Id: 1062305

Participatory Educational Research (PER)
Vol.9(5), pp. 288-304, September 2022
Available online at http://www.perjournal.com
ISSN: 2148-6123
http://dx.doi.org/10.17275/per.22.115.9.5

Tendencies towards Computational Thinking: A Content Analysis Study

Servet Kılıç *

Computer Technologies Department, Technical Sciences Vocational High School, Ordu University, Ordu, Turkey. ORCID: 0000-0002-1687-3231

Article history

Received:
24.01.2022

Received in revised form:
01.02.2022

Accepted:
15.06.2022

Key words:
computational thinking, research tendencies, content analysis, practice-based research

In this research, we analyzed the content of a practice-based research published in SSCI, ESCI and ERIC indexed journals related to Computational Thinking (CT) between 2019 and 2021. For this purpose, we searched Science Direct, Google Scholar and Web of Science databases and examined 97 papers. We evaluated the papers under the headings of development approaches, learning tools, sub-skills, research groups, measurement tools, and prominent findings. According to the results, while for programming, robotics, Science, Technology, Engineering and Mathematics (STEM), development courses and computer science unplugged approaches were adopted in the development of CT, CT was mostly associated with the field of computer science. Programming and robotics software such as Scratch, Lego Mindstorms, M-Bot, Arduino and Bee-Bot are tools with a block-based coding interface. While there was no consensus on the scope and measurement of CT, CT was generally studied within the framework of abstraction, decomposition, algorithmic thinking, and debugging sub-skills. CT developments were measured through scales and tests consisting mostly of multiple-choice and open-ended questions. The research focused on primary and secondary school students while it was limited on preschool level. In addition, studies stating that gender is an effective factor in the development of CT in different age groups are in the majority. Whilst trying to integrate CT into courses in schools, the number of development courses for pre-service and in-service teachers is increasing. Within the framework of the results obtained from the research, the differences in the scope, development, measurement, and evaluation of CT are discussed.

Introduction

With the use of computer in many fields of daily life, the knowledge and skill of using it effectively in order to develop solutions to the problems encountered has gained importance. In the 21st century, where problems are digitized and transferred to the computer environment, it is necessary to have some high-level thinking skills. Problem solving, analytical thinking, critical thinking, algorithmic thinking, and computational thinking (CT) are among these important skills (International Society for Technology in Education [ISTE], 2015). CT skill is the knowledge, skill and competence required for using computers and

* Correspondency: servetkilic@odu.edu.tr

other information processing tools to solve problems encountered in daily life (Özden, 2015). Seymour Papert (1996) referred to the current understanding of CT by emphasizing the use of the computer as a tool to assist thinking. There has been more discussion on CT since Wing (2006) showed that CT was among important basic skills such as reading, writing and arithmetic operations in everyday life, not just for computer science.

Although research on CT has increased in recent years, there is no consensus on the definition, scope, development and evaluation of CT (Weintrop et al., 2016). Román-González, Pérez-González and Jiménez-Fernández (2017) categorized CT-related definitions into general definitions (Aho, 2012; Wing, 2006), operational definitions (CSTA & ISTE 2011), and curriculum-related definitions (Brennan & Resnick, 2012). Tang, Yin, Lin, Hadad and Zhai (2020) classified CT as definitions related to computer science (Brennan & Resnick, 2012; Weintrop et al., 2016) and problem-solving skills (CSTA & ISTE 2011; Selby & Woollard, 2013). Within these definitions, different sub-skills revealing the development of CT are also included. Researchers have stated that skills such as algorithmic thinking, abstraction, decomposition, generalization, and debugging are a part of CT (Grover & Pea, 2013). Hsu, Chang and Hung (2018) addressed CT within the framework of 19 different sub-skills, also including abstraction. Stating that CT is the ability to design planned, systematic and reproducible methods to solve problems, the researchers associated CT with high-level thinking skills such as decomposition, pattern recognition, abstraction and algorithmic thinking skills (Rowe, Asbell-Clarke, Baker, Gasca, Bardar, & Scruggs, 2018; CSTA, 2017; Wing, 2006). As can be seen, different sub-skills used within the framework of different definitions can also expand the scope of CT.

The inclusion of different sub-skills showing the development of CT in research also differentiates the development approaches, learning environments and tools used. Programming (Kong & Wang, 2019; Zhang & Nouri, 2019), robotics (Atmatzidou & Demetriadis, 2016; Relkin, de Ruiter, & Bers, 2021), CS unplugged activities (Brackmann, Román-González, Robles, Moreno-León, Casali, & Barone, 2017; Sun, Hu, & Zhou, 2021), and STEM applications (Sırakaya, Alsancak Sırakaya, & Korkmaz, 2020; Sun, Hu, Yang, Zhou, & Wang, 2021) are improving CT. In these studies, different measurement and evaluation tools were used for the development and evaluation of CT. These tools changed as process and result oriented (Yeni, 2018). Scales (Kılıç, Gökoğlu, & Öztürk, 2021; Korkmaz, Çakir, & Özden, 2017; Kukul & Karataş, 2019; Yağcı, 2019) and tests (Atmatzidou & Demetriadis, 2016; Chen, Shen, Barth-Cohen, Jiang, Huang, & Eltoukhy, 2017) were generally used in outcome-based assessments. In process-oriented evaluations, different evaluation models were preferred (Basawapatna, Koh, Repenning, Webb, & Marshall, 2011; Brennan & Resnick, 2012; Koh, Basawapatna, Bennett, & Repenning, 2010; Seiter & Foreman, 2013). In the studies in which CT development was evaluated through programming within the framework of models, evaluations were made by examining the codes related to the activities performed by the students or the games they designed using some software (Dr. Scratch, Scrape) (Ma, Zhao, Wang, Wan, Cavanaugh, & Liu, 2021; Moreno-León, Robles, & Román-González, 2015). Again, in result-oriented assessments, open-ended, multiple-choice, or fill-in-the-blank tests were developed and applied to students (Atmatzidou & Demetriadis, 2016; Chen et al., 2017). In the process, rubrics were generally used to determine the development levels of CT-related skills (Chen et al., 2017).

