What concepts of Computational Thinking are being effectively used in K12 education: a Systematic Mapping

Rodrigo Hagstrom
Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)
São Paulo, Brazil
rodrigo.hagstrom@usp.br

Calebe Rezende
de Matemática e Estatística (IME)
Universidade de São Paulo (USP)
São Paulo, Brazil
calebe.rezende@usp.br

Leônidas de Oliveira Brandão
Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)
São Paulo, Brazil
leo@ime.usp.br

Anarosa Alves Franco Brandão
Escola Politécnica
Universidade de São Paulo (USP)
Sao Paulo, Brazil
anarosa.brandao@usp.br

Priscila da Silva Neves Lima
Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)
São Paulo, Brazil
priscila.snl@usp.br

***Abstract*—This research article assesses the concepts of Computational Thinking (CT) as applied in basic education. Although widely used in K12 education, many articles indicate a lack of uniformity in the conceptualization of CT. CT is defined as the systematic problem-solving ability based on algorithmic processes. It is seen as crucial for stimulating interest and diversity in STEM careers. Effective CT teaching requires precise conceptualization and appropriate teaching techniques. With the world's increasing digitization, CT is considered as fundamental as reading or math skills and should be taught from K12 education. This article reports a systematic mapping of CT concepts applied in K12 activities, reflecting on the chosen activities and teaching techniques. Despite some articles highlighting the unclear definition of CT, there is a notable agreement on its key concepts. The study used the IEEEXplore database, analyzing primary articles from 2019-2023. The findings reveal a high degree of similarity in CT concepts, indicating a trend towards standardization. A catalog of frequently cited activities used to teach CT is presented, aiding teachers in introducing CT in their classes. This study supports the development of teaching policies and initiatives, contributing to STEM interest and diversity.**

***Index Terms*—computational thinking, computational thinking concepts, K12, STEM education, STEM, STEAM**

I. Introduction

Computational Thinking (CT) has been considered for around two decades as a skill that should be taught to students not only as part of computing and technology education, but also as a support tool in solving problems across different areas of knowledge [1]–[5]. In this sense, the correct understanding of CT concepts, as well as the effective transmission of this knowledge, becomes of utmost importance.

Although CT concepts have been heavily studied and debated over the last two decades by several education and computing scholars, it is still possible to find many authors who point out a scenario of relative misunderstanding or misconceptions regarding the definition of CT, its characteristics, or its associated skills. Consequently, this lack of clear definition can lead to difficulties in the related educational processes [5]–[7].

The composition of school curricula that include Computational Thinking may be at risk of becoming incomplete if the definition itself is not a consensus. The idea that school curricula covering the topic are free from gaps becomes questionable. School curricula need well-formed definitions [5], [7], [8].

What definitions of Computational Thinking should be taught and applied in the K12 classroom? What teaching methods are most commonly used in imparting CT skills? With the premises described above and with the objective of answering these questions through comparison with experiences and teaching initiatives, this work seeks to assess which definitions of Computational Thinking have been taught in various educational initiatives in elementary school.

The remainder of the article is organized as follows: in Section II, we present a literature review and related works. In

Section III, we demonstrate the methodology adopted in this research. In Section IV, we present the results found. Finally, in Section V, we present the conclusions of the article and suggestions for future work.

II. Literature Review

In this article, the literature review is divided into two subsections. Subsection II-A reflects on the history of the definition of CT. It is necessary to consider that, before mathematics separated from computer science, the idea of algorithmic processes was already understood, as will be discussed below. This is why we begin with Pólya and Lakatos. We then reference Papert, although, as seen in [9] and in his own work [1], he rarely mentions the term "Computational Thinking." Finally, we present Wing's works. Subsection II-B offers a brief overview of CT concepts and K12 education.

A. A Brief CT Defnition History & Contributions

1) Pólya & Lakatos: George Pólya, in his book *How to Solve It* [10], introduced a method for addressing mathematical problems. His work was published before the separation of computer science from mathematics. However, we can clearly identify elements of CT in his work, especially if we consider that problem-solving and algorithms are fundamental to the definition of CT. In his book, we encounter terms or synonyms like problem-solving, algorithm (plan), logical thinking, and mathematical thinking.

Almost 30 years later, Imre Lakatos, in *Proofs and Refutations* [11], presented similar concepts for addressing problems. He uses terms such as problem-solving, "social" problem-solving, and mathematical thinking or mathematical problem-solving. Essentially, his work focuses on methods of problem-solving.

2) Papert: Papert had an immense responsibility in shaping the way we learn CT today. Papert was a South African mathematician and a student of Piaget, who introduced reflections on studying computing as a means, not as an end. CT, as brought into his work, sparked a debate between objective learning and programming tools when Papert developed LOGO[1]. The fact that, in 1985 [2], he became a founding member of the Massachusetts Institute of Technology (MIT) Media Lab exemplifies his vision of computing, which is fully supported by our review, aiming to present the most common keywords used in regular education.

It is also worth mentioning that Papert emphasizes that learning these topics without technological resources tends to become boring after a short period. In [2], antagonistic views are presented regarding the teaching of CT—one instructional and the other constructionist. The latter, which aligns with Papert's vision, details that the key is learning as a form of thought emancipation, meaning that technology is not just useful for thinking about what is learned but also for learning and thinking about the paths to follow with technology [1].

[1]It is a tool capable of being used by people of any age. In it, the student controls the entire process as they wish, without pre-established standards set by the teacher.

