurum, Turkey*
e-mail: etaslibeyaz@erzincan.edu.tr, ekursun@atauni.edu.tr, skaraman@atauni.edu.tr

Received: February 2020

Abstract. The primary purpose of this study is to investigate CT skills development process in learning environments. It is also aimed to determine the conceptual understanding and measurement approaches in the studies. To achieve these aims, a systematic research review methodology was implemented as the research design. Empirical studies on computational thinking indexed in the Web of Science and ERIC databases were selected without constraint on the publication dates. The studies found were examined and a pre-analysis was conducted by the researchers. Following the pre-analysis, 29 articles were selected to be included in the study. Content analysis was applied in order to determine and evaluate the common codes and themes related to the findings. In conclusion, instead of relying on attractiveness, functionality, market share of educational tools (robotic sets, software packets etc.), availability of qualified learning activities focused on problem solving is the main point practitioners should consider.

Keywords: computational thinking, computational thinking interventions, empirical studies, systematic review.

1. Introduction

Computing was previously considered a specialized skill associated with computer scientists or those from similar disciplines. However, now almost everyone is expected to possess basic computing skills in parallel with the developments in today's technology (Kalelioglu, Gulbahar, & Kukul, 2016). Among these skills, computational thinking (CT) is believed to be an important skill to enable future generations (Zengin, 2016), but how to implement this skills requires further investigation.

The published literature contains a broad range of definitions for CT, such as (Wing, 2006), who defines it as an approach to problem solving, systems design, and the understanding of human behaviors based on computer-based concepts whilst (Kazimoglu, Kiernan, Bacon, & Mackinnon, 2012) consider CT as a skill that requires the use of computer systems to solve problems in any discipline. The concept of CT has attracted attention due to the general increase in the importance of computer science. Some studies have examined this concept and revealed different approaches by associating CT with programming skills in computer science (Chaudhary, Agrawal, Sureka, & Sureka, 2016a; Şahiner & Kert, 2016). Thus, studies have largely focused on the effect of programming training on CT. As a result, programming education has been shown to have a positive effect on CT skills (Jaipal-Jamani & Angeli, 2017; Zengin, 2016). Although to gain CT skills actually starts with the use of programming tools, the tools for gaining programming skills rather than producing programs become widespread (Pan, Zhan, & Li, 2016; Sung, Ahn, & Black, 2017)

It is also seen in the literature that it is trying to develop CT skills by making it more attractive using robotic coding (Chaudhary, Agrawal, Sureka, & Sureka, 2016b; (Krugel & Hubwieser, 2017; Leonard vd., 2016).In particular, studies on CT have presented various different ideas about the definition and development of CT skills (Davies, 2008a; Kazimoglu vd., 2012). These differences revealed in the understanding of CT have also been reflected in the interventions for the measurement and development of the related skills. It has also prevented reaching a consensus on how it is developed,even if there are different strategic approaches to the development of CT are used. There is this diversity among countries, regions, schools. Within this range, the approaches of students to improve their CT skills; age (primary, secondary, high school and higher education), environment (schools or private courses), context (STEM, programming lesson, ICT lesson, etc.) varied according to variables.

However, there are various studies that focused on the definition of CT in the literature. Therefore, rather than becoming lost in the variety of CT definitions, it is necessary to focus on a CT skills development approach that accommodates this diversity. One of the concrete steps in creating such an approach is to derive current experience from a review of empirical studies that reflects the practices which include interventions focused on CT skills.

The past decade has seen a dramatic increase in the number of studies conducted about CT (Berland & Wilensky, 2015). Some of these studies are based on qualitative research and literature reviews while others are designed as quantitative research. Most of the review studies examined the characteristics of computational thinking; however, others only examined the definition of CT (Selby & Woollard, 2013) or focused on general trend investigating methodological dimensions and bibliographic features, such as participants, publication date, and methodology in papers (Kalelioglu vd., 2016; Lockwood & Mooney, 2017). In addition, there are some review studies which only examined empirical research investigating methodological trends. For example, Heintz, Mannila, & Färnqvist, (2016) conducted a research study only on empirical studies for K-12 students, in which computational practices and perspectives of the empirical studies were examined. (Hsu, Chang, & Hung, 2018) discussed dimensions, such as learning strategies, participants, teaching tools, programming languages, and course categories of CT education. Thus, it was seen that review studies generally focused on conceptual and methodological trends.

The current study focused on combining the experiences in the practice of improving CT skills in learning environments, by reviewing empirical studies. Previously, CT skills were attributed as technical computing education. Today, it is identified as the training of some thinking skills beyond computing education. CT skills are generally acquired through different computing tools and software as well as content development tools. This study is important because analyzing the interventions used to develop CT skills can guide future computing education practices and studies in the context of both computing for computing skills and computing for thinking skills. In this systematic review, the researchers examined, in detail, the dimensions related to CT skills development, such as contexts, content areas, interventions, measurement tools, variables, and their relationships with each other. The researchers also reviewed the definitions and perceptions of CT in empirical studies since the understanding of CT skills affects both measurement and development of the related skills. This study is unique since it focuses on examining the development process of CT skills rather than categorizing the methodological dimensions of empirical research studies.