With the increasing understanding of the importance of CT after the research, while policy makers and researchers debate on how to integrate CT into their curriculum (Bocconi et al., 2016; So, Jong, & Liu, 2020), educators, on the other hand, have started to make applications in the courses for the development of CT skills of students in different age groups. Various attempts are being made to integrate CT and computer science into primary, secondary and high school curricula in many countries such as the USA, UK and Australia (Passey, 2017). The fact that CT is seen as a basic skill used in daily life (Wing, 2006) has also accelerated research on CT development of students at different levels from pre-school to higher education. However, since CT is a new concept, it has been stated that teachers have some lack of knowledge about what exactly this skill is, how it is developed, how it is evaluated and how it can be integrated into the courses (Hsu et al., 2018; Mannila et al., 2014; Yadav, Gretter, Good, & McLean, 2017). Recent research has focused on teacher education (Baroutsis, White, Ferdinands, Lambert, & Goldsmith, 2019; Umutlu, 2021) and CT development of younger students (del Olmo-Muñoz, Cózar-Gutiérrez, & González-Calero, 2020, Wang, Choi, Benson, Eggleston, & Weber, 2021). With increasing research, the trends of CT are also discussed. As CT is seen as a basic skill and its importance has been revealed by researchers, it has also increased the efforts to include this concept in the curriculum of students at different levels. We anticipate that it will also support efforts to reveal common points in research, include CT in curricula and apply it to classrooms. For this reason, examining the research in the literature, especially in practice-based and high-indexed journals, will guide policy makers, practitioners, and educators. There has been previous systematic review or content analysis research on CT. Haseski and İliç (2019) conducted a content analysis study to determine the data collection tools used to measure CT. Kalelioğlu (2018) systematically examined the research on CT and revealed the trends of them. Tang et al. (2020) conducted content analysis research to reveal the international publication trends and research typology related to CT. Since CT is a new and rapidly developing concept, it is important to examine the research conducted at more frequent intervals. In addition, it is necessary to examine the relevant research in a wider framework and to reveal the common points of the practices adopted in them. This research evaluates the development approaches adopted, the learning tools used, the sub-skills addressed, the research groups applied, the measurement tools used, and the results obtained regarding CT. In this context, we attempted to answer the following research questions:

- RQ1: What are the development approaches adopted regarding CT?
- RQ2: What are the learning tools used in the development of CT?
- RQ3: How are the research groups in which CT-related applications are made?
- RQ4: What are the sub-skills covered in CT?
- RQ5: What are the preferred measurement tools for the evaluation of CT?
- RQ6: What are the prominent results pertaining to the research on CT?

Method

In this research, we examined the practice-based research (experimental, case) published on CT between 2019 and 2021 and analyzed the content of the research. We did not include survey research, scale development and systematic review research. Content analysis is an effective method for summarizing, classifying, comparing, and expressing the findings obtained in research numerically (frequency, percentage, mean, and alike) (Chen, Monion, & Morrison, 2007).

Sample

We searched Science Direct, Google Scholar and Web of Science article indexes in order to reach academic publications on CT. We used the keyword "Computational Thinking" in these indexes. By filtering according to the years 2019 and 2021, we accessed the article contents published in full text. In accordance with the purpose of the research, we conducted content analysis of a total of 97 articles scanned in SSCI, ESCI and ERIC indexes.

Data collection tool

We examined the topics covered in the content analyzes related to CT (Haseski & İlic, 2019; Kalelioğlu, 2018; Tang, Chou, & Tsai, 2020). We determined the main elements of the publication classification form according to the development approaches, sub-skills and measurement tools that emerged in these studies. We examined the research that made content analysis using the publication classification form (Çiltaş et al., 2012; İslamoğlu et al., 2015). We created the publication classification form using the Excel spreadsheet application (Appendix-A). The form was reviewed by another researcher conducting research in the field of computer and instructional technologies, and necessary arrangements were made. In the first part of the form, descriptive information such as article title, article year and article index is included. In the other sections, there are topics related to development approaches, learning tools, sub-skills, research groups, measurement tools and results.

Data Analysis

According to the sampling criteria, the researcher first made the first classification of 97 articles with the publication classification form. Then, 10 publications randomly selected from the bulk of research were sent to another male researcher. The classification consistency of two researchers over 10 articles was compared. As a result of the evaluation, we determined the reliability of the classifications as 85%, according to the Huberman and Miles (2002) safety level formula (reliability=consensus/(consensus+disagreement)). We calculated the consistency of the headings in the classification form separately, and in general, the consistency of all categories was over 75%. The fact that this rate is over 70% indicates that the coding is consistent, and the research is reliable (Huberman & Miles, 2002). By transferring the data obtained from the classification form to the SPSS statistical program, we revealed the frequency values and percentiles of the codes. We transferred the results of the research to a word processing program and carried out the coding on the Nvivo software. The first researcher coded. Then the second researcher made the coding. A common code list was created by comparing the resulting codes. Both researchers recoded according to this code list. Cohen's Kappa reliability coefficient of consistency between coding was calculated as 0.80. The resulting value indicates that the agreement between encoders is at a good level (Landis & Koch, 1977).