We can also mention that for Papert, the skills aimed at learning computational thinking include [1]:

- Compositional Reasoning: The ability to combine multiple pieces of information or solutions to subproblems to form a complete solution.
- Pattern Matching: Identifying and utilizing patterns to predict future problems or simplify complex solutions.
- Procedural Thinking: The process of executing a series of logical steps to achieve a final outcome.
- Recursive Thinking: The ability to apply the same solution or method to repeated or nested subproblems.

In our context, we can extract several defining terms from Papert's work, such as problem-solving, algorithm, logical thinking, mathematical thinking, and abstraction. However, it is important to note that Papert does not always use the same terminology found in more recent articles on these concepts. We find alternative terms in his work, synonyms (e.g., "procedural thinking" instead of "algorithmic thinking" or even "computational thinking"), as mentioned earlier. More important than the specific terms used by Papert, however, is the emphasis he placed on the contribution of computers to pedagogical processes in education [1].

3) Jeannet Wing: Probably the most cited author when it comes to CT today is Wing. Her publications have been instrumental in various fields beyond computer science.

In [3], Wing presented computational thinking as a skill that involves problem-solving, systems design, and understanding human behavior through computer science principles. She argued that these skills are essential for everyone, promoting computational thinking as a fundamental skill similar to reading, writing, and arithmetic.

Later, in [12], Wing further developed these ideas, emphasizing the broad applicability of CT across other disciplines. She discussed how this way of thinking integrates and complements mathematical and engineering principles, thereby improving problem-solving strategies and fostering innovation in various professional fields. Additionally, Wing continued to explore how CT could influence educational practices and cognitive processes. Her ideas emphasized the importance of integrating CT into educational curricula to equip future generations with the tools needed to solve complex problems in an increasingly digital world.

These contributions by Wing have shaped the discourse around CT, and in [13] she complements the definitions by adding that CT has value as a universal set of skills that improve analytical abilities and problem-solving.

As an example of her influence, [14] describes CT definitions as: decomposition, pattern identification, pattern generalization, abstraction, and algorithm design.

For the purposes of our article, we present some of the defining terms (skills) of CT according to Wing's work: abstraction, decomposition, pattern recognition, algorithmic thinking, and parallelism. However, her work focuses on a problem-solving method that can be applied to several areas of knowledge or subjects that involve complex problems, as we have seen in the works of Papert, Lakatos, and Pólya. Therefore, keeping in mind that Wing defines CT as a problem-solving method, we can find a connection between her work and the works of the authors mentioned above.

However, particularly when comparing the perspectives on CT between Papert and Wing, it is interesting to highlight that, regardless of the similarities and differences in their concepts, it is important to remember what we found in [9], which shows us that *"We will conclude that "Wing's CT" and "Papert's CT," when correctly understood, are both relevant to today's computer science education. From Wing, we should retain computer science's centrality, CT being the (scientific and cultural) substratum of the technical competencies. Under this interpretation, CT is a lens and a set of categories for understanding the algorithmic fabric of today's world. From Papert, we should retain the constructionist idea that only a social and affective involvement of students in the technical content will make programming an interdisciplinary tool for learning (also) other disciplines"*.

B. A Little Beyond Wing in K12 Education

Nowadays, the literature on CT is vast. Many authors have dedicated themselves to CT and its applications, including applications in K12 education. We can start with [15], a work that introduces ways of envisioning CT in K-12, which is important for our research. According to that work, we can describe nine skills or capabilities related to CT, which are: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization, and simulation.

[16] proposed a curriculum standard for K-12 based on CT. The work categorized skills (as themes) to be studied or applied in teaching K12 grades. The themes, as described in the article, were: Problem Solving (PS), Problem Decomposition (PD), Algorithms (AL), Data Representation (DR), Data Analysis (DA), Modeling and Simulation (MS), Abstraction (AB), Automation (AU), and Connections to Other Fields (CO).

Now that we have a better understanding of the definition of CT and some of the main authors in the field, we move on to Section III, where we discuss the methodology of our research.

III. Methodology

As mentioned earlier in this article, there are authors who argue that there may be a lack of clear definition regarding CT, and that this poses certain challenges in composing K12 curricula. As a result, some questions arise, for example:

- What are the most cited terms/abilities to compose the definition of CT?
- What are the methods, educational activities, initiatives or tools that have been used to teach these abilities?

In attempting to answer these two questions through systematic mapping, we aim to support the achievement of the main objective, which is related to the following issue:

- **What concepts of Computational Thinking are being effectively used in K12 education?**

This scenario, along with the questions mentioned, allows for a systematic mapping that can help identify the educational activities and the most frequently cited skills and competencies that have been commonly used as characteristics or definitions of CT and as skills to be taught. Therefore, we conducted a mapping of the competencies that define CT, which have been effectively applied in educational activities focused on K12 CT education.

To achieve the objectives discussed, we conducted our mapping through a literature review. On IEEE Xplore, we used queries as follows:

- ("Full Text .AND. Metadata":"**computational thinking concepts**" AND "Full Text .AND. Metadata":"basic education") OR ("Full Text .AND. Metadata":"**computational thinking concepts**" AND "Full Text .AND. Metadata":"middle education") OR ("Full Text .AND. Metadata":"**computational thinking concepts**" AND "Full Text .AND. Metadata":"K12") OR ("Full Text .AND. Metadata":"**computational thinking concepts**" AND "Full Text .AND. Metadata":"elementary school") OR ("Full Text .AND. Metadata":"**computational thinking concepts**" AND "Full Text .AND. Metadata":"high school") NOT "Full Text .AND. Metadata":"literature review" NOT "Full Text .AND. Metadata":"systematic mapping" NOT "Full Text .AND. Metadata":"literature mapping" NOT "Full Text .AND. Metadata":"mapping of literature" NOT "Full Text .AND. Metadata":"mapping of literacy" NOT "Full Text .AND. Metadata":"scientific review"

The bold text (computational thinking concepts) was the first of eight options, all synonyms, for this query component. The eight options were:

1) "computational thinking concepts";
2) "computational thinking definitions";
3) "computational thinking definition";
4) "concepts of computational thinking";
5) "definitions of computational thinking";
6) "definition of computational thinking";
7) "concept of computational thinking";
8) "computational thinking concept".

