JTAM (Jurnal Teori dan Aplikasi Matematika)
http://journal.ummat.ac.id/index.php/jtam
p-ISSN 2597-7512 | e-ISSN 2614-1175
Vol. 8, No. 4, October 2024, pp. 1327-1348

Promoting Computational Thinking through Programming Trends, Tools, and Educational Approaches: a Systematic Review

Edi Irawan[1], Rizky Rosjanuardi[2], Sufyani Prabawanto[2]
[1]Study Program of Mathematics Education, Institut Agama Islam Negeri Ponorogo, Indonesia
[2]Study Program of Mathematics Education, Universitas Pendidikan Indonesia, Indonesia
nawariide@iainponorogo.ac.id

Article History:
Received : 01-10-2024
Revised : 20-10-2024
Accepted : 30-10-2024
Online : 30-10-2024

Keywords:
Computational Thinking;
Educational Approaches;
Programming;
Promoting;
Systematic Review.

ABSTRACT

This systematic research aims to provide a comprehensive overview of the development of analysis related to the use of programming in the development of Computational Thinking (CT), especially in the context of education from primary to tertiary levels. This study analyzed 88 articles from empirical studies related to the use of programming to develop CT sourced from the Scopus database. The analysis process followed the PRISMA 2020 guidelines and consisted of three stages: search, selection, and data analysis. Descriptive and thematic statistical approaches were used for data analysis. Instruments used in the selection of articles included Rayyan for screening based on inclusion criteria, as well as Microsoft Excel for coding and thematic analysis. The results showed that articles related to the use of programming to promote CT have appeared since 2011 but have increased significantly since 2016, with an annual growth rate of 17.6%. Most studies used quantitative approaches, followed by qualitative and mixed methods. Overall, 270 authors from 27 countries contributed to the study, with the United States having the highest number of publications. A total of 33 programming tools were identified, with Scratch being the most widely used tool, followed by Blockly, LEGO, Scratch Jr., Code.org, Python, Alice, App Inventor, Kodu, R, MakeCode, and Arduino. Scratch Jr. is most commonly used at the early childhood education level, while programming languages such as Python, R, and MATLAB are more commonly used in higher education. The implications of these findings suggest that the trend of using programming tools such as Scratch and Blockly has the potential to influence CT teaching strategies in the classroom, as well as the importance of using varied programming tools in efforts to integrate CT into the education curriculum.

https://doi.org/10.31764/jtam.v8i4.26407

A. INTRODUCTION

Computational thinking (CT) is regarded as an essential skill in the current digital era as a problem-solving approach. Enhancing CT skills is crucial for individuals who want to face complex challenges due to the ever-increasing global dependence on technology (Assainova et al., 2023; Jamal et al., 2021; Yadav et al., 2016). CT is now considered an essential competency, on par with basic skills, such as reading, writing, and arithmetic (Wing, 2017). In addition, CT ownership should not be limited to computer scientists, but rather extended to various fields (Grover & Pea, 2013; Irawan & Herman, 2023). As technology continues to evolve, individuals in any field must master CT to remain relevant to their work and daily lives. CT and

programming have strong connections. Programming has played an important role in the development of CT as it involves logical thinking, abstraction, and problem-solving skills (Chan et al., 2023; Hsu et al., 2018; Zhang & Nouri, 2019). As technology advances, various programming tools such as Scratch, Python, and Blockly are increasingly being used to develop CT skills at different educational levels (Tawfik et al., 2024; Vidal-Silva et al., 2024). Programming allows students to learn through exploration and experimentation, thus strengthening their CT skills.

Various efforts have been made to understand how programming can effectively be used to develop CT. One of the most effective approaches is systematic research, which enables in-depth synthesis and analysis of emerging research trends (McKenzie, et al., 2021). This research not only provides insight into the best techniques for developing CT through programming but also identifies gaps in the literature that require further exploration. Although individual studies have provided valuable insights, a systematic review is needed to map existing trends and foresee future research directions. While there has been much research on CT and programming, there has not been a systematic review that specifically focuses on the use of programming to promote CT in various educational contexts. This gap hinders our understanding of the best ways to integrate programming into educational curricula to support CT development (Buitrago Flórez et al., 2017; Hew & Brush, 2007). This problem requires a systematic and comprehensive solution that can answer questions regarding current research trends, the most effective approaches, and the most relevant and frequently used programming tools.

Some systematic reviews have addressed topics such as the integration of CT in mathematics Barcelos et al. (2018); Chan et al. (2023); Irawan et al. (2024); Ye et al. (2023), statistics education Irawan et al. (2024), STEM education Wang et al. (2022), early childhood education Bati, 2022; Su & Yang, 2023; Zeng et al. (2023), primary and secondary education Grover & Pea, 2013; Montiel & Gomez-Zermeño, 2021; Quiroz-Vallejo et al. (2021), higher education Lu et al. (2022); Lyon & J. Magana, (2020), computer science education Lee et al. (2022), learning and teaching CT Hsu et al. (2018), CT learning without the use of computers Kuo & Hsu (2020), mapping programming for a specific CT in high school students Tikva & Tambouris (2021), and CT learning using Scratch (Zhang & Nouri, 2019). However, no systematic review has been conducted on the use of programming to promote computational thinking. This study seeks to fill the existing knowledge gap by comprehensively synthesizing the available literature on programming to promote CT.

This systematic review provides a comprehensive overview of existing research, identifies gaps, and maps the analyses related to programming to promote CT. The results of this study are expected to provide guidance for educators and policymakers in integrating programming to improve CT in the classroom (Chen et al., 2023; Irawan et al., 2024c; Tekdal, 2021). These findings can also serve as a reference for future research focusing on improving CT skills at different levels of education through the use of programming. Based on the research objectives, this study seeks to answer the following five research questions (RQs):

RQ1: What are the research trends related to programming that promote CT?
RQ1 is important because it provides an understanding of the direction of global developments related to programming and CT research, allowing researchers to understand how this topic has evolved over time.

RQ2: What types of approaches are most commonly used in programming research to promote CT?
This question is important because by understanding the dominant approach, we can determine the effectiveness of different methods in supporting CT development through programming.

RQ3: Which journals, institutions, and countries are most productive in producing articles related to programming to promote CT?
The importance of RQ3 is that it provides insight into the key actors in this study and helps identify countries or institutions that are the center of innovation in the field of CT development.

RQ4: What programming has been used in the research to promote CT?
RQ4 is important because an understanding of the most effective tools can help educators choose the appropriate methods for use in their classrooms.

RQ5: What is the distribution of research on using programming to promote CT across different educational levels?
This question provides information on the applicability of programming at different levels, helping educators to design strategies that suit different age groups.

B. METHODS

The method used in this study was a systematic literature review. The review process followed the stages of PRISMA 2020 (Page, McKenzie, et al., 2021; Page, Moher, et al., 2021). Furthermore, the technical description of the method followed by Ye et al. (2023) consists of three processes: search, selection, and data analysis.

1. Search Process

The literature data source was the Scopus database, accessed from Scopus (https://www.scopus.com) on May 31, 2023. The search was conducted using the query string: "(Title ("computational thinking") And Title-Abs-Key ("programming" OR "programming languages" OR "coding") and Title-Abs-Key ("promote" OR "development" OR "learning" OR "teaching" OR "integrating" OR "integrated"))." (2023) This query was designed to generate relevant articles based on key terms that combine "computational thinking" with the use of programming tools, language programming, and learning or teaching processes. Terms such as "promote" and "development" were chosen to cover a range of approaches in research targeting CT development. The subjects of this study were peer-reviewed articles addressing the use of programming in CT development. The collected data included various studies that explored the programming tools used to promote CT in various educational contexts. The search focused on journal articles written in English to ensure broader coverage of quality literature. Journal publications were selected to ensure the credibility and relevance of the review results. This search process yielded 1011 articles related to the exploration of programming tools used to promote CT in various educational contexts.

2. Selection Process

The literature to be analyzed was based on the inclusion and exclusion criteria listed in Table 1. The inclusion and exclusion criteria were determined based on the research objectives.