Results

In this research, we examined 16 (16%) practice-based research conducted in 2019, 31 (32%) conducted in 2020, and 50 (52%) conducted in 2021. Most of the reviewed articles (76%) were published in SSCI indexed journals. Other articles (24%) were published in ESCI and ERIC indexed journals. The publication of articles mostly in SSCI indexed journals indicates that serious research has been carried out on CT and that the importance given to CT is gradually increasing.

Approaches Adopted for the Development of CT

We report the different approaches used to enable the development of CT (Figure 1).

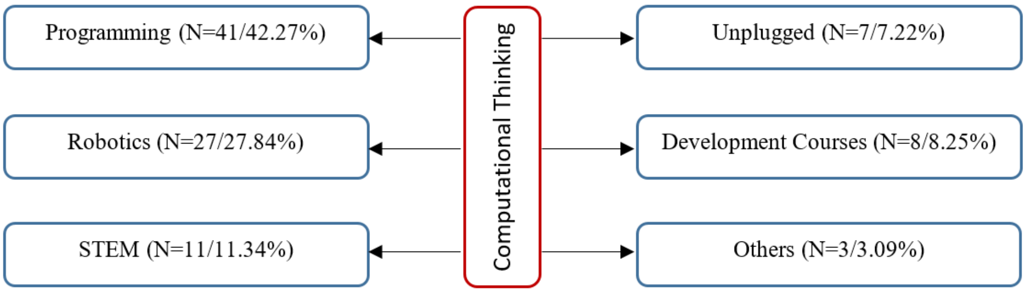

Figure 1. Approaches used to development of CT

Programming, robotics, STEM, unplugged and development courses are among the approaches to CT development. Programming and robotics are among the most used approaches in CT development.

Learning Tools Used for the Development of CT

In the research, different coding interfaces and learning tools were preferred in teaching programming and robotics (Table 1).

Table 1. Coding interfaces and learning tools used in programming and robotics education

	Programming				Robotics					
	Block-based			Hybrid	Block-based					Text-based
	Scratch	Code.org	Others (Pencil Code, Toys, Alice, Blockly, Code Monkey etc.)	Others	mBot (Scratch)	Lego (EV3, WeDo)	Bee-Bot	Arduino	Others (Kibo, Mico:bit)	Arduino
Pre-school			1		1		3	1		
Primary school	7	3	5		2		1	1	2	
Secondary school	6		5	2	1	4		4		3
High school	1		4	1				2		
Undergraduate/Pre-service teachers	3	1	1			2				
In-service teachers	3				2	1			1	
Total	20 27%	4 5%	16 22%	3 4%	6 8%	7 9%	4 5%	8 11%	3 4%	3 4%

Block-based coding tools have been widely used in programming and robotics education. While Scratch is used for programming education in almost all age groups, it is used more in primary and secondary school student groups. In robotics education, mBot, Mindstorms and Arduino robotic kits, which can be coded with a block-based coding interface, are generally used. In text-based programming education, Arduino robotic kits are mostly preferred. Bee-Bot is partially used in small age groups such as pre-school and primary school.

Research Groups Focused on CT Research

The CT development approaches shown in Figure 1 were applied to different age groups at different levels of education (Figure 2).

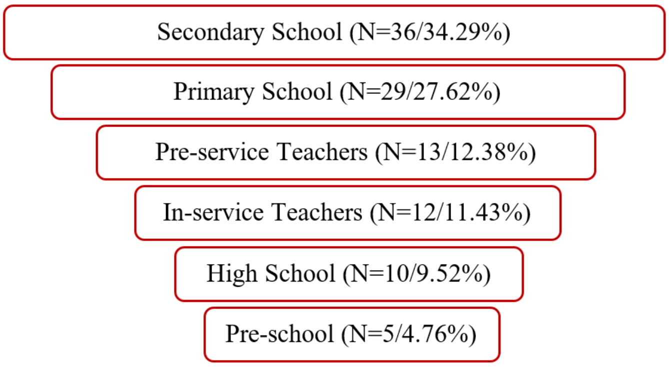

Figure 2. Research groups focused on CT research

In the last three years, practice-based research on CT has focused more on primary and secondary school students (62%). Total research on pre-service and in-service teachers constituted approximately 25 percent of all research. It can be said that the studies on preschool student groups are more limited.

Sub-skills Addressed in CT Research

In research using different development approaches, the sub-skills were differentiated (Table 2).