As shown in the query above, we used five terms synonymous with K12 educational levels. This was necessary because K12 can be referred to by different names, such as "basic education," which is the term used in Brazil. The synonyms we used are:

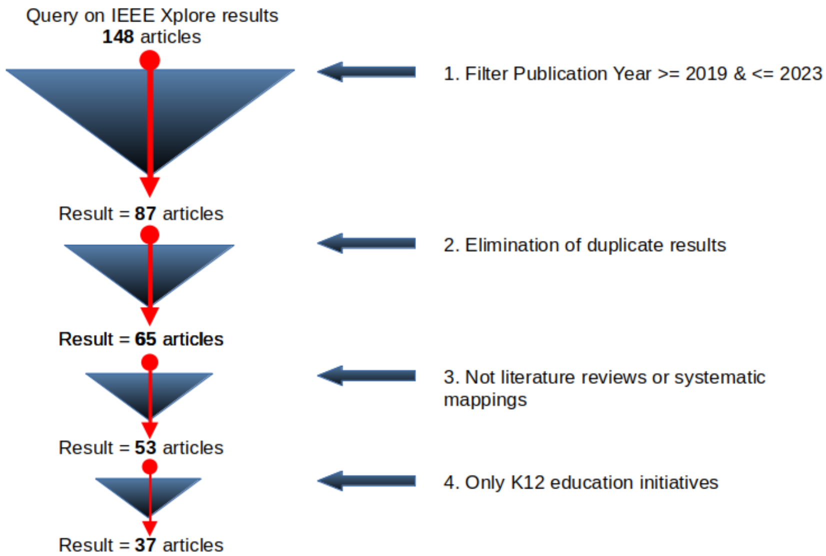

Fig. 1. A representation of our filtering process.

1) ”K12”;
2) ”basic education”;
3) ”middle school”;
4) ”elementary school”;
5) ”high school”.

We found 148 documents as results. The *IEEE Xplore* search engine automatically applied a filter to show only articles or documents published between 2012 and 2023. After that, we took the following steps:

1) filter by years from 2019 to 2023;
2) elimination of duplicate results;
3) filter which is not literature reviews, systematic mappings, systematic reviews or synonyms - this step was carried out by reading the titles or abstracts; and
4) filter which is not about K12 education initiatives - this step was carried out by reading the titles or abstracts.

Steps 3 and 4 included reading the full text in cases where the titles or abstracts were not sufficient to determine whether the article was relevant to our research objective. The number of articles resulting from each step can be seen in Figure 1. At the end of the filtering process, we had 37 articles remaining to be read.

Most of the articles analyzed contained their own literature review sections or related works, often citing definitions of CT. However, these review sections were not the focus of our research. Instead, what needed to be checked was whether the defining terms were included in the descriptions of the educational initiatives presented by the articles. To make this determination, all articles had to be read in their entirety. Only through this complete reading was it possible to discern whether or not the terms were being applied in the described educational initiatives. This process required disregarding defining terms when they appeared in citations that were not directly linked to the educational processes studied in the article. Often, these discarded citations occurred when the terms were present in literature reviews or related works sections. For example, [17] (fifth in Table I) describes several defining terms of CT, but the list of terms or competencies applied in the educational process was not as extensive as the literature review in that article suggests. However, in some cases, the terms or skills applied were cited in literature reviews or related works, as seen in the first article listed in Table I ([18]).

It is also important to note that each term was only counted once per article, regardless of whether the skill was employed in more than one activity or group of students. Therefore, when we report that a term has a frequency distribution equal to 5, it means that it appeared as an applied term in 5 articles.

Using a similar method to collect definitions used in educational processes, we also collected the methods employed to teach CT skills. These two collection processes resulted in a table listing CT articles, skills, and teaching methods. This table serves as the basis for our discussions later in this article.

In Section IV, the next section, we present our results.

IV. Results

As described above, we collected CT definition terms, skills, and methods used to teach these definitions that have been effectively applied in K12 education. This process allowed us to create Table I, which presents a detailed description of the CT terms and skills found in each of the selected papers.

Our article aims to answer several questions about teaching CT. One of them was, ”What are the most cited terms/abilities used to define CT?” The answer is that we found a strong tendency for the initiatives studied to use or apply definitions based on Papert [1] or Wing [3], as well as other definitions

TABLE I
CT DEFINITION TERMS/SKILLS AND METHODS.