The main aim of our study is to examine the CT skills development process in learning environments by reviewing empirical studies. The second aim is to reveal the relationship between the CT skill development process and CT-related dimensions (perspectives, context, content, interventions, etc.). The dimensions examined in the study are given below as research questions:

1. What are critical points of CT definitions?
2. What are perceptions of CT definitions (features of CT emphasized)?
3. What are context of CT?
4. What is intervention tools used in CT skills development?
5. What is data collection tools used to measure CT skills?
6. What are content areas in development of CT skills?
7. What are dependent and independent variables used in the studies and the relationships between them?

2. Methodology

2.1. *Research Design*

A systematic review methodology was employed in this study. According to (Gough, Thomas, & Oliver, 2012), a systematic review is a undertaken to evaluate research literature, using systematic and rigorous methods. Systematic reviews provide a mechanism for identifying the most robust evidence-based research among a range of published studies (Lam & Kennedy, 2005). This method was applied to examine the trends of empirical studies on CT since it utilizes descriptive evaluations conducted on a specific content area (Calik & Sözbilir, 2014).

2.2. *Procedure*

The empirical studies about computational thinking indexed in the Web of Science and ERIC databases were searched without constraint on the publication date. Most relevant studies on CT could be found in these two databases, and are commonly indexed. A total of 40 empirical studies were obtained from the Web of Science database and 68 studies from the ERIC database. In total, 108 studies were obtained using keyword searches for titles including "computational thinking" and topics containing the words "experimental" or "empirical." The obtained studies were examined and subjected to pre-analysis (The pre-analysis study actually includes the process of deciding whether to include the studies (search results in databases) or not) by the researchers. After the pre-analysis, a total of 29 articles were deemed appropriate for the current study.

Fig. 1 shows the distribution of the articles about CT by years. Some studies were eliminated because they were theoretical (not empirical), lacked adequate information about CT, or did not meet the criteria of the goal of the current study. The researchers examined the studies' critical points of CT definitions, perceptions, contexts, content areas, interventions, measurement tools, variables, and their relationships with each other. The procedure of the study is illustrated in Fig. 2.

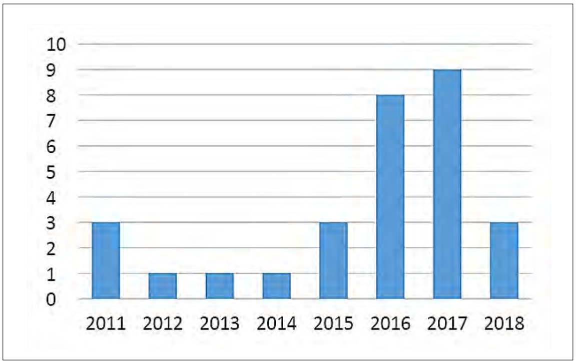

Fig 1. Distribution of the articles about CT by years.

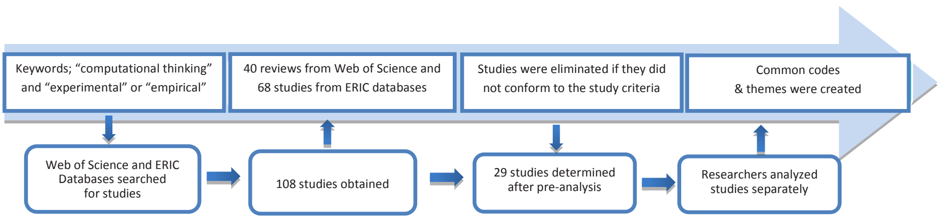

Fig 2. Data collection and analysis process.

2.3. *Data Analysis*

The data collected were analyzed using content analysis. The common codes and themes related to the findings were determined and evaluated during the analysis. Then, the categorizations were revised after a consensus was reached among the researchers. Tables were created from the themes, including the frequency values.

3. Results and Discussion

The frequencies for each code and theme were determined. The findings are presented in this section based on differences in definition, perceptions, CT measurement tools, context, relationships between dependent and independent variables, and relationships between the sample groups, subjects, and study findings. The quotes from the text in the reviewed studies are shown as A1, A2, etc. The names and codes of the articles are given in Attachment-1 at the end of this paper.

3.1. *Definition of CT*

There were definitions of CT in the basis for the interventions to develop CT skills. Therefore, in the systematic analysis, the introduction parts of the reviewed studies were examined, revealing how each of them defined CT was analyzed and the clear differences in the aspects of the given definitions were seen. The concepts in the different definitions of CT were grouped under the categories and frequencies calculated, as shown in Table 1.

