frontiers | Frontiers in Education

TYPE Original Research
PUBLISHED 29 September 2022
DOI 10.3389/feduc.2022.933316

Check for updates

OPEN ACCESS

EDITED BY
Attila Gilanyi,
University of Debrecen, Hungary

REVIEWED BY
Russ Meier,
Milwaukee School of Engineering,
United States
Karin Tengler,
Pädagogische Hochschule
Niederösterreich, Austria

*CORRESPONDENCE
Pálma Rozália Osztián
osztian.palma@ms.sapientia.ro

SPECIALTY SECTION
This article was submitted to
Digital Learning Innovations,
a section of the journal
Frontiers in Education

RECEIVED 30 April 2022
ACCEPTED 29 August 2022
PUBLISHED 29 September 2022

CITATION
Osztián PR, Kátai Z and Osztián E
(2022) On the computational thinking
and diagrammatic reasoning of
first-year computer science and
engineering students.
Front. Educ. 7:933316.
doi: 10.3389/feduc.2022.933316

COPYRIGHT
© 2022 Osztián, Kátai and Osztián.
This is an open-access article
distributed under the terms of the
Creative Commons Attribution License
(CC BY). The use, distribution or
reproduction in other forums is
permitted, provided the original
author(s) and the copyright owner(s)
are credited and that the original
publication in this journal is cited, in
accordance with accepted academic
practice. No use, distribution or
reproduction is permitted which does
not comply with these terms.

On the computational thinking and diagrammatic reasoning of first-year computer science and engineering students

Pálma Rozália Osztián[1]*, Zoltán Kátai[2] and Erika Osztián[2]

[1]Faculty of Informatics, University of Debrecen, Debrecen, Hungary, [2]Mathematics and Informatics Department, Sapientia Hungarian University of Transylvania, Târgu Mureş, Romania

Computational thinking (CT) and diagrammatic reasoning (DR) are important competencies from the perspective of both Computer Science and Engineering education. CT is often described as a critically important attitude or skill set for all students regardless of the educational program in which they are enrolled. Diagrammatic reasoning is commonly referred to as a student's ability to think logically and solve complex problems. Accordingly, these two competencies are closely related and both skills are parallelly linked to several curriculum subjects (with preponderance in the case of STEM disciplines) during the educational process. Consequently, one might conclude that even without an explicit focus on them, students might develop these abilities latently as they advance with the K-12 current curriculum. We have proposed to test this assumption. In the experiment, 137 first-year students were involved in six different Computer Science and Engineering educational programs. Students were invited to participate in a CT and a DR test. We were particularly interested in possible correlations between the results of the two tests. Our results confirmed that computational thinking and diagrammatic reasoning are closely related abilities. We also found that CT, DR, and students' prior programming experience positively correlate with their first course exam results in Computer Science.

KEYWORDS

computational thinking, diagrammatic reasoning, STEM education, K-12 education, computing

1. Introduction

The importance of STEAM (Science, Technology, Engineering, Art, and Mathematics) education has become increasingly emphasized in recent decades. Since the 1990s, there has been a STEM movement (Martín-Páez et al., 2019) that has had an impact on both K-12 education (Holmlund et al., 2018) and higher education (National Academies of Sciences and Medicine, 2018). Although most previous research has focused on either K-12 or higher education (HE), these two main (and consecutive) stages of the educational process are strongly interconnected. Since the STEM concept is particularly prominent in Computer Science and Engineering Education (CSE, ED), we concentrated on these branches of higher education.

One way to address the K-12—HE connection in the context of STEM education is to focus on key competencies. Two such competencies that are critically important from the perspective of both CSE and ED are computational thinking (CT) and diagrammatic reasoning (DR). For example, Lyon and J. Magana (2020) in their review of literature conclude that CT is of growing interest to the STEM education research community. In addition, Kiernan et al. (2021) describe DR as a learning strategy central to STEM disciplines. In this research, we examined the CT and DR of first-year CS and Engineering students with a particular focus on possible correlations between these two competencies.

2. Connecting CT and DR to STEM

STEM education has a decisive role in our modern society because of the unique nature of the referred four fields (Xie et al., 2015). While it is widely admitted that the acronym STEM is related to a set of science-oriented domains or subjects, there is no consensus on the definition of this set (Gonzalez and Kuenzi, 2012). In this study, we focus on those disciplines (or subjects) that have been included in the Romanian K-12 educational program. In addition to this subject-oriented approach, STEM education can also be addressed from the perspective of impacted abilities. Two key abilities in this sense are computational thinking and diagrammatic reasoning.