ID	Definition Terms or Skills	Methods or Strategies
1 [18]	problem-solving, abstraction, decomposition, pattern recognition, algorithmic thinking	gamification, programming
2 [19]	parallelism, data representation, abstraction, decomposition, problem-solving, programming, flow control	programming language
3 [20]	algorithms, decomposition, abstraction, pattern recognition	elementary science data analysis, elementary science problem-solving
4 [21]	sequencing, looping, problem-solving, programming,	robotics, programming
5 [17]	analysing data, visualizing data	geometry problems, problem-solving
6 [22]	problem solving	binary counting, problem-solving
7 [23]	programming, problem-solving	programming
8 [24]	analysing data, visualizing data	geometry problems, problem-solving
9 [25]	sequence, programming, debugging, problem-solving	gamification, programming
10 [26]	problem-solving, programming, algorithm, sequence, loop, conditional, decomposition, abstraction, modeling, debugging	programming, algorithm, unplugged programming, plugged programming
11 [27]	programming, debugging	robotics
12 [28]	problem-solving, abstraction, modeling, debugging, decomposition, data representation, interaction, logic, flow control, parallelism, synchronization	blocks programming
13 [29]	data collection, data analysis, data representation, problem decomposition, abstraction, algorithms, automation, parallelization, simulation, problem-solving	robotics
14 [30]	abstraction, sequence, logic, problem-solving, programming, algorithm, modeling, decomposition, pattern recognition, generalization, parallelism	unplugged programming, pseudocode, programming, computer parts real world representations
15 [31]	problem-solving, programming, abstraction, design, analysis, implementation, evaluation, loop, branches, conditional, algorithm, sequence, modularity, control structures, design, debugging	gamification, graphical programming
16 [32]	algorithm, programming	programming, robotics, automation, maker processes
17 [33]	problem-solving, decomposition, programming. logic	graphical art, programming
18 [34]	problem-solving, programming, debugging, analysing, creating, executing	graphical programming
19 [35]	abstraction, algorithm, communication, conditional logic, data collection, data structures analysis, heuristics, pattern recognition, simulation , modeling.	PBL, problem-solving
20 [36]	sequence, repetition, branch, debug	graphical programming
21 [37]	algorithm, programming, problem-solving, debugging	programming
22 [38]	debugging, problem solving,	problem-solving, troubleshooting, unplugged programming
23 [39]	creativity, algorithm thinking, cooperativity, critical thinking, problem-solving, abstraction, modeling, summarization	blended teaching, programming
24 [40]	abstraction, decomposition, algorithm, programming	graphical problem-solving
25 [41]	decomposition, abstraction, problem-solving	graphical programming
26 [42]	abstraction, algorithm, problem-solving	graphical programming
27 [43]	problem-solving, programming, modeling, design	blocks programming, programming, PBL
28 [44]	algorithms, flow control, logic, parallelism, abstraction, sequence; conditions, iterations, logical operations, relational operations, variables	graphical programming, blocks programming
29 [45]	problem-solving, abstraction, algorithm, reuse, design, decomposition, debugging	PBL, problem solving, programming, unplugged programming
30 [46]	abstraction, decomposition, problem-solving, programming	gamification
31 [47]	problem-solving, programming, algorithm	lectures, graphical programming, game design, presentations, slide shows
32 [48]	algorithm, programming, decomposition, abstraction, pattern recognition	gamification
33 [49]	analyzing data, problem-solving, critical thinking, creativity, algorithm, algorithm thinking, conditional, loop, design	gamification, programming
34 [50]	testing, debugging, reuse, remixing practice, sequence, loop, event, parallelism, conditions, operators, data, problem-solving, programming	programming, robotics
35 [51]	programming, problem-solving	robotics, blocks programming
36 [52]	problem-solving, programming	programming, unplugged programming
37 [53]	problem-solving, modeling, algorithm, design, sequence	information technology lecture, CT problem-solving

related to both. For instance, "problem-solving" is cited as an applied skill in 27 out of 37 articles.

Other frequently mentioned terms include 'programming,' 'abstraction,' 'algorithm,' 'decomposition,' 'debugging,' 'sequence,' 'modeling,' 'design,' 'conditional,' 'pattern-recognition,' 'parallelism,' 'loop,' and 'logic,' all of which have a frequency distribution of five or more. Additionally, 'programming,' 'abstraction,' 'algorithm,' 'decomposition,' and 'debugging' each have a frequency distribution greater than ten.

Table II presents a frequency distribution list of the terms and definitions for CT based on the number of selected papers in which they appear, provided they appear at least twice.

TABLE II
DEFINITION/SKILL TERMS FREQUENCY DISTRIBUTION AMONG SELECTED PAPERS

Skill Term Distribution	
Term	**Freq. Distribution**
problem-solving	27
programming	20
abstraction	17
algorithm	16
decomposition	13
debugging	11
sequence	9
modeling	7
design	6
conditional	6
pattern-recognition	5
parallelism	5
loop	5
logic	5
flow-control	3
data-representation	3
analyzing-data	3
visualizing-data	2
simulation	2
reuse	2
data-collection	2
critical-thinking	2
creativity	2
branch	2
algorithm-thinking	2

Specifically regarding problem-solving and algorithms, it is important to highlight that the authors of the analyzed articles often cite Wing as the source of their definitions. However, it is also important to remember that some of the definitions included in Wing's concept of CT derive from earlier works, such as those cited in subsection II-A, written by Pólya, Lakatos, and Papert.

Our results lead us to conclude that the education community has largely reached a consensus on the definition of CT. The definition of CT, as described by Papert and later by Wing, has become almost universally accepted. Even when an article did not specifically cite "problem-solving" or "decomposition," it often referenced other skills related to Papert's and Wing's definitions.

However, we found skills that are not necessarily related to problem-solving. For example, it was common in our study to find terms like "loop," "branch," "conditional," "test," and similar terms. These terms are more related to the structures used to construct a program or an algorithm. As mentioned above, 'programming' is cited by several articles (20). Other terms related to "programming" or even computer programs have been found many times. This demonstrates that, although the most widespread definitions of CT often try not to focus solely on issues related to computer programming or algorithms, it is still common for these CT skills to be closely associated with programming structures.