The CT definitions in the studies were carefully examined for obvious differences, and the most common keywords were then grouped together. Most of the studies considered CT as analyzing and solving problems (f = 22). Some keywords were not used

Table 1
Frequency of concepts in different definitions of CT

Keywords in CT definitions	f
Analyzing and solving a problem	22
Computer programming	8
Understanding system design	8
Understanding human behavior	8
Logical thinking	5
Algorithmic thinking	4
Development of thinking habits	1
Mathematical thinking	1

alone, but used together in different combinations in the studies. For example, the concepts of “system design” and “human behavior” were included in the reviewed studies together (f = 8). Although there were different definitions of thinking skills, they had a significant place in the definitions when types, such as algorithmic, mathematical, and logical were combined. An important point concerning CT was that its definitions changed according to the contexts. For example, attention was given to programming skills in studies focusing on computer use whereas thinking skills became prominent in the definitions in studies focusing on courses, such as mathematics.

3.2. *Perceptions of CT*

After semantically analyzing the studies, the researchers developed different categories for each study that mentioned the term CT. This part of the study was formed by considering how CT skills were developed, how the research was justified, and how the results were predicted. Thus, there was particular focus on the introduction and conclusion parts of the reviewed studies. The different perceptions of CT found in the studies are presented as frequencies in Table 2.

The content analysis of the CT definitions in the reviewed studies focused on the perceptions of CT; therefore, the most emphasized concepts of CT referenced in the studies were examined in order to reveal which concepts CT was associated with. According to the results, the first three prevailing perceptions were the development of problem solving skills (f = 17), lifelong learning skills (f = 16), and development of programming skills (f = 13). There were also some studies which perceived CT as solving problems with the help of technology (f = 4).

Extracts from the reviewed studies in relation to the emphasis on problem solving and lifelong learning are given below;

> *“…Structured problem solving is vital for preparing students for future academic and professional success. Meanwhile, computational systems have permeated much of modern professional and personal life, making computational thinking an essential skill for members of modern society…” (A1)*

Table 2
Frequency of perceptions of CT

Perceptions of CT	f
Development of problem solving skills	17
Lifelong learning	16
Development of programming skills	13
Algorithmic thinking skill	6
Numerical/logical thinking skill	6
Solution of problems with technology	4

> *"…CT skills are not limited to the field of computing, but are extensible to mathematics, biology, science, economics, reading and other areas…" (A20)*

> *"…everyone should learn about CT considering its influence in many fields of study… (A20)*

3.3. *Context of the Studies*

In order to understand the improvement in CT, the contexts of interventions were examined. The contexts of the reviewed studies were determined by analyzing the methodology sections. The main goal of the interventions and intervention environments was examined to define the context. The various contexts within studies that had been conducted were inferred. The number of studies relating to each context is given in Table 3.

Table 3 shows that the studies were carried out in two main contexts with the most common overall context being programming (f = 17). This result indicated that many of the studies associated CT skills with programming education. However, not only programming was required to develop CT skills but also other contexts were investigated, such as science, technology, engineering and mathematics (STEM) applications (f = 7) and game design (f = 3), as well as learning software and contexts other than programming. For example, in study A29, the students were encouraged to develop their writing skills to support their CT skills.

3.4. *Relationship Between Content Areas and Intervention Tools*

One of the important aspects in the CT development process is the use of intervention tools. In order to clearly define the developmental process, the content areas and intervention tools need to be analyzed together to maintain the relationship between intervention tools and content areas. The results of the analyses were presented by grouping

Table 3
Context of the studies

	Context	f
Programming Context	Programming education	17
	STEM (Science/Math education)	7
Non-Programming Context	Design of games/preferences of games	3
	Learning program/software (Excel-Word etc.)	2
	Pedagogical contexts (Class management etc.)	1
	Interactive writing	1
	Gerontology	1
	Web-mediated learning	1

according to levels. The reviewed articles were mainly found to have been conducted on two groups: K-12 pupils and university students. Content areas were obtained from the title and abstract sections of the studies. The intervention tools in the reviewed articles were also determined by examining the methodology sections. The cross tabulation of student levels by general content areas, intervention tools, and their frequencies are shown in Table 4.

The results of the analysis of the content area revealed the following categories: game design or game preferences, STEM, programming, and other. First, the number of studies was determined for each student level. Then, the intervention tools were grouped into programming-related and non-programming content. Table 4 shows that the studies related to CT skills mostly focused on the programming content area. Moreover, the results indicated that most of the studies were conducted on scratch programming, robot programming, and game design for K-12 students. The studies also focused on CT through the students' writing skills in other content areas.

There were many studies concerning CT skills and programming types (e.g., C++) and the use of programs, such as Excel and Word for university students. In addition, studies were also conducted on university students about other content areas, as well as programming; e.g., CT skills being examined in the interventions of pedagogical courses, such as classroom management.