Although the expression CT was introduced by Papert (1980), the modern wave of this expression was generated by Wing (2006) who defined it as a "universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use." According to Lodi and Martini (2021), a common idea suggested by both Papert and Wing is that the competencies acquired as CT can easily be transferred to other disciplines. More recent definitions of CT are also in line with this conclusion. For example, Aho (2012) explained CT as "the thought processes involved in formulating problems so that their solutions can be represented as computational steps and algorithms." Shute et al. (2017) describe CT as "the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts." With respect to K-12 education, Csizmadia et al. (2015) emphasized the following five key computational skills: abstraction, decomposition, algorithmic thinking, evaluation, and generalization. These related skills can be witnessed in many domains with preponderance in STEM disciplines. In line with this, some authors suggest that since CT is a combined skill with cross-disciplinary implications, students' CT might develop latently (without an explicit focus on CS) as they advance with their K-12 curriculum (Katai et al., 2021).

The roots of DR go way back to Venn diagrams, which are referred to as "a graphical alternative to sentential and algebraic forms to represent logical relations" (Hoffmann, 2011). Moreover, some authors trace the concept back to Euclid, which inspired both Peirce and Polya. Sowa (2020) stresses the value of combining the diagrams proposed by Euclid, the graph logic introduced by Peirce, and the heuristics suggested by Polya. According to Peirce (1868), there is a duality in DR due to the fact that it can be considered as a tool to generate knowledge but also as a "solution of problems of Logic" (Hoffmann, 2011). In addition, Hoffmann emphasizes the importance of interplay between diagrams and reasoning with the goal of promoting human creativity and learning. Based on Stieff et al. (2010), Stieff (2011), and Kiernan et al. (2021), DR can be seen as "the application of heuristics or algorithms to domain-specific diagrams which enable students to deduce complex spatial transformations without necessarily invoking mental images." With all this in mind, it is not surprising that researchers often link diagrammatic thinking to STEM education (Kiernan et al., 2021). For example, STEM instructors regularly induce DR by creating sketches that model, characterize and communicate the essence of the studied scientific phenomenon (Latour, 1990).

2.1. The impact of K-12 STEM subjects on CT and DR

According to Baran et al. (2016), STEM education is an interdisciplinary teaching-learning strategy that integrates science, technology, engineering, mathematics, and other knowledge, skills, and beliefs particular to these disciplines. In line with this, and based on the European recommendations, the Romanian K-12 framework curriculum highlights the importance of supporting all students in developing such abilities as mathematical and digital competencies and basic competencies in sciences and technology. In Romania, the curriculum includes some focused STEM disciplines that are introduced at different age levels. For example Mathematics (grade 1), Information and communication technology (grade 5), Introduction to computer algorithms (grade 5), Science (chemistry and physics; grade 6), Technology (to prepare students from specific classes for engineering education; grade 9), and Computer Science (focused computer programming for informatics classes; grade 9).

Prior research explicitly links CT and DR to these subjects. For example, the Next Generation Science Standards (Next Generation Science Standards, 2013) describe CT as a core scientific practice. Wing (2016), ten years after publishing the above referred defining article on CT, emphasizes that computation can be seen as the third pillar of modern science and engineering disciplines, supplementing theory and experimentation. Weintrop et al. (2016) proposed a taxonomy-based definition of CT, particularly for mathematics and science. These authors addressed four main areas: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices. According to these trends, there is a growing number of initiatives for infusing CT in STEM disciplines [e.g., mathematics: (Wilkerson and Fenwick, 2017); physics: (Hutchins et al., 2020); chemistry: (Chongo et al., 2021), etc.].

Computational thinking can be linked to CSE even more directly. In recent decades, authors have reported on several plugged-in and unplugged methods that promote students' CT in this context at all levels (Brackmann et al., 2017). Obviously, teaching-learning programming remains a distinct method in this sense (Mannila et al., 2014), being considered the mainstream approach. Of course, by programming, we mean more than just coding. Relevant programming practice exposes students to the three main dimensions of CT: computational concepts, computational practices, and computational perspectives. Lye and Koh (2014), after reviewing 27 intervention studies, concluded that programming has the potential to foster CT. More recently, Tikva and Tambouris (2021) stated that CT through programming is considered "an ideal medium for the development of twenty-first century skills."

Understandably, DR is intertwined with the teaching of STEM subjects earlier than CT. For example, diagrams, as a type of representation of mathematical objects, are long-established tools in mathematics education. Interestingly, according to Sochański (2018), despite all the criticisms that DR could be at most a secondary tool for mathematics education, in recent years we are witnessing a revival of interest in the role of visualization in mathematical practice. Novak (1977) developed a DR-oriented system (ISAAC) for solving physics problems more than 50 years ago. A defining particularity of the work was that diagrams were considered as an integrated part of understanding and solving the addressed problems. With respect to chemistry education, Kiernan et al. (2021) reported on research that investigated visuospatial thinking through students' use of imagistic, analytical, and diagrammatic reasoning when predicting molecular geometry. According to these authors, DR has the potential to reduce learners' visuospatial load by providing an analytical route. In the context of CS education, perhaps the most commonly used diagrams are the logical schemes for representing algorithms. Creating and analyzing such representations are an integral part of programming education and clearly assume DR.