Table 1. Inclusion and Exclusion Criteria

Aspect	Inclusion Criteria	Exclusion Criteria
Publication	Journal article	Books, Book Chapters, Reviews, Conference Papers, Conference Reviews, Editorials, Erratums, Notes, and Short Survey
Language	Written in English	Not written in English
Study type	Experimental and investigative research	Reviews, overviews, or meta-analyses
Focus study	Promoting computational thinking through programming	Promoting computational thinking through other treatments

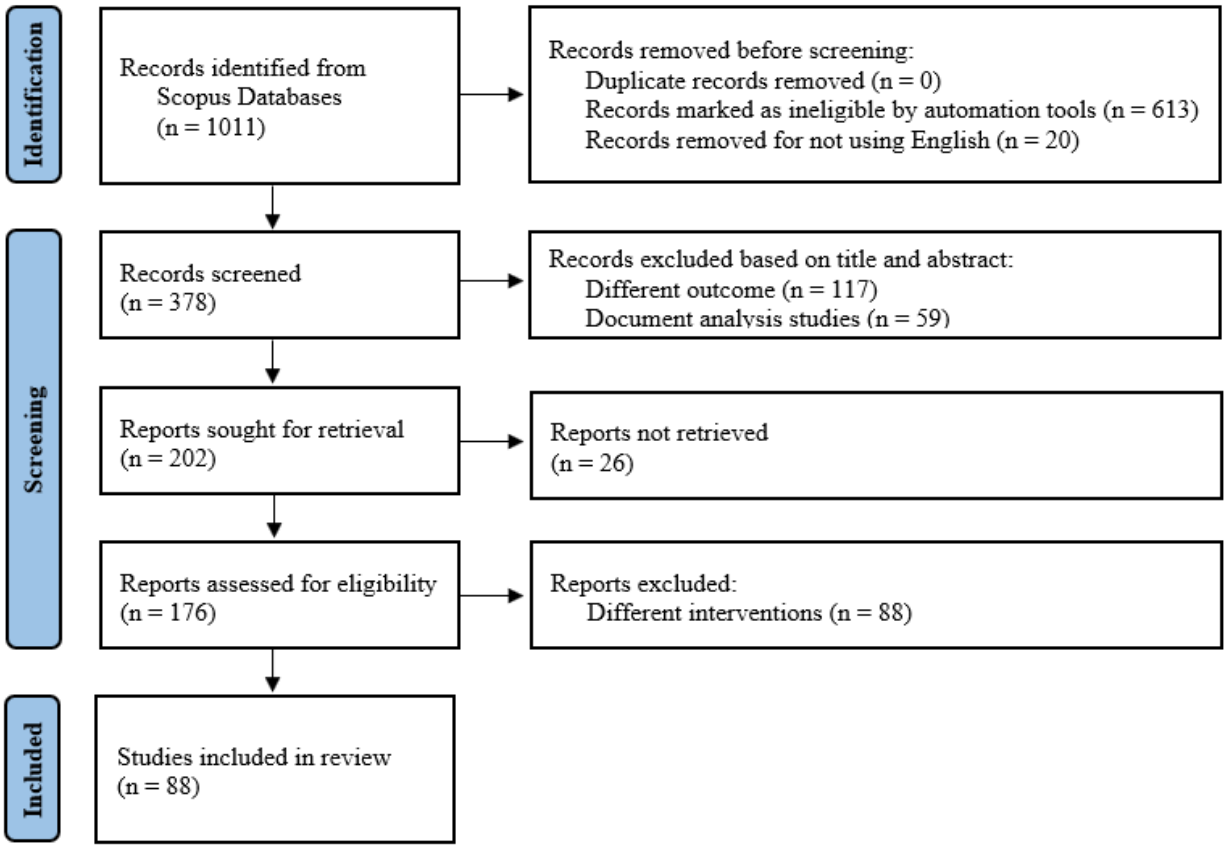

Figure 1. The Process of Systematic Review Using the PRISMA 2020

Technically, the literature selection process was conducted in four stages PRISMA 2020 (Page, et al., 2021), as presented in Figure 1.

a. The initial selection stage was performed during the literature search in the Scopus database. This study focused on articles published in journals and written in English. Therefore, after adding these three criteria, the query string became: "(TITLE ("computational thinking") And Title-Abs-Key ("programming" or "programming languages" or "coding") And Title-Abs-Key ("promote" or "development" or "learning"

or "teaching" or "integrating" or "integrated") And (Limit-To (Doctype, "ar") AND (Limit-To (Language, "English") and (Limit-To (Srctype, "j").” Owing to this selection, 613 literature pieces were not journal articles, and 20 papers needed to be written in English, resulting in 378 articles after the initial screening process.

b. A rapid screening process was conducted on the titles, abstracts, and keywords. The screening process was performed using the Rayyan website (https://www.rayyan.ai/) Ouzzani et al. (2016), which helped categorize the articles according to the inclusion criteria. The selection results yielded 59 systematic review articles and 117 with different outcomes. Therefore, 202 articles were selected for further screening in the next stage.
c. Screening was conducted to determine the availability of the full-text documents. Document availability was searched using the Zotero application, and email was used to obtain closed-access articles. Articles with incomplete manuscripts or relevant information were excluded. There were 26 papers for which the full text still needed to be accepted, resulting in 176 articles that were screened in-depth in the next stage.
d. A comprehensive screening was conducted by reading the complete manuscript of the articles. Full-text screening yielded data from 88 studies that used different interventions. This research focuses on articles discussing the utilization of various computer programs to promote CT.

3. Data Analysis

Data analysis in this study was conducted quantitatively using a descriptive statistical approach to identify the frequency of trends and themes to explore patterns of findings. Each study was mapped, and codes were assigned based on a thorough reading. The articles were identified based on the year of publication, publishing journal, publisher, research approach used, educational level investigated, and programming. The articles were identified and coded based on the results of thorough reading. Microsoft Excel was used in the coding and data analysis processes in this study. Academic levels were categorized into seven groups: kindergarten/preschool, primary school, secondary school, higher education, primary and secondary school, community, and unspecified. The approaches used were classified as qualitative, quantitative, mixed, or unidentified. Other attributes were adjusted based on the findings of each study.

C. RESULT AND DISCUSSION

1. Results

a. Research Trends on Promoting CT Through Programming

The development of research related to CT, particularly the utilization of programming to promote CT, can be observed from the trend in publications. The search results, followed by the process of exclusion and inclusion, yielded data on the number of article publications based on the year, as shown in Figure 2. As shown in Figure 2, articles related to the utilization of programming to promote CT were first discovered in 2011; however, no documents were found in 2011 and 2012. Subsequently, papers on this topic were published from 2014 to 2023. The year 2022 witnessed the highest number

of publications, temporarily reaching a peak of 22. Based on the number of publications from 2011 to 2023, an annual growth rate of 17.6% was obtained with an average document age of 2.93 years. Therefore, it is predicted that the number of publications in 2023 will continue to increase.

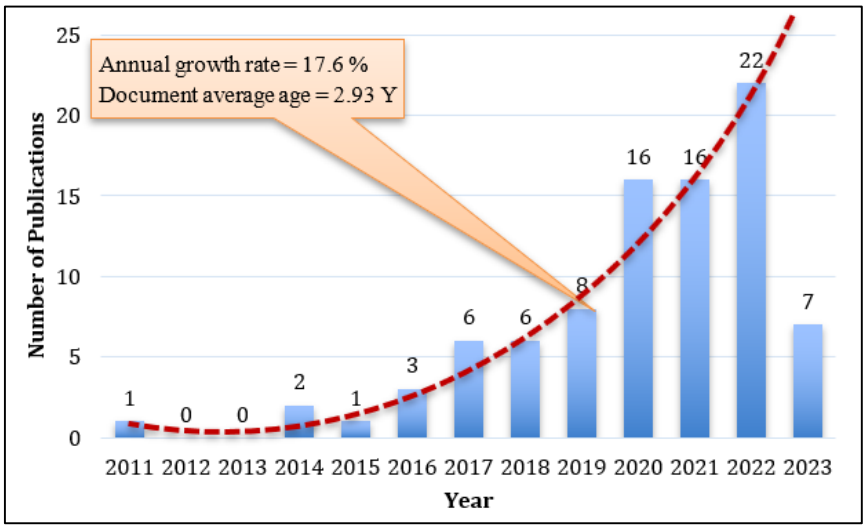

Figure 1. Research Trends on Promoting CT Through Programming

The use of programming to promote CT has increased significantly since 2015. This result aligns with that of Tekdal (2021), where CT has been a rapidly growing field since 2013. Similarly, Chen et al. (2023) mentioned that since 2017, CT has quickly developed as a popular topic in various fields, particularly in education. These research developments are predicted to continue and mature (Chen et al., 2023). The increasing interest in CT-related research contributes to advancing CT research, including integration aspects, programming tools, assessments, and strategies to promote CT.

b. Research Approach on Promoting CT Through Programming

A total of 88 articles analyzed in this study employed different research approaches. The identification results showed three categories of research approaches: qualitative, quantitative, and mixed-methods. The exact numbers and percentages for each approach are shown in Figure 3.