Table 4

Relationship between content areas and intervention tools in the development of CT skills by student levels

		Programming				Non- programming			
Sample	Content areas	Game	Scratch	Other (C++ etc.)	Robotics	Programs/Software (Word, Excel, Photoshop etc.)	Interactive Writing tool	STEM (Math-Science animation, simulations)	Other (Graphic design etc.)
K-12 Students	Programming	A4, A27	A3, A12, A16, A19, A24	A4	A3, A8				
	STEM							A14, A25 A26, A27	
	Game Design	A9, A23							
	Other						A29		A13, A4
University Students	Programming		A11	A2, A15, A18, A20 A22, A28	A10				
	Use of Programs					A5, A6			
	STEM							A17	
	Other (Gerontology, pedagogical courses, etc.)								A1, A7, A21

3.5. *Relationship Between Content Development Tools and Dependent Variables*

Another important aspect in the CT development process is content development tools. These tools which focus on variables were examined together to clearly understand developmental process of CT. To this end, the methodology sections of the reviewed studies were examined, and the relationships between content development tools and dependent variables were investigated. The findings are presented in Table 5.

As shown in Table 5, the variables obtained from the studies were listed as CT, attitude-motivation, and problem solving, and programming skills. Some of these studies only focused on CT skills, with others concentrating on the variables associated with CT, such as problem solving and attitude-motivation.

Content development tools were also divided into two groups: programming and was non-programming. The programming tools were further divided into four categories of Scratch, games, robotics, and other tools. Table 5 shows that Scratch, robotics, and other tools were prominent among the programming tools. In addition, it was noted that most of the tools were particularly associated with CT and problem solving skills; however, programming tools were rarely associated with programming skills in the studies in Table 5. It has been observed that the target of using programming tools is mostly not to develop programming skills.

Other content development tools were categorized as use of programs, STEM, interactive writing tools, and other. The results showed that use of programs and STEM tools (e.g., animation, simulation in Science/Math Education) were more related to CT skills. The studies about STEM were also conducted based on CT skills, problem-solving skills, and attitude-motivation. In addition, there were various studies in which other content development tools were used to develop CT skills. However, although non-programming content development tools are used to support CT and related skills, programming tools still dominate the development of CT skills.

Table 5
Relationship between content development tools and dependent variables

		CT	Attitude-Motivation	Problem Solving	Programming Skills
Programming Tools	Scratch	A3, A11, A12, A14, A16, A19 A24	A11, A16	A3	A19
	Games	A4, A23, A9	A4, A23,A9	A4, A13	
	Other (C++ etc.)	A2, A18, A22, A28	A28	A2	A18
	Robotics	A3, A8, A9 A10	A10	A3	
Other Development Tools	Use of Programs	A5, A6, A15		A5, A6	
	STEM (Science/Math Education)	A17, A26, A27 A14, A25	A17, A27	A14, A25	
	Interactive Writing Tool	A29	A29		
	Other	A1, A7, A13, A21, A4		A20	

Table 6
CT Measurement tools

	Tool	f
Formative	Performance test	9
	Activity scale	9
	Observation	8
	Interview	4
	Self-evaluation	1
	Total	31
Summative	Survey	6
	Attitude scale	5
	Self-efficacy scale	4
	Achievement test	14
	CT skill scale	3
	Total	32

3.6. *CT Measurement Tools*

Measurement and development processes are complementary elements. In addition, the measurement tools used in the studies are important in terms of giving ideas about the structure of CT interventions and the steps to be taken to improve the related skills. Therefore, the measurement tools used in CT interventions were also included, and the methodology sections of the reviewed studies were examined. The data collection tools used in the studies and their frequencies are shown in Table 6.

As shown in Table 6, in the examined studies, many different measurement tools were used and were grouped as formative and summative. As expected from educational interventions, the most used instrument was an achievement test (f = 14), followed by an activity scale (f = 9) to measure the CT skills of students after an activity. A performance test (f = 9) was used to measure the CT skills during performance whilst observation (f = 8) was used in order to understand whether the target CT skills had been acquired.

The use of achievement tests in the studies was considerably higher than the other measurement tools. However, overall, the use of formative tools was close to that of summative tools. This result shows that formative tools, such as performance test and activity scales are frequently used to measure CT skills since the process of CT skills development takes time.

3.7. *Variables Used in the Studies*

In the reviewed studies, one of the findings concerned the variables which are important for understanding the development process of CT skills. Therefore, the variables before and after the interventions were examined. In addition, in order to understand the interventions clearly, the relationships between the variables were revealed. The