Table 2. Sub-skills addressed within the framework of CT development approaches

Sub-skills	Programming	Robotics	STEM	Unplugged	Total Frequency	Percent
Algorithm design/ thinking	19	10	6	6	41	22%
Decomposition	12	3	4	4	23	12%
Debugging	8	7	2	3	20	11%
Abstraction	9	3	3	3	18	10%
Concepts/ Sequencing	8	3		1	12	6%
Pattern recognition	6	2	3		11	6%
Problem solving	4	1	3	1	9	5%
Creativity	4	1	3	1	9	5%
Concepts, Practices, and Perspectives	7				7	4%
Generalization	2	3	1	1	7	4%
Cooperativity/ Collaboration	2	1	3	1	7	4%
Evaluation	2	1	1	3	7	4%
Critical Thinking	2	1	3	1	7	4%
Logical inquiry/ Logic/ Conditional logic	3	1			4	2%
Data Representation	2	1			3	1%
Parallelism	2				2	-
Modularity	1	1			2	-

Algorithmic thinking, decomposition, abstracting and debugging skills are among the skills that were examined more. Concepts, practices, and approaches model introduced by Brennan & Resnick (2012) was used in programming activities. Concepts and Sequencing skills revealing the conceptual structure in programming and robotics applications, were examined.

Preferred Measurement Tools for Evaluation of CT

The measurement tools used in the research are shown in Figure 3.

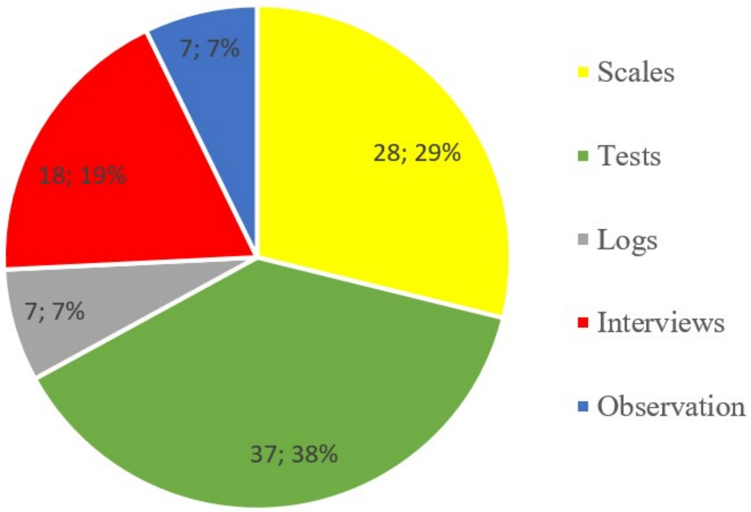

Figure 3. Measurement tools used in the research

In order to measure their CT development, mostly open-ended questions, multiple choice tests or fill-in-the-blank tests (38%) were preferred. Among these tests, block-based programming tests, bebras activities and tests developed by researchers themselves about daily life are common (González, 2015; Román-González et al.,2017). Another frequently used tool to measure CT development is scales (29%). Sometimes students' coding logs are also used in programming-based learning environments. Interview and observation forms are mostly used to explain and support the data obtained for CT developments.

Prominent Results in CT Research

Content analyzes of 97 research results conducted based on the application were made. The variables having a positive effect on the development of CT skills were coded and their frequency values were extracted (Figure 4).

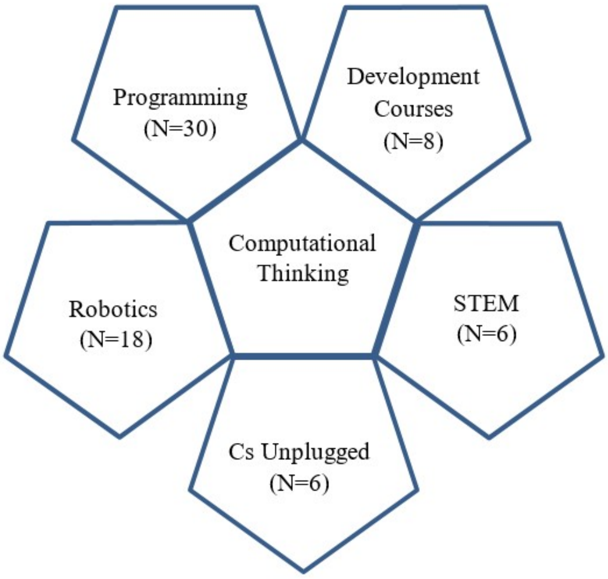

Figure 4. Approaches that positively affect CT development

70 percent of research has shown that the approaches adopted have a positive impact on the development of CT. Research results also revealed some factors that are effective in the development of CT. Studies showing that gender differences are effective in the development of CT (N=7) are more than studies showing that it is not effective (N=3). After the applications, it was revealed that boys (N=5) showed more improvement than girls (N=2).

Discussion

It has been observed that the articles published between 2019 and 2021 are concentrated in SSCI indexed journals. Educators, policy makers, and practitioners are increasingly recognizing the importance of CT (Saxena, Lo, Hew, & Wong, 2020), and both efforts to include computer science-related fields (programming, robotics) in curricula (Bocconi et al., 2016) and the increase in STEM applications in schools (Rich, Yadav, & Larimore, 2020) provide clues that the importance given to CT will increase.