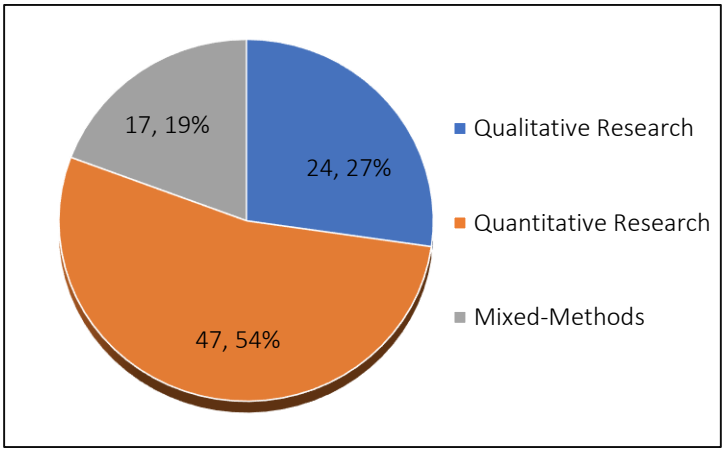

Figure 3. Research Approach to Promote CT Through Programming

Figure 3 shows that research related to the utilization of programming to promote CT is predominantly conducted using a quantitative approach. Furthermore, 47 articles, equivalent to 54% of all studies, were included using a quantitative approach. Of the remaining articles, 24 (27%) employed a qualitative approach, whereas 17 (19%) used a mixed method. The quantitative research conducted was mainly based on experimental designs, comparing the effects of programming utilization on students' CT skills, both in comparison with other programming types and non-computer-based learning methods. The use of different research approaches indicates the diversity in CT research. CT is performed from multiple perspectives. Several experimental studies using quantitative methods have been conducted to ensure the effectiveness of different breakthroughs in the development of CT. Furthermore, CT is a new and evolving research field (Tekdal, 2021).

c. Top 10 Most Productive Journals, Countries, and Institutions

A total of 44 journals published articles related to the use of programming to promote CT. Table 2 presents a list of the top 10 most productive journals, along with their publishers, total publication (TP) of articles on the utilization of programming to promote CT, and total citations (TC) received by those articles. Among these journals, four had the highest number of publications. These include ACM Transactions on Computing Education, Journal of Educational Computing Research, Informatics in Education, and Frontiers in Psychology published by the Association for Computing Machinery, SAGE Publications Inc., Institute of Mathematics and Informatics, and Frontiers Media S.A. Furthermore, ACM Transactions on Computing Education was the most cited journal among the listed journals. It received a remarkable 218 citations from other articles, indicating its significant role as a source of inspiration and reference.

Table 2. Top 10 Most Productive Journals

No	Journal	Publisher	TP* (%)	TC**
1	ACM Transactions on Computing Education	Association for Computing Machinery	5 (3,36%)	218
2	Journal of Educational Computing Research	SAGE Publications Inc.	5 (3,36%)	103
3	Informatics in Education	Institute of Mathematics and Informatics	5 (3,36%)	80
4	Frontiers in Psychology	Frontiers Media S.A.	5 (3,36%)	7
5	Education and Information Technologies	Springer	4 (2,68%)	15
6	Education Sciences	MDPI	4 (2,68%)	6
7	International Journal of Child-Computer Interaction	Elsevier B.V.	3 (2,01%)	169
8	Computers in Human Behavior	Elsevier Ltd	3 (2,01%)	111
9	Interactive Learning Environments	Routledge	3 (2,01%)	102
10	Technology, Knowledge, and Learning	Springer Science and Business Media B.V.	3 (2,01%)	64

* TP: Total Publication

** TC: Total citation

A total of 27 countries were identified based on the countries of the corresponding authors. The list of the top ten countries and the number of publications are presented in Figure 4. The United States is the most productive country, followed by China, Taiwan, Spain, Greece, Peru, Italy, Cyprus, and South Korea.

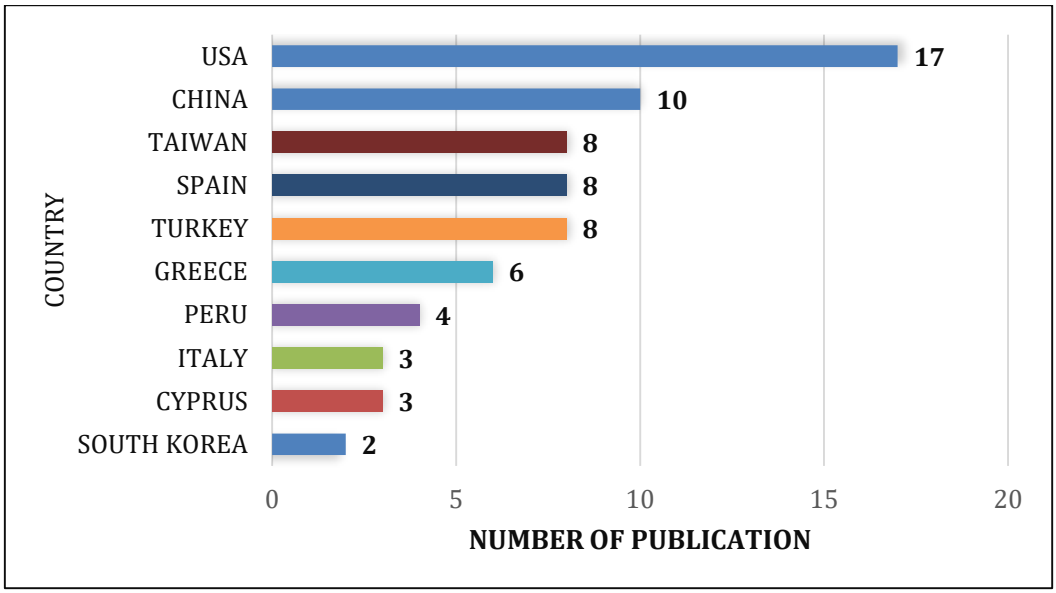

Figure 4. Top 10 Most Productive Countries by Correspondence Author Affiliation

Published articles related to the use of programming to promote CT involved 270 authors from 110 institutions. The list in Table 3 presents the top 10 institutions, their respective countries, the TP of articles related to the utilization of programming to promote CT, and the TC received. The ranking of institutions in Table 3 was determined based on TP and TC.

Table 3. Top 10 Most Productive Institutions

No	Institution	Country	TP (%)	TC
1	Tufts University	United States	2	568
2	National and Kapodistrian University of Athens	Greece	2	23
3	Shaanxi Normal University	China	2	23
4	Spanish National University of Distance Education (UNED)	Spain	2	21
5	Zhejiang University of Technology	China	2	13
6	National Taichung University of Science and Technology	Taiwan	2	10
7	Jeju National University	South Korea	2	8
8	University of Macedonia	Greece	2	8
9	Universidad Nacional de San Agustin de Arequipa	Peru	2	3
10	The Education University of Hong Kong	China	2	0

Based on Table 3, among the 110 institutions that have generated articles on the utilization of programming to promote CT, ten stand out as the most productive and highly cited. Tufts University has the highest number of referenced articles. However, in

terms of publication counts, no institution dominates. Therefore, research focused on programming to promote CT has attracted the interest of numerous colleges in different countries. Based on a comprehensive tally of authors, totaling 270 individuals from 27 distinct countries, programming to promote CT has emerged as a captivating research theme of global prominence. Scholarly journals dominate article publications in educational and computer science. Moreover, the United States is the most productive nation in publishing articles on programming to promote CT. This outcome resonates with other findings that position the United States at the forefront of CT research (Tekdal, 2021).

d. The Variety of Programming to Promote CT

The identification results indicate that there were a total of 33 programming tools used to promote and develop CT. The list of programming names, creators/developers, number of articles (%), and corresponding articles are presented in Table 4.

Table 4. List of Programming Used to Promote CT

Programming	Creator/Developer	Freq. (%)	Source
Scratch	MIT Media Lab	37 (32,17%)	(Bell & Bell, 2018; Eloy et al., 2022; Gabriele et al., 2019; Hou et al., 2020; Hsu et al., 2022; Huang et al., 2023; Jiang et al., 2021; Jiang & Li, 2021; Ma et al., 2021; Maraza-Quispe et al., 2021), (Allsop, 2019; Basogain et al., 2018; Pérez-Marín et al., 2020; Rodríguez-Martínez et al., 2020; Romero et al., 2017; Wei et al., 2021; Wong & Cheung, 2020)
Blockly	Google and MIT	8 (6,96%)	(Díaz-Lauzurica & Moreno-Salinas, 2019; Karakasis & Xinogalos, 2020; Luo et al., 2020; Tikva & Tambouris, 2022; Tran, 2019)
LEGO	MIT Media Lab	8 (6,96%)	(Angeli, 2022; Ardito et al., 2020; Chalmers, 2018; Chiazzese et al., 2019; Weng et al., 2022)
Scratch Jr	MIT Media Lab	8 (6,96%)	(Chou, 2020; Kourti et al., 2023; Kyza et al., 2022; Papadakis et al., 2016; Pugnali et al., 2017; Rose et al., 2017; Silva et al., 2021; Yang et al., 2023)
Code.org	Hadi and Ali Partovi	6 (5,22%)	(Sun et al., 2022)
Python	Guido van Rossum	6 (5,22%)	(Bai et al., 2021; Ezeamuzie et al., 2022; Laura-Ochoa et al., 2022; Laura-Ochoa & Bedregal-Alpaca, 2022; Lin et al., 2021; Song et al., 2021)
Alice	Carnegie Mellon University	5 (4,35%)	(Grover et al., 2017)
App Inventor	Google and MIT	4 (3,48%)	(Kim & Kim, 2016; Shih et al., 2015; P.-J. Wu et al., 2021)