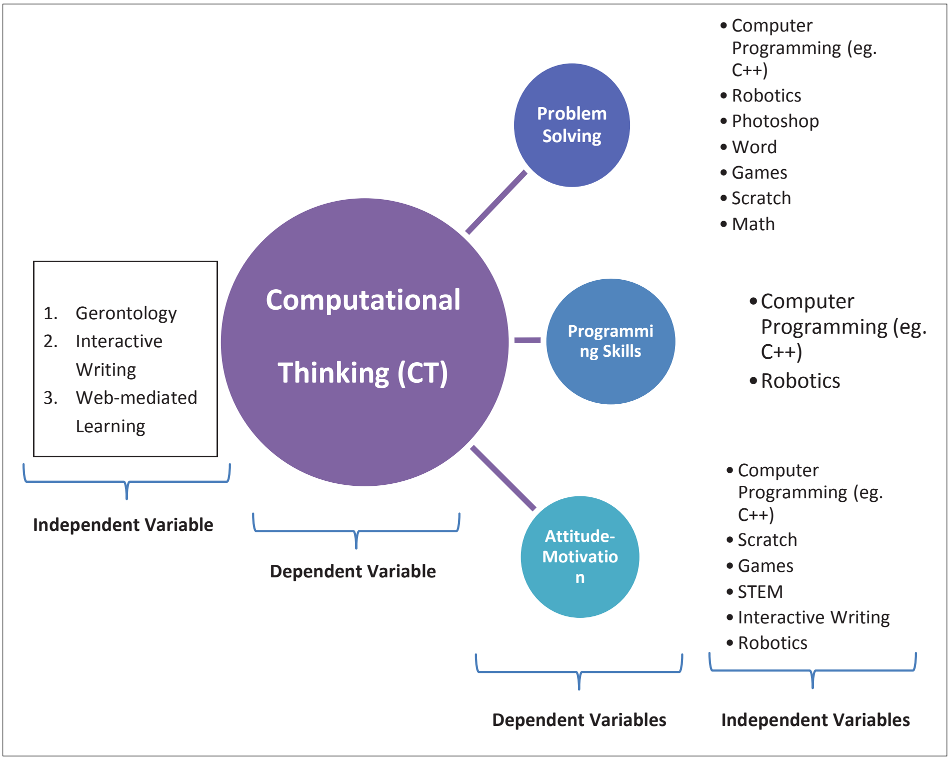

Fig 3. The relationship between dependent and independent variables in the studies.

dependent and independent variables affecting CT were obtained from the examined studies, separated into codes and themes by content analysis, and then the variable groups and their relationships were determined. The relationships between the variables are shown in Fig. 3.

The studies addressed dependent variables, such as CT skills, problem solving, attitude-motivation, and programming skills to measure CT skills. Most of these interventions were based on CT skills and problem-solving variables as previously illustrated in Table 5. However, as shown in Fig. 3, CT was the dependent variable most frequently studied, followed by problem solving, attitude-motivation and programming skills.

The effect of all the independent variables on CT skills is shown in Fig. 3. Independent variables investigated on only CT were gerontology, interactive writing, and web-mediated learning. Non-programming independent variables, such as the use of Photoshop, Word, and games were mostly used in the studies which included problem-solving skills, as well as CT. The studies that included computer programming as an active independent variable examined the effects on all dependent variables (CT skills, problem solving, attitude-motivation, and programming skills). Similar to computer programming, the use of robotics was also associated with most dependent variables.

Table 7
Effects of variables on CT

Other Variables	Type of Effect		
	Negative	No effect	Positive
Computer Programming (e.g., C++)		A2	A12, A13, A14, A16, A18, A20, A22, A26, A28
Robotics			A3, A8, A9, A10
Photoshop			A5
Word-PowerPoint-Excel			A6, A7, A15
Games			A13, A23 A27, A4
Gerontology			A1
Scratch			A11 A14, A19 A24
STEM (Math, Science)			A14, A17, A25
Interactive Writing			A29
Web-mediated Learning		A7	
Gender		A19	A23 (higher in males than females)

3.8. *The Results of the Study*

The results of the study were examined, and the effects of independent variables on CT were evaluated. The results about how these variables affected each other were determined. Table 7 shows the effects of these variables on CT.

In the context of this review, the findings of the selected studies were also examined in terms of related issues of CT skill development process. The independent variables in the studies had a positive effect especially on CT skills (Table 7). In two of the studies, the results showed that there was no effect of computer programming education and web-mediated learning on CT skills (A2 and A7). In terms of some passive variables, gender had different effects on attitudes about programming in some studies. For example, gender could cause differences in the game design content area. Males had more positive attitudes than females about game design content (A23). Moreover, no significant difference was found between males and females in A19.

4. Discussion

There were six main findings of the study: the definition and perception of the CT concept, contexts, the relationship between content area and intervention tools, the relationship between content development tools and dependent variables, measurement tools, independent and dependent variables, and the findings reported in the examined studies:

- Firstly, it was revealed that the literature was focused on problem analysis and problem solving where CT skills were concerned. This may be an indication that the definition of CT is perceived as problem analysis and solution, and the studies approached their research questions in the light of this definition (Chaudhary, Agrawal, Sureka, & Sureka, 2016; Pan vd., 2016). Additionally, the studies showed that researchers' perceptions about CT skills were related to lifelong learning, as well as problem solving (Yang vd., 2011). CT skills were initially perceived as the ability of programming and algorithmic thinking skills (Krugel & Hubwieser, 2017; Leonard vd., 2016). Later, the researchers interpreted that CT skills were perceived as human behaviors and an ability to understand life and to solve problems in life (Pan vd., 2016). CT was even expressed as an essential skill and a process that affects many areas for the members of modern society, thereby emphasizing lifelong learning (A1 and A20). Accordingly, there has been an increase in studies approaching CT skills as lifelong learning and problem-solving skills in recent years.