Research on CT has generally included programming, STEM, robotics, development courses and CS Unplugged activities. Weinberg (2013) stated that CT could be developed through applications such as unplugged activities, block-based programming, game design, educational robotics and STEM. In recent years, researchers have generally preferred these approaches for the development of CT. Programming is a popular practice used to teach CT concepts, practices, and perspectives (Brennan & Resnick, 2012; Lye & Koh, 2014; Román-González et al., 2017). In addition, robotics plays an important role in the development of students' problem solving, creative thinking (Karim, Lemaignan, & Mondada, 2015), learning programming (Numanoğlu & Keser, 2017), STEM and CT skills (Becker, Cummins, Davis, Freeman, Hall, & Ananthanarayanan, 2017). In some courses, CT concepts are integrated into STEM applications (Weintrop et al., 2016). CS Unplugged activities are preferred for beginners and younger age groups to gain programming logic and develop CT skills (Bakala, Gerosa, Hourcade, & Tejera, 2021; del Olmo-Muñoz et al., 2020). Apart from these studies, there are also CT development courses for pre-service and in-service teachers in the articles. Since CT is a new concept, it is stated that teachers have some difficulties in teaching and integrating this concept into their courses (Hsu et al., 2018; Mouza, Pan, Yang, & Pollock, 2017; Yadav et al., 2017). For this reason, in recent years, there has been an increasing number of courses organized on what the scope of CT is and how to teach it to pre-service and in-service teachers (Umutlu, 2021).

The inclusion of different definitions of CT in the literature causes different views on the scope of CT (Weintrop et al., 2016). CSTA and ISTE (2011) emphasized the 9 sub-skills of CT; data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation. Rose, Habgood and Jay (2017) stated that components such as abstraction, algorithms, data, problem decomposition, parallelism, debugging, testing and control structure were the most commonly used sub-skills. This research has shown that sub-skills such as algorithmic thinking, decomposition, abstraction, and debugging are generally considered in common. Although Kalelioglu, Gulbahar, and Kukul (2016) expressed the full disclosure of the definition and scope of CT as a difficult goal, researchers and educators expressed abstraction, algorithmic thinking, decomposition and debugging as common skills reflecting CT (Barr & Stephenson, 2011; Grover & Pea, 2013; Lee et al., 2011; Rich et al., 2020; Yadav, Larimore, Rich, & Schwarz, 2019). Selby and Woollard (2013) reviewed CT research conducted between 2006 and 2013 to contribute to the debate about the definition of CT. They stated that the terms abstraction, decomposition, algorithmic thinking, evaluation and generalization are skills that are frequently used in definitions. Angeli et al. (2016) stated that for the CT development of primary school students, the skills of stripping, generalizing, decomposition, algorithmic thinking and debugging should be included in the curriculum. The common CT subskills demonstrated by the results of this research are similar to the CT subskills mentioned earlier in the literature. In addition to CT basic skills, the concepts, practices, and approaches model introduced by Brennan and Resnick (2012) were used in block-based programming activities. While Kong (2016) stated that this framework covered CT comprehensively, Nouri, Zhang, Mannila, and Norén (2020) stated that this framework provided a theoretical basis for the block-based visual programming language. In programming and robotics education, tools with a block-based coding interface are generally preferred. Scratch has become a software frequently used by all age groups. However, it is widely preferred in primary and secondary school levels. Scratch, which can work on mobile devices with different operating systems apart from computers, is used in 150 different countries and more than 60 languages around the world (Scratch, 2021). The Scratch program was used for CT development in previous years and positive results were obtained after the applications (Brennan & Resnick, 2012; Moreno-León et al., 2015). In robotic programming applications, mBot, Lego Mindstorms, Arduino and Bee-Bot robotic kits, which could be coded as block-based, were used. Arduino, which can also be coded as text-based, is preferred because it is an open-source software and easy for use, is open to access to different projects and many different electronic parts, especially different sensors (Arduino, 2021). Lego Mindstorms' NXT and EV3 series are among the robotics kits that are widely used in different education levels (Oluk & Korkmaz, 2018; Üçgül, 2018). Since mBot Scratch can be used with a coding interface, it can be used easily in primary and secondary school age levels, while Bee-Bot is mostly used in preschool student groups because it has an icon-based visual coding interface without the need for reading and writing knowledge.

With the increasing importance of CT, research has focused more on primary and secondary school students. While the research conducted by organizing courses for teachers to introduce CT is increasing, the research conducted for pre-school students is quite limited. Buitrago Flórez, Casallas, Hernández, Reyes, Restrepo and Danies (2017) state that CT should be taught to students at an earlier age in order to ensure their cognitive development. There are studies supporting that students can acquire CT skills at primary school level (Hsu et al., 2018; Lye & Koh 2014; Shute, Sun, & Asbell-Clarke, 2017). Djurdjevic-Pahl, Pahl, Fronza and El Ioini (2017) stated that CT could be taught at an early age starting from the first year of primary school. Due to the fact that programming learning and teaching activities are given more place at the university level, the research carried out in previous years mostly focused on the undergraduate level (Kalelioğlu, 2018). With the increase in programming activities carried out with primary and secondary school students in recent years, it is thought that CT research will be frequently included in these levels (Nouri et al., 2020). Although some studies emphasize the importance of CT development in early childhood (del Olmo-Muno et al., 2020; Djurdjevic-Pahl et al., 2016; Espino & González, 2016), the results show that studies on preschool are rather limited compared to others. As the importance of CT becomes better understood over time, it is anticipated that future research will tend towards these age groups.

In recent years, studies revealing the effect of the gender factor on the development of CT have been increasing. It is stated that the CT skills of boys develop more than girls (Esteve-Mon et al., 2020; Polat et al., 2021; Mouza et al., 2020). There are also studies showing that gender is not effective in the development of CT (Alsancak, 2020; del Olmo-Muñoz et al., 2020). When students start secondary school, gender differences and inequalities in prior knowledge and experience with computers become more evident (Ardito, Czerkawski, & Scollins, 2020; Witherspoon et al., 2017). The fact that male students have a slightly higher interest in computer science and technical subjects than female students may cause this situation (Polat et al., 2021).