Programming	Creator/Developer	Freq. (%)	Source
Kodu	Microsoft	3 (2,61%)	(Chiazzese et al., 2018)
R	Ross Ihaka, Robert Gentleman	2 (1,74%)	(Benakli et al., 2017; Wiedemann et al., 2020)
MakeCode	Microsoft	2 (1,74%)	(Andersen, 2022; Kastner-Hauler et al., 2022)
Arduino	arduino.cc	2 (1,74%)	(Karaahmetoğlu & Korkmaz, 2019)
Bomberbot	Cristian Bello	1 (0,87%)	(Fanchamps et al., 2023)
C/C++	ISO/IEC JTC 1	1 (0,87%)	(Markandan et al., 2022)
ChoiCo	Chronis K. and Marianthi G.	1 (0,87%)	(Kynigos & Grizioti, 2020)
CodeMonkey	CodeMonkey Team	1 (0,87%)	(S.-Y. Wu & Su, 2021)
Hopscotch	Hopscotch Technologies	1 (0,87%)	(Zha et al., 2020)
Kodetu	Kodetu et al.	1 (0,87%)	(Eguiluz et al., 2020)
Logo	Bolt Beranek and Newman Inc	1 (0,87%)	(Kynigos & Grizioti, 2018)
Matlab	MathWorks	1 (0,87%)	(Yuen & Robbins, 2015)
mBlock	Makeblock	1 (0,87%)	(Paucar-Curasma et al., 2023)
Minecraft	Mojang Studios	1 (0,87%)	(Kutay & Oner, 2022)
ModKit	Ed Baafi	1 (0,87%)	(Richard & Giri, 2019)
NEPO	Fraunhofer IAIS	1 (0,87%)	(Weber et al., 2022)
Parsons Puzzles	Dale Parsons and Patricia Haden	1 (0,87%)	(Bender et al., 2023)
Pencil Code	David Bau	1 (0,87%)	(Deng et al., 2020)
RobotC	Robomatter Inc	1 (0,87%)	(Witherspoon et al., 2017)
ScratchThAI	Kantinee Katchapakirin et al.	1 (0,87%)	(Katchapakirin et al., 2022)
Sonic Pi	Sam Aaron and Raspberry Pi	1 (0,87%)	(Petrie, 2022)
Sprego	Maria C. and Piroska B.	1 (0,87%)	(Csernoch et al., 2021)
TangiBlek	Marina Umaschi Bers	1 (0,87%)	(Bers et al., 2014)
Thunkable	Arun Saigal and WeiHua Li	1 (0,87%)	(Amnouychokanant et al., 2021)
Visual Basic	Microsoft	1 (0,87%)	(Deng et al., 2020)

As shown in Table 4, many programming-based tools have been used to develop and promote CT. Scratch is the most popular programming tool and is widely used in research. A total of 37 articles, equivalent to 32.17% of the analyzed articles, utilized this visual block-based programming tool developed by the MIT Media Lab. This was followed by Blockly, LEGO, and Scratch Jr, which are also visual block-based programming tools, with eight articles each. Python is the most commonly used programming language, followed by R, C/C++, and Visual Basic. Based on the publication year of the articles, data on the programming tools used were obtained, as presented in Table 5.

Table 5. Programming Utilized to Promote CT by Year

Year	Programming (Frequency)
2011	Scratch (1)
2014	Matlab (1), and TangiBlek (1)
2015	App Inventor (1)
2016	Scratch Jr (1), App Inventor (1), and LEGO (1)
2017	Scratch Jr (2), Scratch (1), R (1), Alice (1), and RobotC (1)
2018	Scratch (3), LEGO (1), Logo (1), Alice (1), and Kodu (1)
2019	Scratch (4), Blockly (2), LEGO (2), Code.org (1), Alice (1), App Inventor (1), and Arduino (1)
2020	Scratch (7), Blockly (3), Scratch Jr (1), R (1) ChoiCo (1), Hopscotch (1), Kodetu (1), LEGO (1), Alice (1), and Pencil Code (1)
2021	Scratch (11), Python (3), App Inventor (1), Scratch Jr (1), Sprego (1), CodeMonkey (1), Arduino (1), Blockly (1), Kodu (1), Code.org (1), and Thunkable (1)
2022	Scratch (8), Code.org (4), Python (3), Blockly (2), LEGO (2), MakeCode (2), C/C++ (1), NEPO (1), Scratch Jr (1), Alice (1), Kodu (1), ScratchThAI (1), and Sonic Pi (1)
2023	Scratch (3), Scratch Jr (2), Bomberbot (1), LEGO (1), Arduino (1), Parsons Puzzles (1), and mBlock (1)

Table 5 shows that Scratch is a popular and widely used programming tool. Meanwhile, LEGO, Scratch Jr., and Blockly were frequently employed in research focused on promoting CT. Table 5 also indicates that with each passing year, more programming platforms were utilized to promote and acquire CT skills in students. New platforms have emerged each year, encompassing visual block-based programming, coding-based programming languages, or a combination of both.

e. Educational Levels and Programming Types Used

Data on educational levels based on the target subjects are presented in Figure 5. The academic levels are categorized into kindergarten/preschool, primary school, secondary school, higher education, primary and secondary school, and community, and they are not specified.

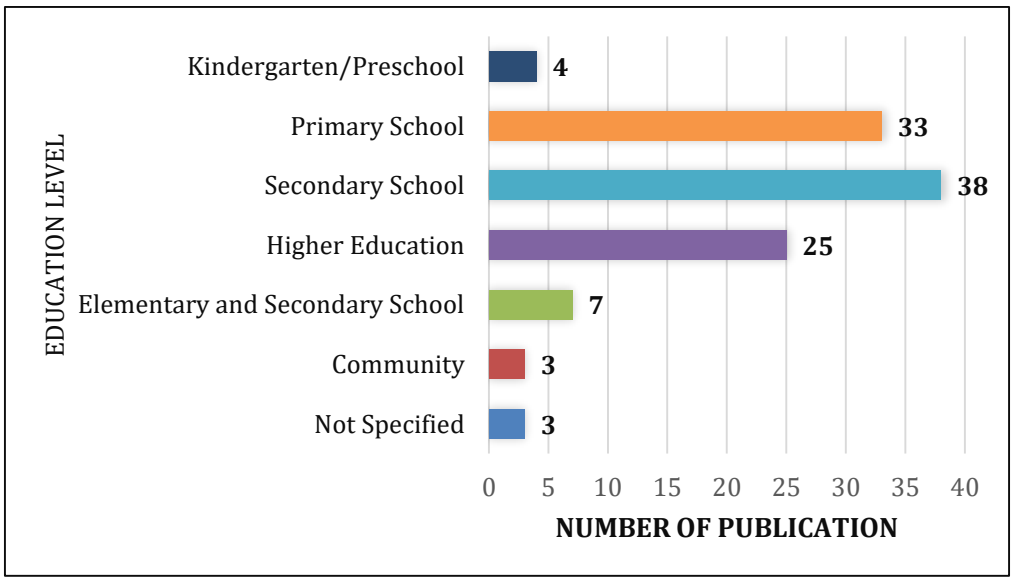

Figure 5. Target Educational Levels of the Selected Articles

Figure 5 shows that most of the research was conducted at the secondary school level, with 38 studies or 34% of the total. This was followed by research conducted at the primary school level with 33 articles (29%), higher education with 25 papers (22%), primary and secondary school with seven papers (6%), and a community with three papers (3%). The programming tools used in each educational level or group are listed in Table 6.

Table 6. Educational levels and programming used to Promote CT

Level Education	Programming (Frequency)
Kindergarten/ Preschool	Scratch Jr (3) and TangiBlek (1)
Primary School	Scratch (12), Scratch Jr (5), LEGO (3), Blockly (3), Code.org (2), Kodu (2), Alice (1), App Inventor (1), MakeCode (1), Bomberbot (1), ScratchThAI (1), and CodeMonkey (1),
Secondary School	Scratch (12), Blockly (3), Code.org (3), Alice (3), LEGO (2), Arduino (2), Python (1), Kodu (1), R (1), Logo (1), C/C++ (1), Minecraft (1), Sonic Pi (1), ChoiCo (1), Parsons Puzzles (1), Pencil Code (1), Visual Basic (1), ModKit (1), and RobotC (1)
Higher Education	Scratch (6), Python (5), LEGO (3), App Inventor (2), R (1), Code.org (1), Arduino (1), Matlab (1), Hopscotch (1), mBlock (1), Thunkable (1), Sprego (1), and NEPO (1)
Primary and Secondary School	Scratch (2), Blockly (2), Alice (1), MakeCode (1), and Kodetu (1)
Community	Scratch (3)

Table 6 shows that two programming tools were used at the kindergarten/preschool level, namely Scratch Jr and TangiBlek. For the primary school level, there were 12 programming tools, including Scratch, Scratch Jr, LEGO, Blockly, Code.org, Alice, App Inventor, Kodu, MakeCode, Bomberbot, ScratchThAI, and CodeMonkey. For the secondary school level, there were 19 programming tools employed, namely Scratch, LEGO, Blockly, Python, Code.org, Alice, Kodu, R, Arduino, Logo, C/C++, Minecraft, Sonic Pi, ChoiCo, Parsons Puzzles, Pencil Code, Visual Basic, ModKit, and RobotC. Thirteen programming tools were utilized in the higher education group, there were 13 programming tools utilized, including Scratch, LEGO, Python, Code.org, App Inventor, R, Arduino, MATLAB, Hopscotch, mBlock, Thunkable, Sprego, and NEPO. In the community group, only one program, Scratch, was used. Scratch Jr. is the most widely used programming tool at the kindergarten/preschool level. Scratch is the most commonly used tool at the primary, secondary, higher education, and community levels. Various programming tools predominantly consist of block-based visual programming. Python is the most widely used programming language in higher education, followed by R, C/C++, and Visual Basic.