- The second finding of this study focused on the relationship between content areas and intervention tools by student level. The CT skills of K-12 students were examined in Scratch programming, robot programming, and game design content areas according to the findings. This result can be related to game-based entertainment and practical applications that teach problem solving and CT skills in a systematic way to K-12 students. Game and animation characters in game-based applications (e.g., Scratch) may facilitate learning and increase student engagement (Kert & Uğraş, 2009), with similar results reported in the literature (Oluk & Korkmaz, 2016; Rose, Habgood, & Jay, 2017). There have not been many studies in non-programming content areas undertaken with K-12 students. This may be related to the need for programming applications to help students more easily gain problem-solving skills. Besides programming education, university students have been involved more as research subjects for other content areas and practices in order to develop their CT skills (Kim, Kim, & Kim, 2013; Pan vd., 2016; Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014). This is due to students working on various content areas (e.g., class management in teacher education) in order to improve their problem-solving skills in these areas. It is therefore possible to develop CT and problem-solving skills through interventions related to various content and courses (Davies, 2008; Rodrigues, Andrade, & Campos, 2016; Yang vd., 2011).
- Thirdly, another focus of this research was the relationship between content development tools and dependent variables. According to the research results, empirical studies have mostly dealt with variables, such as problem solving and CT skills. The reason for variables focusing on problem solving, as well as CT skills can be linked to the definition presented that CT skills include problem solving, as previously mentioned. In addition, the results of the present study showed that CT and other skills (problem solving, programming, etc.) as dependent variables were generally related to advance computer programming (e.g., C++), Scratch and robot programming in programming tools. The studies in literature that especially used CT skills as a variable were generally related to Scratch and robot programming in programming tools (Berland & Wilensky, 2015; Jaipal-Jamani & Angeli, 2017; Krugel & Hubwieser, 2017; Malizia, Turchi, & Olsen, 2017). This

may be because in past and present studies, CT skills have been historically associated with programming skills (Calao, Moreno-León, Correa, & Robles, 2015). Programming education was also considered as the most appropriate way to develop CT skills (Barut, Tuğtekin, & Kuzu, 2016). The reason for the investigation into the effect of robot programming on CT skills may be that robotic interventions have increased in recent years (Berland & Wilensky, 2015). This result could also be explained by the effect of robotics programming on CT skills being positive due to its ability to attract students' attention as a study subject. In addition, some studies in recent years aimed at developing CT skills and problem solving in different contexts, such as science education (Garneli & Chorianopoulos, 2018; Jaipal-Jamani & Angeli, 2017). This may be an indication that measuring CT skills is not specific to programming education and can be improved in different contexts (Yılmaz, Yılmaz, & Durak, 2018).

- The fourth finding of our study was about data collection tools used in the reviewed studies. The tools were grouped as summative and formative, with achievement test being the most used measurement method among summative tools. This may be due to variables often being evaluated by one-shot achievement tests in educational empirical studies. According to the results, performance tests and activity scales were also frequently used as measurement tools. This can lead to convincing results as it provides the opportunity to observe and provide feedback to students directly in formative measurement (Boston, 2002; Taras, 2005). A performance test can give a clearer understanding of the development of CT skills because this process needs to occur over a period of time. A performance test can give a clearer understanding of the development of CT skills because this process needs to occur over a period of time. Performance tests are also convenient for CT skills due to the real life nature of CT skills.
- The fifth finding of this study was in relation to the relationship between dependent and independent variables of the studies. Dependent variables, such as CT skills and problem solving have frequently been used in the studies. In addition, computer programming and robotics have been the most commonly used independent variables. This may be an indicator of the importance of computer programming and the increasing interest in robotics in the development of CT skills (Koç & Böyük, 2013; Yolcu & Demirer, 2017), although there are also studies taking into account different variables in interventions for CT skills. However, non-programming independent variables also had an effect on CT skills. These independent variables also showed that the development of CT skills differed from the impact of computer programming.
- The last finding of the study concerned the results of the reviewed studies which showed that the effects of independent variables on dependent variables were generally positive. Some studies in the literature also reported that the aforementioned dependent variables positively affected programming and non-programming subjects (Adler & Kim, 2018; Mouza, Marzocchi, Pan, & Pollock, 2016; Wolz, Stone, Pearson, Pulimood, & Switzer, 2011). Only a few studies obtained

negative results. Only two of these studies reported no significant relationship between programming skills and CT skills (A2 and A7). This may be related to the duration of the training or intervention, student interest, or the quality of the course. Another finding is that in a study about game design, male students had better attitudes toward the study than female students. This may be related to male students being more interested in computer use than females. Furthermore, males tend to be more experienced with computers and higher skill levels in applications, such as programming and games than females (Schumacher & Morahan-Martin, 2001).