Although the term "programming" itself is related to Papert's or Wing's definitions, for instance, CT is not only about Information Technology solutions, and "programming" is not necessarily limited to computer programming or even their "flow-control" (another term we found).

It is also necessary to consider that, although some CT definition terms are not being applied and therefore may not appear in many of the analyzed articles, we mentioned earlier that many authors refer to definitions or skills present in Papert's or Wing's definitions in their literature reviews or articles. In other words, even if a skill is not currently being applied, this does not diminish its status as part of the academic consensus on what defines CT. On the contrary, if a skill is widely applied in educational initiatives, it demonstrates that this skill is understood as an integral part of the CT content to be taught. Thus, our results show that there is a reasonable consensus in the academic community that studies CT, even though some studies suggest otherwise.

Perhaps the opposing view is based on the large number of definition terms that can be found in CT studies like ours. In the 37 articles, we found 55 different terms. However, even terms that appeared in only one or two articles are often related to the most cited terms. For instance, we found "parallelization," which can be a synonym for "parallelism." Similarly, we can relate "algorithmic thinking" with "algorithm," or "control structures," "branch," or "loop" with "flow control."

Thus, we can conclude that Papert's and Wing's definitions, and related definitions, have been accepted as the definition of CT, at least based on our research results. The divergences in definition terms have shown weak evidence.

Most important to our study is the fact that the definitions or skills that have been effectively applied to K12 students include "problem-solving," "programming," "abstraction," "algorithm," "decomposition," "debugging," "sequence," "modeling," "design," "conditional," "pattern recognition," "parallelism," "loop," and "logic." In other words, problem-solving skills and programming/algorithm skills. Terms like "loop" and "flow-control" show us how some teaching initiatives strongly relate CT to algorithms and computer programming. Finally, we can state that our results, in addition to not confirming a lack of definition about what CT is, since agreement with Papert's definitions was demonstrated, also confirm that these same definitions have been applied in real CT teaching initiatives in K12 (elementary and high school).

Another question we proposed was, "What are the methods, educational activities, initiatives, or tools that have been used to teach these abilities?" The analysis of our results for this question can also be started through Table I, in the third column, and in Table III. We found several articles mapping both approaches to CT (problem-solving skills and programming/algorithm skills) to programming activities.

In fact, more than 20 articles cited programming activities as ways to teach CT. In some cases, we have unplugged programming and programming (text, code, programming language) within the same teaching initiative. However, it is important to keep in mind that programming teaching/learning activities are often accompanied by other activities, such as problem-solving, gamification, or other programming methods, like blocks or graphical or visual programming.

TABLE III
METHOD TERMS FREQUENCY DISTRIBUTION AMONG SELECTED PAPERS

Method Term Frequency Distribution	
Term	**Freq. Distribution**
programming	16
problem-solving	7
graphical-programming	7
gamification	6
robotics	6
unplugged-programming	5
blocks-programming	4
PBL	3
geometry-problems	2
troubleshooting	1
slide-shows	1
pseudocode	1
presentations	1
plugged-programming	1
maker-processes	1
lectures	1
information-technology-lecture	1
graphical-problem-solving	1
graphical-art	1
game-design	1
elementary-science-problem-solving	1
elementary-science-data-analysis	1
computer-parts-real-world-representations	1
blended-teaching	1
binary-counting	1
automation	1
algorithm	1
CT	1

There are some cases where we found subjects unrelated to computer science, such as arts and elementary sciences. Although CT definitions reviewed above are skills that can be applied across various subjects, our research found that most articles were related to teaching programming. This may be because the articles came from an engineering and computing articles repository. However, since a problem solved by a program is not only about computing or engineering, a programming class can be applied to solve or address different subjects, even art, as we can see in [33].

Other methods that can be highlighted include those described as "problem-solving" or PBL[2]. These methods are directly related to the "problem-solving" definition term. They have been used with or without programming teaching processes.

Also highlighted are robotics and gamified activities, which are not necessarily separate from programming and problem-solving.

Therefore, the most frequently used methods for teaching CT include programming activities, problem-solving activities, gamification, and robotics.

When we analyze the activities and teaching methods of computational thinking, it becomes clear that there is a strong association with activities involving computers, even though we have cited cases where programming is unplugged. However, when we consider activities such as robotics or the use

[2]Project-Based Learning, Place-Based Learning, and Problem-Based Learning [45].

of programming languages, it is evident that these activities are still closely linked to the use of computers. It is also noteworthy that the term "programming" has been among the most frequently mentioned, not only as a skill that constitutes the definition of CT but also as a method, a means for teaching CT.

This also demonstrates a certain alignment or consensus with Papert's vision regarding the use of computers as a teaching tool. However, it is essential for educators to understand that Papert's vision did not limit the use of computers to teaching programming or computing. The vision presented in [1] advocates for the use of computing in education in a broad sense, utilizing computational potential to teach skills that can address the educational needs of various subjects. In our research, we found only a few articles that clearly achieved this objective: using computers and computational thinking in disciplines beyond computing.

To succinctly answer the question of what methods have been used to teach CT, we can say that the community has primarily used programming classes, often focused on computer programming.

Next, we move on to Section V, where we present the conclusion of the article.

V. Conclusion & Future Works

Considering the possibility of disagreements or misconceptions about what CT concepts entail, this work investigated which CT concepts and skills have been effectively applied in K12 education and the types of learning activities in which they were employed. Almost 60 different terms were identified for CT skills and definitions, with a strong concentration and similarity to concepts derived from the works of Seymour Papert and Jeannette Wing. Problem-solving and skills related to algorithms and programming were almost universally accepted in the articles analyzed.