2. Discussions

The utilization of programming to promote CT has experienced a significant increase since 2015. This result aligns with Tekdal's research, where CT has been a rapidly growing field since 2013 (Tekdal, 2021). In a similar manner, Chen et al. (2023) mentioned that since 2017, CT has quickly developed as a popular topic in various fields, particularly in education. The development of this research is expected to continue and mature further (Chen et al., 2023). The increasing interest in CT-related research contributes to advancing CT research, including integration aspects, programming tools, assessment, and strategies to promote CT. The use of different research approaches indicates the diversity of CT research. CT development is carried out from multiple aspects and perspectives. Several experimental studies using quantitative methods have been conducted to ensure the effectiveness of different breakthroughs in the development of CT. Furthermore, CT is a new research field still evolving (Tekdal, 2021).

Based on the comprehensive tally of authors, totaling 270 individuals from 27 distinct countries, the utilization of programming to promote CT has emerged as a captivating research theme of global prominence. Articles in educational technology and computer science are largely dominated by scholarly journals. Moreover, the United States is the most productive nation in terms of publishing articles on the utilization of programming to promote CT. This outcome resonates with other findings that position the United States at the forefront of CT research (Tekdal, 2021). Through the identification of all articles, 33 programming tools were identified to promote and develop CT. Scratch was the most widely used programming tool, followed by Blockly, LEGO, Scratch Jr, Code.org, Python, Alice, App Inventor, Kodu, R, MakeCode, Arduino, Bomberbot, C/C++, ChoiCo, CodeMonkey, Hopscotch, Kodetu, Logo, Matlab, mBlock, Minecraft, ModKit, NEPO, Parsons Puzzles, Pencil Code, RobotC, ScratchThAI, Sonic Pi, Sprego, TangiBlek, Thunkable, and Visual Basic. These findings contribute to previous research conducted by Hsu et al. (2018) who identified 14 teaching tools, and Chan et al. (2023) who placed 21 programming teaching tools for CT development.

Research on programming tools has been predominantly conducted in secondary schools, followed by primary schools, higher education, and the community. This finding supported and strengthened the results of other research, such as Tang et al. (2020), where CT research in secondary schools was more dominant. Scratch, a block-based visual programming language, is relatively easier to use, making it widely adopted in primary and secondary schools, higher education, and the community. Different results indicate that the use of Scratch develops students' CT skills (Chiazzese et al., 2018; Huang et al., 2023; Irawan et al., 2023; Ma et al., 2021; Maraza-Quispe et al., 2021; Oluk et al., 2018; Pou et al., 2022; Xing, 2021). However, these results differed from other findings, where Alice had a more positive effect on students' CT skills than Scratch (Yildiz Durak, 2020). Scratch does not significantly affect problem-solving or algorithmic thinking skills (Jiang & Li, 2021). Meanwhile, Scratch Jr. is mainly used in kindergarten/preschool settings (Chou, 2020; Kyza et al., 2022; Rose et al., 2017). Coding-based programming languages are commonly used in higher education and secondary schools. Research has shown that its use in higher education enhances students' CT skills (Bai et al., 2021; Laura-Ochoa & Bedregal-Alpaca, 2022; Song et al., 2021). The same applies to R, which integrates CT into calculus, probability, and data analysis (Benakli et al., 2017). Other research

has indicated that R instilled CT while improving students' mathematical abilities (Wiedemann et al., 2020).

3. Implication and Limitation

a. Implication for Practice, Policy, and Future Research

This study has several implications. *First*, it has practical implications. These findings indicate that the utilization of programming to promote CT has increased significantly since 2015. Therefore, educators should consider utilizing various programming tools, such as Scratch, Blockly, Code.org, Python, and R, in curricula designed to improve students' CT skills. Furthermore, educators can leverage various programming tools to design engaging activities and projects that promote CT across multiple subjects including mathematics, science, language, and other social topics.

Second, policy implications. his research highlights the global interest in and dissemination of research related to the utilization of programming to promote CT. Policymakers must recognize the significance of CT in education and consider its integration into curricula at different educational levels. In addition, policies can be supported through training and professional development for educators to effectively integrate CT into their teaching practices. Policymakers must also encourage innovation to develop programming designed explicitly to promote CT.

Third, implications for future research. The identified research trends demonstrate the diverse approaches and perspectives for studying CT. Future research should explore instructional strategies and designs for the systematic and practical development of CT. Meanwhile, comparative research should be conducted to investigate the impact of different programming languages, such as Scratch, Python, and R, on promoting CT at different educational levels. Longitudinal research is required to examine the long-term effects of CT integration into the curriculum and its influence on student's academic achievement and career paths. Further research is required to explore innovative methods to accurately evaluate and measure CT skills.

b. Limitations

This study has several limitations. First, there is a limitation regarding the literature sources, and this research is limited to literature sourced from the Scopus database only. This research does not include other studies indexed in the Web of Science or other indexing institutions. Second, there are limitations to the scope of the data analysis. The data analysis was conducted based on the formulated research questions. Therefore, other aspects, such as the types of CT skills, duration of the research, instruments used for data collection, and approaches employed, need to be explored in depth. Third, the analysis was limited to a systematic review and did not perform a meta-analysis because the number of eligible articles was limited.

D. CONCLUSION AND SUGGESTIONS

Several conclusions were drawn regarding the utilization of programming to promote CT. *First*, there has been a trend of increasing publications related to the utilization of programming to promote CT since 2011, with an annual growth rate of 17.6%. This increase demonstrates a growing interest in the use of programming tools to support the development of CT skills in

various educational contexts. *Second*, the most commonly used approach is quantitative, followed by qualitative and mixed methods. *Third*, the publication of articles related to the utilization of programming to promote CT involved 270 authors from 27 countries, with the United States being the most productive. *Fourth*, 33 programming tools are identified. Scratch was the most widely used programming tool, followed by Blockly, LEGO, Scratch Jr, Code.org, Python, Alice, App Inventor, Kodu, R, MakeCode, and Arduino. *Fifth*, this study indicated a distribution of research based on educational levels. Scratch is most commonly used in primary, secondary, higher education, and community settings. Meanwhile, Scratch Jr was predominantly used in kindergarten/preschool education, while programming languages such as Python, R, and MATLAB were more commonly utilized in higher education. This research provides valuable insights into understanding trends, approaches, types of programming tools, and the distribution of research related to the utilization of programming to promote CT. The significance of these findings in the context of education is enormous, especially in efforts to develop students' critical thinking, problem-solving, and creative skills through programming. The results of this study have practical implications that can encourage curriculum development that is more adaptive to the use of programming tools as well as help educators design teaching strategies that are more innovative and relevant to the needs of students in the digital era.

Recommendations for future research include further exploration of the effectiveness of different programming languages in promoting CT at different educational levels. Comparative research is required to compare the impact of different programming tools, such as Scratch, Python, and R, in different teaching contexts. In addition, longitudinal research should be conducted to examine the long-term effects of integrating programming into the curriculum on students' academic achievement and career development. Future research could also explore innovative methods to measure CT skills more accurately and develop programming tools specifically designed to support the development of CT skills in students from different educational backgrounds.

ACKNOWLEDGEMENT

This research was supported by Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia on the Doctoral Dissertation Postgraduate (PDD) research scheme for the 2024 fiscal year (Grant No. 1218 /UN40.LP/PT.01.03/2024).