5. Conclusion and Suggestions

In the reviewed literature, CT skills were mostly considered as problem-solving and lifelong learning skills, although there was no consensus on the definition and perception of CT skills. However, the lifelong learning that emerged in the definitions and perceptions were not revealed in the intervention designs or measurements in CT skill development processes. Most importantly, real life reflections were not considered in the longitudinal studies, and skill tests were not performed on real-life cases outside the classroom context.

The results showed that the interventions were more often implemented for programming education rather than other subject areas, and CT skills were overshadowed by problem-solving skills which were associated as part of programming education. For K-12 students, the development of CT skills was mostly examined with programming content tools, such as Scratch and robotics. However, there were some interventions (math, games, graphic design in gerontology, etc.) for CT skill development in non-programming content areas besides programming education, especially for university students. Robot programming stood out as an independent variable often used in the studies examined. Still, the results showed that CT skills were not limited to programming education, with a tendency for CT skills to include different skills (especially problem-solving skills) beyond programming skills. CT skills were also measured by performance tests and activity scales which include a process, besides an achievement test, since CT skills development occurs over a period of time.

It is known that applications such as robotics, blocked based programing tools, CS-unplugged activities, which are becoming widespread because of popular expectations and needs. These efforts especially derived from wind of popularity should be designed in line with the pedagogical principles of computing education to improve computing and CT skills. As shown in the findings of this study, for effective computing education, it is important to have qualified learning activities focused on problem analysis and solving rather than focusing on attractiveness, number of functions, market share of educational tools (robotic sets, software packets etc.). Practitioners (school managers, instructors, trainers etc.) should prefer educational tools considering availability of effective ready-to-use activities or potential of tools to develop effective learning activities .

Another suggestion inferred from the study is that the formative assessment should intensively be combined with learning activities. Trainings focusing on development of CT skills, in addition to computing education, content development tools and general STEM applications may be included. At this point, simple programming tools were found to be more suitable for students at K12 level. For older learners, the range of tools for problem solving applications can be expanded.

Suggestions for future studies can be divided into three groups. First, there is a need to develop reliable measurement tools to assess computational thinking levels beside the technical skills based on sound conceptual framework. Second, longitudinal studies for developing computing skills and computational thinking skills can be conducted with different methods. Last, prospective studies focusing on developing computing skills should also consider the CT skills. Future studies about CT should also focus on non-programming tools without being stuck with the gravitational field of popular computing. In addition, an instructional design can be created on CT. As a result,

- CT skills are perceived as a problem solving and lifelong learning skill.
- The development of CT skills is mostly examined with programming content tools, such as Scratch and robotics for K-12 students.
- Studies on the development of CT are carried out in also other content areas besides programming education for university students.
- Formative assessment tools are used to measure CT, since the development of CT skills requires a process.

References

Adler, R. F., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies*, 23(4), 1501–1514.

Barut, E., Tuğtekin, U., & Kuzu, A. (2016). *Programlama Eğitiminin Bilgi İşlemsel Düşünme Becerileri Bağlamında İncelenmesi.* Paper presented at the International Technologies & Teacher Education Symposium, Elazig.

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, 24(5), 628–647.

Boston, C. (2002). The concept of formative assessement: ERIC Clearinghouse on Assessment and Evaluation.

Calao, L., Moreno-León, J., Correa, H., & Robles, G. (2015). Developing mathematical thinking with Scratch. Design for teaching and learning in a networked world (pp. 17–27): Springer International Publishing.

Chaudhary, V., Agrawal, V., Sureka, P., & Sureka, A. (2016). *An experience report on teaching programming and computational thinking to elementary level children using Lego robotics education kit.* Paper presented at the Technology for Education (T4E), 2016 IEEE Eighth International Conference on.

Davies, S. (2008). The effects of emphasizing computational thinking in an introductory programming course.

Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: exploring computational thinking through code analysis. *Interactive Learning Environments,* 26(3), 386–401.

Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology,* 26(2), 175–192.

Kert, S. B., & Uğraş, T. (2009). *Programlama eğitiminde sadelik ve Eğlence: Scratch Örneği.* Paper presented at the The First International Congress of Educational Research, Çanakkale, Turkey.

Kim, B., Kim, T., & Kim, J. (2013). and-Pencil Programming Strategy toward Computational Thinking for Non-Majors: Design Your Solution. *Journal of Educational Computing Research,* 49(4), 437–459.

Koç, A., & Böyük, U. (2013). Fen ve teknoloji eğitiminde teknoloji tabanlı öğrenme: Robotik uygulamaları. *Türk Fen Eğitimi Dergisi,* 10(1), 139–155.

Krugel, J., & Hubwieser, P. (2017). *Computational thinking as springboard for learning object-oriented programming in an interactive MOOC.* Paper presented at the Global Engineering Education Conference (EDUCON), 2017 IEEE.