Regarding the teaching activities most commonly used to learn computational thinking, there was a strong concentration in programming, gamification, and robotics teaching activities, as well as in activities that the authors describe as problem-solving.

For future work, we intend to deepen the historical study of CT definition terms. Additionally, we plan to expand our systematic mapping to include articles from academic databases beyond IEEE Xplore. Furthermore, we aim to enhance the means of supporting research and building curricula to increase the presence of CT in K12 education.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

We thank Mr. Marco Antonio de Sousa Reis, Mr. Adriano Lívio Ribeiro Alves, and Mr. Daniel de Carvalho Alves Filho for their support.

References

[1] S. Papert, *Mindstorms: children, computers, and powerful ideas*. USA: Basic Books, Inc., 1980.

[2] S. Papert and I. Harel, "Situating constructionism," *constructionism*, vol. 36, no. 2, pp. 1–11, 1991.

[3] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006. [Online]. Available: https://dl.acm.org/doi/10.1145/1118178.1118215

[4] ——, "Pensamento computacional," *Educação e Matemática*, no. 162, pp. 2–4, 2021.

[5] R. Ismail, T. A. Steinbach, and C. S. Miller, "A Guide Towards a Definition of Computational Thinking in K-12," in *2022 IEEE Global Engineering Education Conference (EDUCON)*. Tunis, Tunisia: IEEE, Mar. 2022, pp. 801–810. [Online]. Available: https://ieeexplore.ieee.org/document/9766703/

[6] A. L. S. O. De Araujo, W. L. Andrade, and D. D. Serey Guerrero, "A systematic mapping study on assessing computational thinking abilities," in *2016 IEEE Frontiers in Education Conference (FIE)*. Erie, PA, USA: IEEE, Oct. 2016, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/7757678/

[7] F. Xu and S. Zhang, "Understanding the Source of Confusion with Computational Thinking: A Systematic Review of Definitions," in *2021 IEEE Integrated STEM Education Conference (ISEC)*. Princeton, NJ, USA: IEEE, Mar. 2021, pp. 276–279. [Online]. Available: https://ieeexplore.ieee.org/document/9764144/

[8] C. Duncan and T. Bell, "A Pilot Computer Science and Programming Course for Primary School Students," in *Proceedings of the Workshop in Primary and Secondary Computing Education*. London United Kingdom: ACM, Nov. 2015, pp. 39–48. [Online]. Available: https://dl.acm.org/doi/10.1145/2818314.2818328

[9] M. Lodi and S. Martini, "Computational Thinking, Between Papert and Wing," *Science & Education*, vol. 30, no. 4, pp. 883–908, Aug. 2021. [Online]. Available: https://link.springer.com/10.1007/s11191-021-00202-5

[10] G. Pólya, "How to solve it. Princeton," *New Jersey: Princeton University*, 1945.

[11] I. Lakatos, *Proofs and Refutations: The Logic of Mathematical Discovery*, ser. Cambridge Philosophy Classics. Cambridge University Press, 1976. [Online]. Available: https://books.google.com.br/books?id=1n6SFdXCOBQC

[12] J. M. Wing, "Computational thinking and thinking about computing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1881, pp. 3717–3725, 2008, publisher: The Royal Society London.

[13] J. Wing, "Research notebook: Computational thinking—What and why," *The link magazine*, vol. 6, pp. 20–23, 2011.

[14] G. Michaelson, "Teaching programming with computational and informational thinking," 2015, publisher: University of Bedfordshire.

[15] V. Barr and C. Stephenson, "Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?" *Acm Inroads*, vol. 2, no. 1, pp. 48–54, 2011, publisher: ACM New York, NY, USA.

[16] H.-C. Chuang, C.-F. Hu, C.-C. Wu, and Y.-T. Lin, "Computational thinking curriculum for K-12 education–a delphi survey," in *2015 International Conference on Learning and Teaching in Computing and Engineering*. IEEE, 2015, pp. 213–214.

[17] L. Echeverria, R. Cobos, and M. Morales, "Improving the Students Computational Thinking Skills With Collaborative Learning Techniques," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 14, no. 4, pp. 196–206, Nov. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8894360/

[18] Y. R. Aslina, A. Mulyanto, and G. S. Niwanputri, "Designing "Bebras" Serious Games Interaction for Indonesian Upper Elementary School Students," in *2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA)*. Tokoname, Japan: IEEE, Sep. 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9429039/

[19] S. Pacheco-Portuguez, A. Gonzalez-Torres, L. Sancho-Chavarria, I. Trejos-Zelaya, J. Monge-Fallas, J. Navas-Su, A. J. Canas, A. Rodriguez, and C. A. Chinchilla, "A method for assessing computational thinking in students using source code analysis," in *2022 International Conference on Advanced Learning Technologies (ICALT)*.