REFERENCES

Allsop, Y. (2019). Assessing Computational Thinking Process Using a Multiple Evaluation Approach. *International Journal of Child-Computer Interaction*, *19*(1), 30–55. https://doi.org/10.1016/j.ijcci.2018.10.004

Amnouychokanant, V., Boonlue, S., Chuathong, S., & Thamwipat, K. (2021). Online Learning Using Block-based Programming to Foster Computational Thinking Abilities During the COVID-19 Pandemic. *International Journal of Emerging Technologies in Learning*, *16*(13), 227–247. https://doi.org/10.3991/ijet.v16i13.22591

Andersen, R. (2022). Block-Based Programming and Computational Thinking in a Collaborative Setting: A Case Study of Integrating Programming into a Maths Subject. *Acta Didactica Norden*, *16*(4), 1–22. https://doi.org/10.5617/adno.9169

Angeli, C. (2022). The Effects of Scaffolded Programming Scripts on Pre-Service Teachers' Computational Thinking: Developing Algorithmic Thinking Through Programming Robots. *International Journal of Child-Computer Interaction*, *31*(C), 1–20. https://doi.org/10.1016/j.ijcci.2021.100329

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/10.1007/s11528-019-00461-8

Assainova, A. Z., Abykenova, D. B., Aubakirova, Z. T., Mukhamediyeva, K. M., & Kozhageldinova, K. A. (2023). Web Technologies in the Development of Computational Thinking of Students with Mental Disabilities. *International Journal of Emerging Technologies in Learning (iJET)*, *18*(11), 74–92. https://doi.org/10.3991/ijet.v18i11.38653

Bai, H., Wang, X., & Zhao, L. (2021). Effects of the Problem-Oriented Learning Model on Middle School Students' Computational Thinking Skills in a Python Course. *Frontiers in Psychology*, *12*(1), 1–14. https://doi.org/10.3389/fpsyg.2021.771221

Barcelos, T. S., Munoz, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *Journal of Universal Computer Science*, *24*(7), 815–845. https://doi.org/10.3217/jucs-024-07-0815

Basogain, X., Olabe, M. Á., Olabe, J. C., & Rico, M. J. (2018). Computational Thinking in Pre-University Blended Learning Classrooms. *Computers in Human Behavior*, *80*(3), 412–419. https://doi.org/10.1016/j.chb.2017.04.058

Bati, K. (2022). A Systematic Literature Review Regarding Computational Thinking and Programming in Early Childhood Education. *Education and Information Technologies*, *27*(2), 2059–2082. Scopus. https://doi.org/10.1007/s10639-021-10700-2

Bell, J., & Bell, T. (2018). Integrating Computational Thinking with a Music Education Context. *Informatics in Education*, *17*(2), 151–166. https://doi.org/10.15388/infedu.2018.09

Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2017). Introducing Computational Thinking Through Hands-on Projects Using R with Applications to Calculus, Probability and Data Analysis. *International Journal of Mathematical Education in Science and Technology*, *48*(3), 393–427. https://doi.org/10.1080/0020739X.2016.1254296

Bender, J., Zhao, B., Dziena, A., & Kaiser, G. (2023). Integrating Parsons Puzzles Within Scratch Enables Efficient Computational Thinking Learning. *Research and Practice in Technology Enhanced Learning*, *18*(22), 1–25. https://doi.org/10.58459/rptel.2023.18022

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational Thinking and Tinkering: Exploration of an Early Childhood Robotics Curriculum. *Computers & Education*, *72*(1), 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

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

Chalmers, C. (2018). Robotics and Computational Thinking in Primary School. *International Journal of Child-Computer Interaction*, *17*(1), 93–100. https://doi.org/10.1016/j.ijcci.2018.06.005

Chan, S.-W., Looi, C.-K., Ho, W. K., & Kim, M. S. (2023). Tools and Approaches for Integrating Computational Thinking and Mathematics: A Scoping Review of Current Empirical Studies. *Journal of Educational Computing Research*, *60*(8), 1–45. https://doi.org/10.1177/07356331221098793

Chen, H. E., Sun, D., Hsu, T.-C., Yang, Y., & Sun, J. (2023). Visualising Trends in Computational Thinking Research from 2012 to 2021: A Bibliometric Analysis. *Thinking Skills and Creativity*, *47*(3), 1–18. https://doi.org/10.1016/j.tsc.2022.101224

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019). Educational Robotics in Primary School: Measuring the Development of Computational Thinking Skills with the Bebras Tasks. *Informatics*, *6*(4), 1–12. https://doi.org/10.3390/informatics6040043

Chiazzese, G., Fulantelli, G., Pipitone, V., & Taibi, D. (2018). Engaging Primary School Children in Computational Thinking: Designing and Developing Videogames. *Education in the Knowledge Society*, *19*(2), 63–81. https://doi.org/10.14201/EKS20181926381

Chou, P.-N. (2020). Using ScratchJr to Foster Young Children's Computational Thinking Competence: A Case Study in a Third-Grade Computer Class. *Journal of Educational Computing Research*, *58*(3), 570–595. https://doi.org/10.1177/0735633119872908

Csernoch, M., Biro, P., & Math, J. (2021). Developing Computational Thinking Skills With Algorithm-Driven Spreadsheeting. *IEEE Access*, *9*(1), 153943–153959. https://doi.org/10.1109/ACCESS.2021.3126757

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. https://doi.org/10.1002/cae.22177

Díaz-Lauzurica, B., & Moreno-Salinas, D. (2019). Computational Thinking and Robotics: A Teaching Experience in Compulsory Secondary Education with Students with High Degree of Apathy and Demotivation. *Sustainability (Switzerland)*, *11*(18), 1–21. https://doi.org/10.3390/su11185109

Eguiluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodriguez, C. (2020). Exploring the Progression of Early Programmers in a Set of Computational Thinking Challenges via Clickstream Analysis. *IEEE Transactions on Emerging Topics in Computing*, *8*(1), 256–261. https://doi.org/10.1109/TETC.2017.2768550

Eloy, A., Achutti, C. F., Fernandez, C., & De Deus Lopes, R. (2022). A Data-Driven Approach to Assess Computational Thinking Concepts Based on Learners' Artifacts. *Informatics in Education*, *21*(1), 33–54. https://doi.org/10.15388/infedu.2022.02

Ezeamuzie, N. O., Leung, J. S. C., Garcia, R. C. C., & Ting, F. S. T. (2022). Discovering Computational Thinking in Everyday Problem Solving: A Multiple Case Study of Route Planning. *Journal of Computer Assisted Learning*, *38*(6), 1779–1796. https://doi.org/10.1111/jcal.12720

Fanchamps, N., Slangen, L., Specht, M., & Hennissen, P. (2023). Effect of SRA-Programming on Computational Thinking Through Different Output Modalities. *Journal of Computers in Education*, *10*(2), 433–462. https://doi.org/10.1007/s40692-022-00236-w

Gabriele, L., Bertacchini, F., Tavernise, A., Vaca-Cárdenas, L., Pantano, P., & Bilotta, E. (2019). Lesson Planning by Computational Thinking Skills in Italian Pre-Service Teachers. *Informatics in Education*, *18*(1), 69–104. https://doi.org/10.15388/infedu.2019.04

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A Framework for Using Hypothesis-Driven Approaches to Support Data-Driven Learning Analytics in Measuring Computational Thinking in Block-Based Programming Environments. *ACM Transactions on Computing Education*, *17*(3), 1–25. https://doi.org/10.1145/3105910

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

Hew, K. F., & Brush, T. (2007). Integrating Technology into K-12 Teaching and Learning: Current Knowledge Gaps and Recommendations for Future Research. *Educational Technology Research and Development*, *55*(3), 223–252. https://doi.org/10.1007/s11423-006-9022-5

Hou, H.-Y., Agrawal, S., & Lee, C.-F. (2020). Computational Thinking Training with Technology for Non-Information Undergraduates. *Thinking Skills and Creativity*, *38*(100720), 1–12. https://doi.org/10.1016/j.tsc.2020.100720

Hsu, T.-C., Chang, C., Wu, L.-K., & Looi, C.-K. (2022). Effects of a Pair Programming Educational Robot-Based Approach on Students' Interdisciplinary Learning of Computational Thinking and Language Learning. *Frontiers in Psychology*, *13*(888215), 1–15. https://doi.org/10.3389/fpsyg.2022.888215

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*(1), 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Huang, S.-Y., Tarng, W., & Ou, K.-L. (2023). Effectiveness of AR Board Game on Computational Thinking and Programming Skills for Elementary School Students. *Systems*, *11*(25), 1–31. https://doi.org/10.3390/systems11010025

Irawan, E., & Herman, T. (2023). Trends in Research on Interconnection of Mathematics and Computational Thinking. *AIP Conference Proceedings*, *2805*, 1–9. https://doi.org/10.1063/5.0148018

Irawan, E., Kusumah, Y. S., & Saputri, V. (2023). Pengembangan Multimedia Interaktif Menggunakan Scratch: Solusi Pembelajaran di Era Society 5.0. *AKSIOMA: Jurnal Program Studi Pendidikan Matematika*, *12*(1), 36–50. https://doi.org/10.24127/ajpm.v12i1.6226

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024a). Advancing Computational Thinking in Mathematics Education: A Systematic Review of Indonesian Research Landscape. *JTAM (Jurnal Teori Dan Aplikasi Matematika)*, *8*(1), 176–194. https://doi.org/10.31764/jtam.v8i1.17516

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024b). Harnessing the Power of Technology in Statistics Education: A Comprehensive Bibliometric Study. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *58*(2), 108–126. https://doi.org/10.37934/araset.58.2.108126

Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024c). Research Trends of Computational Thinking in Mathematics Learning: A Bibliometric Analysis from 2009 to 2023. *Eurasia Journal of Mathematics, Science and Technology Education*, *20*(3), 1–16. https://doi.org/10.29333/ejmste/14343