There are also different views on how to evaluate CT. In this study, it was revealed that CT was evaluated mostly through scales and tests consisting of multiple choice and open-ended questions. It is seen that interview forms have been frequently used to support the data obtained with these data collection tools. These forms are also used to reveal the knowledge of pre-service and in-service teachers about CT (Li, 2021; Nouri et al., 2020). In addition to interview forms, observation forms are also used to monitor and evaluate students' activity processes (Herro et al., 2021; Mouza et al., 2020). Researchers state that there is uncertainty about how to evaluate CT (Lockwood & Mooney, 2018; Hsu et al., 2018). In her research, Yeni (2018) stated that students benefited from writing code, tests, observation, code block interpretation, ordering code blocks and open-ended questions to determine their level of understanding of programming terms depending on CT. While tests consisting of multiple choice and open-ended questions were used frequently in previous research (Atmatzidou & Demetriadis, 2016; Chen et al., 2017; Grover et al., 2014), scales for different student levels were also developed by researchers (Korkmaz et al., 2017; Kukul & Karataş, 2019; Yağcı, 2019; Kılıç et al., 2021). Especially in block-based programming environments such as Scratch, software such as Dr Scratch and Scrape are also used to evaluate the code sets quickly created by the students and to provide quick feedback to the students (Moreno-León et al., 2015; Wolz, 2011). Kalelioğlu (2018) also stated in his content analysis study that CT scales and tests were the most used data collection tools. While it is seen in the literature that tests and scales are mostly used in the evaluation of CT, it can be said that similar evaluation methods are also preferred in new studies.

Conclusion

In recent years, serious practice-based research on CT has been carried out and the importance of CT is better understood. While discussions on the definition, scope, development, measurement, and evaluation of CT still continue, the results of this research show that there is no consensus on this point. While CT development is evaluated within the framework of many different sub-skills, abstraction, decomposition, algorithmic thinking and debugging skills are considered common in different development approaches. It is important to address and evaluate these sub-skills primarily in research. The use of tests consisting of open-ended or multiple-choice questions that relate the subjects to daily life or the use of scales with validity and reliability are seen as effective measurement tools in the evaluation of CT developments. Although disagreements are not seen as an obstacle to the progress of CT, teachers who work at different levels, starting from pre-school, have important responsibilities to better understand CT. At this point, it is important to increase the development courses given to teachers within the scope of programming, robotics, STEM and CS Unplugged in the coming years in order to accelerate the integration of CT into courses. We especially attach importance to the fact that teachers related to the field of computer science have content knowledge in the use of Scratch, Lego Mindstorms, Arduino, mBot or other different programming and robotic learning tools. Research on pre-school student groups is limited, and at this point, it is necessary to increase practice-based research. Since gender differences appear to be influential in the development of CT, teachers need to take these differences into account in their classroom practices.

References

Aho, A. V. (2012). Computation and computational thinking. *The computer journal*, *55*(7), 832-835. https://doi.org/10.1093/comjnl/bxs074

Alsancak, D. (2020). Investigating computational thinking skills based on different variables and determining the predictor variables. *Participatory Educational Research*, *7*(2), 102-114. https://doi.org/10.17275/per.20.22.7.2

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. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1040.5105&rep=rep1&type=pdf

Ardito, G., Czerkawski, B., & Scollins, L. (2020). Learning computational thinking together: Effects of gender differences in collaborative middle school robotics program. *TechTrends*, *64*(3), 373-387. https://doi.org/10.1093/comjnl/bxs074

Arduino (2021). About Arduino. https://www.arduino.cc/en/about.

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

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, 100337. https://doi.org/10.1016/j.ijcci.2021.100337

Baroutsis, A., White, S. L., Ferdinands, E., Lambert, E., & Goldsmith, W. (2019). Computational thinking as a foundation for coding: Developing student engagement and learning. *Australian Primary Mathematics Classroom*, *24*(2), 10-15. https://eprints.qut.edu.au/130998/2/130998.pdf

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

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011, March). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 245-250). https://doi.org/10.1145/1953163.1953241

Becker, S. A., Cummins, M., Davis, A., Freeman, A., Hall, C. G., & Ananthanarayanan, V. (2017). NMC horizon report: 2017 higher education edition. USA: The New Media Consortium. https://www.learntechlib.org/p/174879/

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education (Policy Report). Spain: European Commission. https://komenskypost.nl/wp-ontent/uploads/2017/01/jrc104188_computhinkreport.pdf

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65-72). https://doi.org/10.1145/3137065.3137069

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25). http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834-860. https://doi.org/10.3102/0034654317710096

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162-175. https://doi.org/10.1016/j.compedu.2017.03.001

Computer Science Teachers Association [CSTA], & International Society for Technology in Education [ISTE]. (2011). Computational thinking in K-12 education leadership toolkit. Retrieved October 9, 2019, from https://id.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

CSTA (2017). CSTA K-12 Computer Science Standards Revised. Retrieved December 20,2017,from https://www.doe.k12.de.us/cms/lib/DE01922744/Centricity/Domain/176/CSTA%20Computer%20Science%20Standards%20Revised%202017.pdf

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Deng, W., Pi, Z., Lei, W., Zhou, Q., & Zhang, W. (2020). Pencil Code improves learners' computational thinking and computer learning attitude. *Computer Applications in Engineering Education*, *28*(1), 90-104. ttps://doi.org/10.1002/cae.22177

Djurdjevic-Pahl, A., Pahl, C., Fronza, I., & El Ioini, N. (2016, October). A pathway into computational thinking in primary schools. In *International symposium on emerging technologies for education* (pp. 165-175). Springer, Cham.