Bucharest, Romania: IEEE, Jul. 2022, pp. 144–146. [Online]. Available: https://ieeexplore.ieee.org/document/9853800/

[20] J. Pietros, S. Sweetman, and M. Shim, "Teacher Practices for Computational Thinking Implementation in Elementary Science," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA: IEEE, Dec. 2022, pp. 2059–2065. [Online]. Available: https://ieeexplore.ieee.org/document/10216466/

[21] T. R. Groechel, I. Goktan, K. Ly, A.-M. Velentza, and M. J. Matarić, "MoveToCode: An Embodied Augmented Reality Visual Programming Language with an Autonomous Robot Tutor for Promoting Student Programming Curiosity," in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. Busan, Korea, Republic of: IEEE, Aug. 2023, pp. 2533–2540. [Online]. Available: https://ieeexplore.ieee.org/document/10309312/

[22] E. Nersesian, M. Vinnikov, J. Ross-Nersesian, A. Spryszynski, and M. J. Lee, "Middle School Students Learn Binary Counting Using Virtual Reality," in *2020 IEEE Integrated STEM Education Conference (ISEC)*. Princeton, NJ, USA: IEEE, Aug. 2020, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9397810/

[23] M. Seraj, C. S. Grosse, S. Autexier, and R. Drechsler, "Look What I Can Do: Acquisition of Programming Skills in the Context of Living Labs," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. Montreal, QC, Canada: IEEE, May 2019, pp. 197–207. [Online]. Available: https://ieeexplore.ieee.org/document/8802089/

[24] L. Echeverria, R. Cobos, M. Morales, F. Moreno, and V. Negrete, "Promoting Computational Thinking Skills in Primary School Students to Improve Learning of Geometry," in *2019 IEEE Global Engineering Education Conference (EDUCON)*. Dubai, United Arab Emirates: IEEE, Apr. 2019, pp. 424–429. [Online]. Available: https://ieeexplore.ieee.org/document/8725088/

[25] J. Liu, Y. Bai, and X. Xia, "Research on Developing Computational Thinking of Middle School Students Based on Gamified Text Programming Teaching Activities," in *2023 5th International Conference on Computer Science and Technologies in Education (CSTE)*. IEEE, 2023, pp. 175–179.

[26] Z. Lin and M. Ke, "Instructional Procedure and Strategy of Primary School Programming Education Focusing on Cultivating Computational Thinking," in *2020 Ninth International Conference of Educational Innovation through Technology (EITT)*. Porto, Portugal: IEEE, Dec. 2020, pp. 240–245. [Online]. Available: https://ieeexplore.ieee.org/document/9320665/

[27] C. A. Berry, A. M. Larregola, K. Collins, and J. McGee, "Robotics for the Streets: Open-Source Robotics for Academics," in *2023 IEEE Frontiers in Education Conference (FIE)*. College Station, TX, USA: IEEE, Oct. 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10343348/

[28] L. Xu, M.-W. Tong, B. Li, J. Meng, and C.-Y. Fan, "Application of Concept Map in the Study of Computational Thinking Training," in *2019 14th International Conference on Computer Science & Education (ICCSE)*. Toronto, ON, Canada: IEEE, Aug. 2019, pp. 454–459. [Online]. Available: https://ieeexplore.ieee.org/document/8845505/

[29] S. Isabelle M. L., W. L. Andrade, and S. Livia M. R., "Analyzing the Effect of Computational Thinking on Mathematics through Educational Robotics," in *2019 IEEE Frontiers in Education Conference (FIE)*. Covington, KY, USA: IEEE, Oct. 2019, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9028384/

[30] F. F. Da Silva, L. Beatrys Ruiz Aylon, and D. E. Flor, "Teaching Computational Thinking to a Student with Attention Deficit Through Programming," in *2020 IEEE Frontiers in Education Conference (FIE)*. Uppsala, Sweden: IEEE, Oct. 2020, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/9273915/

[31] M. C. Utesch, N. D. Faizan, H. Krcmar, and R. Heininger, "Pic2Program - an Educational Android Application Teaching Computational Thinking," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. Porto, Portugal: IEEE, Apr. 2020, pp. 1493–1502. [Online]. Available: https://ieeexplore.ieee.org/document/9125087/

[32] D. E. Flor, E. Henrique Molina Da Cruz, A. T. Possebom, C. Roberto Beleti Junior, R. Hubner, and L. B. Ruiz Aylon, "MannaTeam: a case of interinstitutional collaborative learning and Education 5.0," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA: IEEE, Dec. 2020, pp. 964–970. [Online]. Available: https://ieeexplore.ieee.org/document/9458164/

[33] F. Kulanova, D. Baitureev, and C. P. Singh, "Fostering Student Competencies in Creative Project Work Through STEAM and Artificial Intelligence in Teaching Art," in *2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM)*. Dubai, United Arab Emirates: IEEE, Dec. 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10449552/

[34] A. M. D. S. Esteves, A. L. M. Santana, and R. Lyra, "Use of Augmented Reality for Computational Thinking Stimulation through Virtual," in *2019 21st Symposium on Virtual and Augmented Reality (SVR)*. Rio de Janeiro, Brazil: IEEE, Oct. 2019, pp. 102–106. [Online]. Available: https://ieeexplore.ieee.org/document/8921049/

[35] S. Feng and D. Yang, "Teachers' Perspective on Implementing Computational Thinking in Elementary Classrooms," in *2021 IEEE Frontiers in Education Conference (FIE)*. Lincoln, NE, USA: IEEE, Oct. 2021, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9637221/

[36] R. Yano, H. Tanioka, K. Matsuura, M. Sano, and T. Ueta, "Quantitative Measurement and Analysis to Thinking as a Way of Programming for Elementary School in Japan," in *2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI)*. Kitakyushu, Japan: IEEE, Sep. 2020, pp. 163–168. [Online]. Available: https://ieeexplore.ieee.org/document/9430269/

[37] W. Xiduo, L. Jialin, and Z. Zhizhen, "Sustainable C++ Education in General High School: from Teaching Programming Skills to Developing Computational Thinking," in *2020 15th International Conference on Computer Science & Education (ICCSE)*. Delft, Netherlands: IEEE, Aug. 2020, pp. 35–38. [Online]. Available: https://ieeexplore.ieee.org/document/9201830/