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance Children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology,* 25(6), 860–876.

Malizia, A., Turchi, T., & Olsen, K. A. (2017). *Block-oriented programming with tangibles: An engaging way to learn computational thinking skills.* Paper presented at the Blocks and Beyond Workshop (B&B), 2017 IEEE.

Mouza, C., Marzocchi, A., Pan, Y.-C., & Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education,* 48(2), 84–104.

Oluk, A., & Korkmaz, Ö. (2016). Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables. *International Journal of Modern Education & Computer Science,* 8(11).

Pan, T., Zhan, G., & Li, Z. (2016). *Exploration on the cultivation of Computational Thinking Ability in the teaching of PhotoShop.* Paper presented at the 2015 1st International Conference on Information Technologies in Education and Learning (ICITEL-15).

Rodrigues, R. S., Andrade, W. L., & Campos, L. M. S. (2016). *Can Computational Thinking help me? A quantitative study of its effects on education.* Paper presented at the Frontiers in Education Conference (FIE), 2016 IEEE.

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.

Schumacher, P., & Morahan-Martin, J. (2001). Gender, Internet and computer attitudes and experiences. *Computers in Human Behavior,* 17(1), 95–110.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning,* 22(3), 443–463.

Taras, M. (2005). Assessment–summative and formative–some theoretical reflections. *British journal of educational studies,* 53(4), 466–478.

Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational thinking and expository writing in the middle school. *ACM Transactions on Computing Education (TOCE),* 11(2), 9.

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), 5.

Yang, H.-I., Martin, P., Satterfield, D., Babbitt, R., Wong, J., Shelley, M., & Chang, C. K. (2011). A novel interdisciplinary course in gerontechnology for disseminating computational thinking.

Yılmaz, F. G. K., Yılmaz, R., & Durak, H. Y. (2018). A review on the opinions of teachers about the development of computational thinking skills in K-12 *Teaching Computational Thinking in Primary Education* (pp. 157–181): IGI Global.

Yolcu, V., & Demirer, V. (2017). A review on the studies about the use of robotic technologies in education. *SDU International Journal of Educational Studies,* 4(2), 127–139.

Attachment 1

A1 A novel interdisciplinary course in gerontology for disseminating computational thinking

A2 The effects of emphasizing computational thinking in an introductory programming course

A3 An experience report on teaching programming and computational thinking to elementary level children using Lego robotics education kit

A4 A serious game for developing computational thinking and learning introductory computer programming

A5 Exploration on the cultivation of computational thinking ability in the teaching of Photoshop

A6 Software application teaching combined with computational thinking

A7 Exploring the effects of web-mediated computational thinking on developing students' computing skills in a ubiquitous learning environment

A8 Comparing virtual and physical robotics environments for supporting complex systems and computational thinking

A9 Using robotics and game design to enhance children's self- efficacy, stem attitudes, and computational thinking skills

A10 Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking

A11 ICT teachers acceptance of scratch as algorithm visualization software

A12 An exploration of the role of visual programmıng tools in the development of young children's computational thinking

A13 Examining the relationship between digital game preferences and computation thinking skills

A14 Introducing computational thinking to young learners: practicing computational perspectives through embodiment in mathematics education

A15 Testing algorithmic skills in traditional and non-traditional programming environments

A16 Development, ımplementation, and outcomes of an equitable computer science after-school program: findings from middle-school students

A17 Enhancing future k8 teachers computational thinking skills through modeling and simulations

A18 Computational thinking as springboard for learning object oriented programmıng in an interactive MOOC

A19 Comparing students' scratch skills with their computational thinking skills in terms of different variables

A20 Can computational thinking help me? A quantitive study of its effect on education

A21 Computational thinking in elementary and secondary teacher education

A22 Block oriented programming with tangibles an engaging way to learn computational thinking skills

A23 Exploring media literacy and computational thinking: a game maker curriculum study

A24 Developing computational thinking abilities instead of digital literacy in primary and secondary school students

A25 Computational thinking in mathematics education: a joint approach to encourage problem solving ability

A26 Teaching computational thinking using agile software engineering methods: a framework for middle schools

A27 Programming video games and simulations in science education: exploring computational thinking through code analysis

A28 Paper-and-pencil programming strategy toward computational thinking for non-majors: design your solution

A29 Computational thinking and expository writing in the middle school

E. Taslibeyaz is an Assistant Professor of computer and instructional technologies department at Erzincan Binali Yildirim University in Turkey. She is studying on video and interactive video usage in medical education and other educational environments.

E. Kursun is currently an Associate Professor at computer and instructional technologies department at Ataturk University in Turkey. He is studying on distance education and ICT usage at educational environments.

S. Karaman is a professor of computer and instructional technologies department at Ataturk University in Turkey. He is studying on distance education and ICT usage at higher education positions they hold at institutions or societies, or any other relevant background information.