Espino, E. E. E., & González, C. G. (2016, September). Gender and computational thinking: Review of the literature and applications. In *Proceedings of the XVII International Conference on Human Computer Interaction* (pp. 1-2). https://doi.org/10.1145/2998626.2998665

Esteve-Mon, F., Llopis, M., & Adell-Segura, J. (2020). Digital competence and computational thinking of student teachers. *International Journal of Emerging Technologies in Learning (iJET)*, *15*(2), 29-41. https://doi.org/10.3991/ijet.v15i02.11588

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, *29*(1), 12-28. https://doi.org/10.1002/cae.22255

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Haseski, H. I., & İlic, U. (2019). An investigation of the data collection instruments developed to measure computational thinking. *Informatics in Education*, *18*(2), 297-319. https://doi.org/10.15388/infedu.2019.14

Herro, D., Quigley, C., Plank, H., & Abimbade, O. (2021). Understanding students' social interactions during making activities designed to promote computational thinking. *The Journal of Educational Research*, *114*(2), 183-195. https://doi.org/10.1080/00220671.2021.1884824

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Huberman, A. M., & Miles, M. B. (2002), *The qualitative researcher's companion*, Sage, Thousand, Oakes, CA.

İslamoğlu, H., Ursavaş, Ö.F., & Reisoğlu, İ. (2015). A content analysis of the academic work on the FATİH project. *Educatıonal Technology Theory Andpractıce*, *5*(1), 161-183. https://doi.org/10.17943/etku.28463

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583–596. http://acikerisim.baskent.edu.tr/bitstream/handle/11727/3831/4_3_15_Kalelioglu.pdf?sequence=1

Kalelioğlu, F. (2018). Characteristics of studies conducted on computational thinking: A content analysis. In *Computational thinking in the STEM disciplines* (pp. 11-29). Springer, Cham.

Karim, M. E., Lemaignan, S., & Mondada, F. (2015, June). A review: Can robots reshape K-12 STEM education? In *2015 IEEE international workshop on Advanced robotics and its social impacts (ARSO)* (pp. 1-8). IEEE. https://doi.org/10.1109/ARSO.2015.7428217

Kılıç, S., Gökoğlu, S., & Öztürk, M. (2021). A Valid and Reliable Scale for Developing Programming-Oriented Computational Thinking. *Journal of Educational Computing Research*, *59*(2), 257-286. https://doi.org/10.1177/0735633120964402

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010, September). Towards the automatic recognition of computational thinking for adaptive visual language learning. In *2010 IEEE symposium on visual languages and human-centric computing* (pp. 59-66). IEEE. https://doi.org/10.1109/VLHCC.2010.17

Kong, S. C., & Wang, Y. Q. (2020). Formation of computational identity through computational thinking perspectives development in programming learning: A mediation analysis among primary school students. *Computers in Human Behavior*, *106*, 106230. https://doi.org/10.1016/j.chb.2019.106230

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

Kukul, V., & Karatas, S. (2019). Computational thinking self-efficacy scale: Development, validity and reliability. *Informatics in Education*, *18*(1), 151-164. https://doi.org/10.15388/infedu.2019.07

Landis, J. R., & Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, *33*(2), 363-374. https://doi.org/10.2307/2529786

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37. https://doi.org/10.1145/1929887.1929902

Li, Q. (2021). Computational thinking and teacher education: An expert interview study. *Human Behavior and Emerging Technologies*, *3*(2), 324-338. https://doi.org/10.1002/hbe2.224

Lockwood, J. & Mooney, A. (2018). Developing a computational thinking test using bebras problems. TACKLE: The 1st systems of assessments for computational thinking learning. *Workshop at EC-TEL 2018 Conference*, Leeds, United Kingdom. https://mural.maynoothuniversity.ie/10316/1/AM-Developing-2017.pdf

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

Ma, H., Zhao, M., Wang, H., Wan, X., Cavanaugh, T. W., & Liu, J. (2021). Promoting pupils' computational thinking skills and self-efficacy: A problem-solving instructional approach. *Educational Technology Research and Development*, 1-18. https://link.springer.com/content/pdf/10.1007/s11423-021-10016-5.pdf

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L, & Settle, A. (2014, June). Computational thinking in K-9 education. In T. J. Cortine (Ed.), In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology In Computer Science Education Conference* (pp. 1-29). NewYork: Association for Computing Machinery. https://doi.org/10.1145/2713609.2713610

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia*, (46), 1-23.

Mouza, C., Pan, Y. C., Yang, H., & Pollock, L. (2020). A multiyear investigation of student computational thinking concepts, practices, and perspectives in an after-school computing program. *Journal of Educational Computing Research*, *58*(5), 1029-1056. https://doi.org/10.1177/0735633120905605

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, *11*(1), 1-17. https://doi.org/10.1080/20004508.2019.1627844

Numanoğlu, M., & Keser, H. (2017). Robot usage in programmıng teachıng-mbot example. *Bartın University Journal of Faculty of Education*, *6*(2), 497-515. https://doi.org/10.14686/buefad.306198

Ogegbo, A. A., & Ramnarain, U. (2021). A systematic review of computational thinking in science classrooms. *Studies in Science Education*, 1-28. https://doi.org/10.1080/03057267.2021.1963580

Oluk, A., & Korkmaz, Ö. (2018). Information technology teachers' opinions on the use of educational robots. *Pegem Citation Index*, *0*, 215-224. https://doi.org/10.14527/338.