Jamal, N. N., Jawawi, D. N. A., Hassan, R., & Mamat, R. (2021). Conceptual Model of Learning Computational Thinking Through Educational Robotic. *International Journal of Emerging Technologies in Learning (iJET)*, *16*(15), 91–106. https://doi.org/10.3991/ijet.v16i15.24257

Jiang, B., & Li, Z. (2021). Effect of Scratch on Computational Thinking Skills of Chinese Primary School Students. *Journal of Computers in Education*, *8*(4), 505–525. https://doi.org/10.1007/s40692-021-00190-z

Jiang, B., Zhao, W., Gu, X., & Yin, C. (2021). Understanding the Relationship Between Computational Thinking and Computational Participation: A Case Study from Scratch Online Community. *Educational Technology Research and Development*, *69*(5), 2399–2421. https://doi.org/10.1007/s11423-021-10021-8

Karaahmetoğlu, K., & Korkmaz, Ö. (2019). The Effect of Project-Based Arduino Educational Robot Applications on Students' Computational Thinking Skills and Their Perception of Basic Stem Skill Levels. *Participatory Educational Research*, *6*(2), 1–14. https://doi.org/10.17275/per.19.8.6.2

Karakasis, C., & Xinogalos, S. (2020). BlocklyScript: Design and Pilot Evaluation of an RPG Platform Game for Cultivating Computational Thinking Skills to Young Students. *Informatics in Education*, *19*(4), 641–668. https://doi.org/10.15388/INFEDU.2020.28

Kastner-Hauler, O., Tengler, K., Sabitzer, B., & Lavicza, Z. (2022). Combined Effects of Block-Based Programming and Physical Computing on Primary Students' Computational Thinking Skills. *Frontiers in Psychology*, *13*(875382), 1–12. https://doi.org/10.3389/fpsyg.2022.875382

Katchapakirin, K., Anutariya, C., & Supnithi, T. (2022). ScratchThAI: A Conversation-Based Learning Support Framework for Computational Thinking Development. *Education and Information Technologies*, *27*(6), 8533–8560. https://doi.org/10.1007/s10639-021-10870-z

Kim, Y.-M., & Kim, J.-H. (2016). Application of a Software Education Program Developed to Improve Computational Thinking in Elementary School Girls. *Indian Journal of Science and Technology*, *9*(44), 1–9. https://doi.org/10.17485/ijst/2016/v9i44/105102

Kourti, Z., Michalakopoulos, C.-A., Bagos, P. G., & Paraskevopoulou-Kollia, E.-A. (2023). Computational Thinking in Preschool Age: A Case Study in Greece. *Education Sciences*, *13*(157), 1–13. https://doi.org/10.3390/educsci13020157

Kuo, W.-C., & Hsu, T.-C. (2020). Learning Computational Thinking Without a Computer: How Computational Participation Happens in a Computational Thinking Board Game. *The Asia-Pacific Education Researcher*, *29*(1), 67–83. https://doi.org/10.1007/s40299-019-00479-9

Kutay, E., & Oner, D. (2022). Coding with Minecraft: The Development of Middle School Students' Computational Thinking. *ACM Trans. Comput. Educ.*, *22*(2), 1–21. https://doi.org/10.1145/3471573

Kynigos, C., & Grizioti, M. (2018). Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space. *Informatics in Education*, *17*(2), 321–340. https://doi.org/10.15388/infedu.2018.17

Kynigos, C., & Grizioti, M. (2020). Modifying Games with ChoiCo: Integrated Affordances and Engineered Bugs for Computational Thinking. *British Journal of Educational Technology*, *51*(6), 2252–2267. https://doi.org/10.1111/bjet.12898

Kyza, E. A., Georgiou, Y., Agesilaou, A., & Souropetsis, M. (2022). A Cross-Sectional Study Investigating Primary School Children's Coding Practices and Computational Thinking Using ScratchJr. *Journal of Educational Computing Research*, *60*(1), 220–257. https://doi.org/10.1177/07356331211027387

Laura-Ochoa, L., & Bedregal-Alpaca, N. (2022). Incorporation of Computational Thinking Practices to Enhance Learning in a Programming Course. *International Journal of Advanced Computer Science and Applications*, *13*(2), 194–200. https://doi.org/10.14569/IJACSA.2022.0130224

Laura-Ochoa, L., Bedregal-Alpaca, N., & Vidal, E. (2022). Improving Computational Thinking in Nursing Students through Learning Computer Programming. *International Journal of Advanced Computer Science and Applications*, *13*(5), 600–605. https://doi.org/10.14569/IJACSA.2022.0130571

Lee, S. J., Francom, G. M., & Nuatomue, J. (2022). Computer Science Education and K-12 Students' Computational Thinking: A Systematic Review. *International Journal of Educational Research*, *114*(102008), 1–13. Scopus. https://doi.org/10.1016/j.ijer.2022.102008

Lin, X., Ma, Y., Ma, W., Liu, Y., & Tang, W. (2021). Using Peer Code Review to Improve Computational Thinking in a Blended Learning Environment: A Randomized Control Trial. *Computer Applications in Engineering Education*, *29*(6), 1825–1835. https://doi.org/10.1002/cae.22425

Lu, C., Macdonald, R., Odell, B., Kokhan, V., Demmans Epp, C., & Cutumisu, M. (2022). A Scoping Review of Computational Thinking Assessments in Higher Education. *Journal of Computing in Higher Education*, *34*(2), 416–461. https://doi.org/10.1007/s12528-021-09305-y

Luo, F., Antonenko, P. D., & Davis, E. C. (2020). Exploring the Evolution of Two Girls' Conceptions and Practices in Computational Thinking in Science. *Computers & Education*, *146*(103759), 1–12. https://doi.org/10.1016/j.compedu.2019.103759

Lyon, J. A., & J. Magana, A. (2020). Computational Thinking in Higher Education: A Review of the Literature. *Computer Applications in Engineering Education*, *28*(5), 1174–1189. https://doi.org/10.1002/cae.22295

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*, *69*(3), 1599–1616. https://doi.org/10.1007/s11423-021-10016-5

Maraza-Quispe, B., Maurice, A., Melina, O., Marianela, L., Henry, L., Cornelio, W., & Ernesto, L. (2021). Towards the Development of Computational Thinking and Mathematical Logic through Scratch. *International Journal of Advanced Computer Science and Applications*, *12*(2), 332–338. https://doi.org/10.14569/IJACSA.2021.0120242

Markandan, N., Osman, K., & Halim, L. (2022). Integrating Computational Thinking and Empowering Metacognitive Awareness in Stem Education. *Frontiers in Psychology*, *13*(872593), 1–18. https://doi.org/10.3389/fpsyg.2022.872593

Montiel, H., & Gomez-Zermeño, M. G. (2021). Educational Challenges for Computational Thinking in K–12 Education: A Systematic Literature Review of "Scratch" as an Innovative Programming Tool. *Computers*, *10*(6), 1–15. Scopus. https://doi.org/10.3390/computers10060069

Oluk, A., Korkmaz, Ö., & Oluk, H. A. (2018). Effect of Scratch on 5th Graders' Algorithm Development and Computational Thinking Skills. *Turkish Journal of Computer and Mathematics Education*, *9*(1), 54–71. https://doi.org/10.16949/turkbilmat.399588

Ouzzani, M., Hammady, H., Fedorowicz, Z., & Elmagarmid, A. (2016). Rayyan—A Web and Mobile App for Systematic Reviews. *Systematic Reviews*, *5*(1), 1–10. https://doi.org/10.1186/s13643-016-0384-4

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. *BMJ*, *372*(71), 1–9. https://doi.org/10.1136/bmj.n71

Page, M. J., Moher, D., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … McKenzie, J. E. (2021). PRISMA 2020

Explanation and Elaboration: Updated Guidance and Exemplars for Reporting Systematic Reviews. *BMJ*, *372*(160), 1–36. https://doi.org/10.1136/bmj.n160

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing Fundamental Programming Concepts and Computational Thinking with Scratchjr in Preschool Education: A Case Study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187–202. https://doi.org/10.1504/IJMLO.2016.077867

Paucar-Curasma, R., Villalba-Condori, K. O., Mamani-Calcina, J., Rondon, D., Berrios-Espezúa, M. G., & Acra-Despradel, C. (2023). Use of Technological Resources for the Development of Computational Thinking Following the Steps of Solving Problems in Engineering Students Recently Entering College. *Education Sciences*, *13*(3), 1–14. https://doi.org/10.3390/educsci13030279

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can Computational Thinking Be Improved by Using a Methodology Based on Metaphors and Scratch to Teach Computer Programming to Children? *Computers in Human Behavior*, *105*(105849), 1–25. https://doi.org/10.1016/j.chb.2018.12.027

Petrie, C. (2022). Interdisciplinary Computational Thinking with Music and Programming: A Case Study on Algorithmic Music Composition with Sonic Pi. *Computer Science Education*, *32*(2), 260–282. https://doi.org/10.1080/08993408.2021.1935603