[38] T. Michaeli and R. Romeike, "Developing a Real World Escape Room for Assessing Preexisting Debugging Experience of K12 Students," in *2021 IEEE Global Engineering Education Conference (EDUCON)*. Vienna, Austria: IEEE, Apr. 2021, pp. 521–529. [Online]. Available: https://ieeexplore.ieee.org/document/9453972/

[39] F. Zhao and S. Liu, "Research on blended teaching reform based on computational thinking," in *2022 12th International Conference on Information Technology in Medicine and Education (ITME)*. Xiamen, China: IEEE, Nov. 2022, pp. 184–188. [Online]. Available: https://ieeexplore.ieee.org/document/10086259/

[40] F. Wang, L. Chen, and W. Wang, "Design and Effectiveness Analysis of Interactive Strategies for Teaching Graphical Programming Oriented to Cultivate Computational Thinking," in *2023 5th International Conference on Computer Science and Technologies in Education (CSTE)*. Xi'an, China: IEEE, Apr. 2023, pp. 10–14. [Online]. Available: https://ieeexplore.ieee.org/document/10261600/

[41] H. Tanioka and R. Yano, "Development and Evaluation of Quizzes Aimed at Quantifying Computational Thinking," in *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*. Niigata, Japan: IEEE, Jul. 2021, pp. 188–191. [Online]. Available: https://ieeexplore.ieee.org/document/9790933/

[42] J. Liu, X. Xia, Y. Bai, and P. Liu, "A Study of Micro:bit Teaching Design Based on the Development of Computational Thinking of Secondary School Students," in *2022 4th International Conference on Computer Science and Technologies in Education (CSTE)*. Xi'an, China: IEEE, May 2022, pp. 198–201. [Online]. Available: https://ieeexplore.ieee.org/document/9973092/

[43] G. Kalogeratos, A. Alexandropoulou, and C. Pierrakeas, "Digital and Socio Emotional Benefits of the Students and the Teachers from the Implementation of a Steam Education Project," in *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*. Volos, Greece: IEEE, Jul. 2023, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/10345899/

[44] E. C. Oliveira, R. A. Bittencourt, and R. P. Trindade, "Introduction to Computational Thinking for K-12 Educators through Distance Learning," in *2019 IEEE Frontiers in Education Conference (FIE)*. Covington, KY, USA: IEEE, Oct. 2019, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/9028492/

[45] D. Zhao and Y. Li, "Research on the Educational Model of Computational Thinking Cultivation in Primary and Middle Schools Oriented to Production-Based Learning," in *2022 11th International Conference on Educational and Information Technology (ICEIT)*. Chengdu, China: IEEE, Jan. 2022, pp. 10–15. [Online]. Available: https://ieeexplore.ieee.org/document/9690630/

[46] X. Ren, T. Cai, H. Wang, and D. Yang, "Research on the Development of Secondary Vocational Students' Computational Thinking Based

on Gamiflcation Teaching : —Taking the Secondary Vocational Information Technology Course as an Example," in *2021 2nd International Conference on Information Science and Education (ICISE-IE)*. Chongqing, China: IEEE, Nov. 2021, pp. 331–334. [Online]. Available: https://ieeexplore.ieee.org/document/9742585/

[47] T. Veron, T. Iles, A. A. Khan Ashar, and J. Liu, "Computer Science (CS) Day for Middle School Students," in *2023 IEEE Frontiers in Education Conference (FIE)*. College Station, TX, USA: IEEE, Oct. 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10343497/

[48] I. Yuliana, L. P. Octavia, E. Sudarmilah, and M. Matahari, "Introducing Computational Thinking Concept Learning in Building Cognitive Capacity and Character for Elementary Student," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*. Ho Chi Minh City, Vietnam: IEEE, Sep. 2019, pp. 549–554. [Online]. Available: https://ieeexplore.ieee.org/document/8905149/

[49] Z. Qu, J. Liu, L. Che, Y. Su, and W. Zhang, "Research on the Application of Gamification Programming Teaching for High School Students' Computational Thinking Development," in *2023 IEEE 12th International Conference on Educational and Information Technology (ICEIT)*. Chongqing, China: IEEE, Mar. 2023, pp. 144–149. [Online]. Available: https://ieeexplore.ieee.org/document/10107843/

[50] R. P. Curasma, N. J. Jara, H. P. Curasma, and V. C. Ornetta, "Assessment of Computational Thinking in regular basic education: case I.E.T.P. "José Obrero"," in *2019 IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. Lima, Peru: IEEE, Aug. 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8853613/

[51] P. Bachiller-Burgos, I. Barbecho, L. V. Calderita, P. Bustos, and L. J. Manso, "LearnBlock: A Robot-Agnostic Educational Programming Tool," *IEEE Access*, vol. 8, pp. 30 012–30 026, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8986589/

[52] R. Herrero-Álvarez, G. Miranda, C. León, and E. Segredo, "Engaging Primary and Secondary School Students in Computer Science Through Computational Thinking Training," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 1, pp. 56–69, Jan. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9749967/

[53] M. Zhao and W. Hu, "A practical research on the training of middle school students' computational thinking based on problem solving in systematic information process view," in *2022 3rd International Conference on Education, Knowledge and Information Management (ICEKIM)*. Harbin, China: IEEE, Jan. 2022, pp. 1037–1040. [Online]. Available: https://ieeexplore.ieee.org/document/10027412/