Polat, E., Hopcan, S., Kucuk, S., & Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. *British Journal of Educational Technology*, *52*, 1965-1980. https://doi.org/10.1111/bjet.13092

Papert, S. (1996). An exploration in the space of mathematics educations. *Int. J. Comput. Math. Learn.*, *1*(1),95-123. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.571.4630&rep=rep1&type=pdf

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies*, *22*(2), 421-443. https://doi.org/10.1007/s10639-016-9475-z

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, *169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Rich, K. M., Yadav, A., & Larimore, R. A. (2020). Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. *Education and Information Technologies*, *25*(4), 3161-3188. https://link.springer.com/article/10.1007/s10639-020-10115-5

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*, 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.http://shura.shu.ac.uk/16235/8/RoseExplorationoftheroleofvisualprogramming%28VoR%29.pdf

Rowe, E., Asbell-Clarke, J., Baker, R., Gasca, S., Bardar, E., & Scruggs, R. (2018, April). Labeling Implicit Computational Thinking in Pizza Pass Gameplay. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-6). https://doi.org/10.1145/3170427.3188541

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education Researcher*, *29*(1), 55-66. https://link.springer.com/article/10.1007/s40299-019-00478-w

Scratch (2021). About Scratch. https://scratch.mit.edu/about

Seiter, L., & Foreman, B. (2013, August). 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

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. Retrieved October 9, 2019, from https://eprints.soton.ac.uk/356481/

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

Sırakaya, M., Alsancak Sırakaya, D., & Korkmaz, Ö. (2020). The impact of STEM attitude and thinking style on computational thinking determined via structural equation modeling. *Journal of Science Education and Technology*, *29*, 561-572. https://doi.org/10.1007/s10956-020-09836-6

So, H. J., Jong, M. S. Y., & Liu, C. C. (2020). Computational thinking education in the Asian Pacific region. *The Asia-Pacific Education Researcher*, *29*(1), 1-8. https://doi.org/10.1007/s40299-019-00494-w

Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-Graders' Computational Thinking Skills Through Unplugged Programming Activities: A Study on the Influence of Multiple Factors. *Thinking Skills and Creativity*, 100926. https://doi.org/10.1016/j.tsc.2021.100926

Sun, L., Hu, L., Yang, W., Zhou, D., & Wang, X. (2021). STEM learning attitude predicts computational thinking skills among primary school students. *Journal of Computer Assisted Learning*, *37*(2), 346-358. https://doi.org/10.1111/jcal.12493

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

Tang, K. Y., Chou, T. L., & Tsai, C. C. (2020). A content analysis of computational thinking research: An international publication trends and research typology. *The Asia-Pacific Education Researcher*, *29*(1), 9-19. https://doi.org/10.1007/s40299-019-00442-8

Umutlu, D. (2021). An exploratory study of pre-service teachers' computational thinking and programming skills. *Journal of Research on Technology in Education*, 1-15. https://doi.org/10.1080/15391523.2021.1922105

Üçgül, M. (2013). History and educational potential of Lego Mindstorms NXT. *Mersin University Journal of the Faculty of Education*. *9*(2), 127-137. https://dergipark.org.tr/en/download/article-file/160881

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2021). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education and Development*, *32*(1), 26-48. https://doi.org/10.1080/10409289.2020.1759012

Weinberg, A. E. (2013). Computational thinking: An investigation of the existing scholarship and research (Unpublished doctoral dissertation). Colorado State University, USA. https://www.proquest.com/docview/1413309206?pq-origsite=gscholar&fromopenview=true

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). 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

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://www.microsoft.com/en-us/research/wp-content/uploads/2012/08/Jeannette_Wing.pdf

Wolz, U., Hallberg, C., & Taylor, B. (2011). Scrape: A tool for visualizing the code of scratch programs. In: *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, Dallas, TX, USA

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-319-52691-1_13

Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019, March). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. In *Society for Information Technology & Teacher Education International Conference* (pp. 347-350). Association for the Advancement of Computing in Education (AACE). https://par.nsf.gov/servlets/purl/10095783

Yağcı, M. (2019). A valid and reliable tool for examining computational thinking skills. *Education and Information Technologies*, *24*(1), 929-951. https://doi.org/10.1007/s10639-018-9801-8

Yeni, S. (2018). *How is computational thinking skill evaluated?* Y. Gülbahar (Ed.). From computational thinking to programming in (pp. 359-391). Ankara: Pegem Academy.

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Appendix A

Title	Year			Indexs		Development Approaches												Sub-Skills							Research Groups								Measuring Tools						Results
						Programming				Robotics																													
	2019	2020	2021	SSCI/SSCI-E	ESCI/ERIC	Tool/Environment	Block	Text	Hybrid	Tool/Environment	Block	Text	Hybrid	STEM	Unplugged	Development Courses	Others	Algorithm	Abstraction	Generalization	Decomposition	Debugging	Concepts, Practices, and Perspectives	Others	pre-school	primary-school	secondary-school	High school	undergraduate	Pre-service teachers	In-service teachers	Others	Scales	Test	Interview	Observation	Logs	Others	