Pou, A. V., Canaleta, X., & Fonseca, D. (2022). Computational Thinking and Educational Robotics Integrated into Project-Based Learning. *Sensors*, *22*(3746), 1–21. https://doi.org/10.3390/s22103746

Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The Impact of User Interface on Young Children's Computational Thinking. *Journal of Information Technology Education: Innovations in Practice*, *16*(1), 171–193. https://doi.org/10.28945/3768

Quiroz-Vallejo, D. A., Carmona-Mesa, J. A., Castrillón-Yepes, A., & Villa-Ochoa, J. A. (2021). Integration of Computational Thinking in Elementary and Secondary School in Latin America: A Systematic Literature Review. *Revista de Educación a Distancia*, *21*(68), 1–33. https://doi.org/10.6018/red.485321

Richard, G. T., & Giri, S. (2019). Digital and Physical Fabrication as Multimodal Learning: Understanding Youth Computational Thinking When Making Integrated Systems Through Bidirectionally Responsive Design. *ACM Transactions on Computing Education*, *19*(3), 1–35. https://doi.org/10.1145/3243138

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational Thinking and Mathematics Using Scratch: An Experiment with Sixth-Grade Students. *Interactive Learning Environments*, *28*(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

Romero, M., Lepage, A., & Lille, B. (2017). Computational Thinking Development Through Creative Programming in Higher Education. *International Journal of Educational Technology in Higher Education*, *14*(1), 1–15. https://doi.org/10.1186/s41239-017-0080-z

Rose, S. P., Jacob Habgood, M. P., & 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. https://doi.org/10.34190/ejel.15.4.2368

Shih, H., Jackson, J. M., Hawkins-Wilson, C. L., & Yuan, P.-C. (2015). Promoting Computational Thinking Skills in an Emergency Management Class with MIT App Inventor. *Computers in Education Journal*, *6*(1), 82–91. https://doi.org/10.18260/1-2--23269

Silva, R., Fonseca, B., Costa, C., & Martins, F. (2021). Fostering Computational Thinking Skills: A Didactic Proposal for Elementary School Grades. *Education Sciences*, *11*(9), 1–11. https://doi.org/10.3390/educsci11090518

Song, D., Hong, H., & Oh, E. Y. (2021). Applying Computational Analysis of Novice Learners' Computer Programming Patterns to Reveal Self-Regulated Learning, Computational Thinking, and Learning Performance. *Computers in Human Behavior*, *120*(7), 1–9. https://doi.org/10.1016/j.chb.2021.106746

Su, J., & Yang, W. (2023). A Systematic Review of Integrating Computational Thinking in Early Childhood Education. *Computers and Education Open*, *4*(100122), 1–12. https://doi.org/10.1016/j.caeo.2023.100122

Sun, L., Hu, L., & Zhou, D. (2022). Single or Combined? A Study on Programming to Promote Junior High School Students' Computational Thinking Skills. *Journal of Educational Computing Research*, *60*(2), 283–321. https://doi.org/10.1177/07356331211035182

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

Tawfik, A. A., Payne, L., & Olney, A. M. (2024). Scaffolding Computational Thinking Through Block Coding: A Learner Experience Design Study. *Technology, Knowledge and Learning*, *29*(1), 21–43. https://doi.org/10.1007/s10758-022-09636-4

Tekdal, M. (2021). Trends and Development in Research on Computational Thinking. *Education and Information Technologies*, *26*(5), 6499–6529. https://doi.org/10.1007/s10639-021-10617-w

Tikva, C., & Tambouris, E. (2021). Mapping Computational Thinking Through Programming in K-12 Education: A Conceptual Model Based on a Systematic Literature Review. *Computers & Education*, *162*(104083), 1–38. https://doi.org/10.1016/j.compedu.2020.104083

Tikva, C., & Tambouris, E. (2022). The Effect of Scaffolding Programming Games and Attitudes Towards Programming on the Development of Computational Thinking. *Education and Information Technologies*, *28*(6), 6845–6867. https://doi.org/10.1007/s10639-022-11465-y

Tran, Y. (2019). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research*, *57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Vidal-Silva, C., Cárdenas-Cobo, J., Tupac-Yupanqui, M., Serrano-Malebrán, J., & Sánchez, A. (2024). Developing Programming Competencies in School-Students with Block-Based Tools in Chile, Ecuador, and Peru. *IEEE Access*, *12*(1), 118924–118936.

Wang, C., Shen, J., & Chao, J. (2022). Integrating Computational Thinking in STEM Education: A Literature Review. *International Journal of Science and Mathematics Education*, *20*(8), 1949–1972. https://doi.org/10.1007/s10763-021-10227-5

Weber, A. M., Bastian, M., Barkela, V., Mühling, A., & Leuchter, M. (2022). Fostering Preservice Teachers' Expectancies and Values Towards Computational Thinking. *Frontiers in Psychology*, *13*(9), 1–14. https://doi.org/10.3389/fpsyg.2022.987761

Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., & Kinshuk. (2021). The Effectiveness of Partial Pair Programming on Elementary School Students' Computational Thinking Skills and Self-Efficacy. *Computers & Education*, *160*(104023), 1–65. https://doi.org/10.1016/j.compedu.2020.104023

Weng, C., Matere, I. M., Hsia, C.-H., Wang, M.-Y., & Weng, A. (2022). Effects of LEGO Robotic on Freshmen Students' Computational Thinking and Programming Learning Attitudes in Taiwan. *Library Hi Tech*, *40*(4), 947–962. https://doi.org/10.1108/LHT-01-2021-0027

Wiedemann, K., Chao, J., Galluzzo, B., & Simoneau, E. (2020). Mathematical Modeling with R: Embedding Computational Thinking into High School Math Classes. *ACM Inroads*, *11*(1), 33–42. https://doi.org/10.1145/3380956

Wing, J. M. (2017). Computational Thinking's Influence on Research and Education for All. *Italian Journal of Educational Technology*, *25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing Computational Thinking Through a Virtual Robotics Programming Curriculum. *ACM Transactions on Computing Education*, *18*(1), 1–20. https://doi.org/10.1145/3104982

Wong, G. K.-W., & Cheung, H.-Y. (2020). Exploring Children's Perceptions of Developing Twenty-First Century Skills Through Computational Thinking and Programming. *Interactive Learning Environments*, *28*(4), 438–450. https://doi.org/10.1080/10494820.2018.1534245

Wu, P.-J., Hou, H.-Y., & Huang, C.-C. (2021). Applying Talent Quality-Management System (TTQS) to Enhance Information Literacy, Learning Motivation, and Computational Thinking Competency of Nursing Undergraduates. *Sustainability (Switzerland)*, *13*(6528), 1–17. https://doi.org/10.3390/su13126528

Wu, S.-Y., & Su, Y.-S. (2021). Visual Programming Environments and Computational Thinking Performance of Fifth- and Sixth-Grade Students. *Journal of Educational Computing Research*, *59*(6), 1075–1092. https://doi.org/10.1177/0735633120988807

Xing, W. (2021). Large-Scale Path Modeling of Remixing to Computational Thinking. *Interactive Learning Environments*, *29*(3), 414–427. https://doi.org/10.1080/10494820.2019.1573199

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yang, W., Ng, D. T. K., & Su, J. (2023). The Impact of Story-Inspired Programming on Preschool Children's Computational Thinking: A Multi-Group Experiment. *Thinking Skills and Creativity*, *47*(3), 1–12. https://doi.org/10.1016/j.tsc.2022.101218

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of Computational Thinking in K-12 Mathematics Education: A Systematic Review on CT-Based Mathematics Instruction and Student Learning. *International Journal of STEM Education*, *10*(3), 1–26. https://doi.org/10.1186/s40594-023-00396-w

Yildiz Durak, H. (2020). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Technology, Knowledge and Learning*, *25*(1), 179–195. https://doi.org/10.1007/s10758-018-9391-y

Yuen, T. T., & Robbins, K. A. (2015). A Qualitative Study of Students' Computational Thinking Skills in a Data-Driven Computing Class. *ACM Transactions on Computing Education*, *14*(4), 1–19. https://doi.org/10.1145/2676660

Zeng, Y., Yang, W., & Bautista, A. (2023). Computational Thinking in Early Childhood Education: Reviewing the Literature and Redeveloping the Three-Dimensional Framework. *Educational Research Review*, *39*(100520), 1–16. https://doi.org/10.1016/j.edurev.2023.100520

Zha, S., Jin, Y., Moore, P., & Gaston, J. (2020). Hopscotch into Coding: Introducing Pre-Service Teachers Computational Thinking. *TechTrends*, *64*(1), 17–28. https://doi.org/10.1007/s11528-019-00423-0

Zhang, L., & Nouri, J. (2019). A Systematic Review of Learning Computational Thinking Through Scratch in K-9. *Computers & Education*, *141*(103607), 1–25. https://doi.org/10.1016/j.compedu.2019.103607