3. Research questions

Based on the above literature review, it can be assumed that first-year CS and Engineering students possess a certain level of CT and DR because of the implicit contribution of their K-12 education. In general, these students had placed an emphasis on STEM disciplines. In Romania, usually, students are admitted to science-oriented Higher Education programs mostly based on their high school graduation exam scores in STEM subjects. In addition, it can also be assumed that students' prior programming experience is reflected in their CT level. We have proposed to test these assumptions. Furthermore, we were also interested in the extent to which students' CT and DR scores at the beginning of the first semester predict their first course in Computer Science (CS1) exam results.

Accordingly, we addressed the following research questions:

RQ1: Is there a correlation between CT and DR test results?
RQ2: To what extent does prior programming experience benefits students' results?
RQ3: To what extent do CT and DR results predict students' performance on their CS1 exam?

4. Methods

4.1. Participants and procedure

The experiment took place at an Eastern European University during the registration week of the academic year (just before the first semester begins). A total of 137 (17% females) first-year students were invited to participate in the study. Subjects were from different majors: Informatics (34%), Computer Science (22%), Automation and applied informatics (10%), Mechatronics (15%), Telecommunications (8%), and Mechanical Engineering (11%). Subjects also showed considerable diversity in terms of their high school academic profiles: Theoretical (Natural Sciences—21.2%, Mathematics and Computer programming—51.8%), Technical (16.8%), Services and Economics (5.8%), and others (4.4%).

According to the participants' previous academic profiles, their prior programming experience in years (PPY) differed. The reason for having different PPY is that programming is not compulsory in Romania and each high school is free to offer one or more academic programs. One of the most demanding academic programs is the Science profile where students have the opportunity to get STEM-related degrees. During the four academic years, students have 4–7 h/week of math classes. Furthermore, students from natural sciences classes learn computer programming for 2 h/week for 2 years (2 PPY), while students from mathematics and computer programming classes can have computer programming classes up to 5–7 h/week for 4 years (4 PPY). Besides the above-mentioned academic programs, none of the other academic profiles include computer programming classes. Consequently, students who come from these profiles have no prior programming experience (0 PPY; Table 1).

TABLE 1 Distribution of participants by prior programming experience.

0 PPY	1–3 PPY	4 PPY
36 (26%)	38 (28%)	63 (46%)

4.2. Design

In order to investigate the correlation between students' CT, DR, and CS1 results a positive correlational research design was used. The aim of this study was to ensure an overall picture based on students' abilities in relation to their CS1 scores. More precisely, we intended to perform the CT and DR tests in the case of all participants in the same way. Therefore, assigning students to different groups was not required.

4.3. Materials

The study took place in one of the lecture halls of the university where subjects were requested to complete two types of paper-based tests:

1. DR test: to assess students' diagrammatic reasoning ability.
2. CT test: to assess students' computational thinking ability.

The testing session lasted a total of 1 h where we used questions shown in Table 2. Time was divided into 10 min for the DR test and ca. 50 min for the CT test. Both DR and CT tests were paper-based face-to-face tests that students completed all at once. First, students were asked to answer four introductory questions about their school experience details (Q1–Q3: university major, high school academic program, gender), their prior programming experience—PPY (Q4), and one privacy policy related question (Q5). Furthermore, the DR test included 10 multiple-choice questions (Q6–Q15), while the CT test consisted of three problems with different subtasks (Q16–Q25) (Table 2).

4.3.1. Diagrammatic reasoning test

The DR test consisted of 10 different questions collected from the[1] online platform which provides a variety of free psychometric aptitude tests such as diagrammatic reasoning. Psychometric Success tests are widely used to assess the aptitudes of different category people during recruitment processes and educational contexts as well (Fernandés, 2011; Bressler, 2014; Mirabueno and Boyon, 2020).

According to Newton and Bristol (2011), DR tests can be closely related to abstract reasoning tests whose aim is to process flow charts and diagrams and to predict the output by following a series of logical instructions or inferring rules. Based on Tóth et al. (2021), DR tests are excellent tools in order to measure the skill set of a person, more precisely the extent to which the person is able to follow a specific set of instructions. These kinds of tests are also well-suited for information technology-related jobs because they can reflect the software design of a project.

In our study, participants were from six different university programs all of which pay close attention to software design and architecture. This justifies the importance of examining DR tests during university courses as well. In order to measure students' DR, we defined two main categories: recognition of regularities with unknown operations (Q6–Q10) in cases in which students were required to specify the output for a given input; and recognition of regularities with known operations (Q11–Q15) where students needed to follow a set of instructions in order to find the solution. Researchers emphasized that mechanical engineering, electrical engineering, and engineering IT students achieved significantly better results. This underlines the decisive role of improving DR ability which can closely reflect inductive cognition. In our study, we used the previously mentioned DR test questions in their original form which can be found[2]. This test was also used by many other authors such as Tóth et al. (2021) who have achieved outstanding results.

4.3.2. Computational thinking test

As a further step, students were asked to complete the CT test questionnaire. Questions for this assessment were collected from the webpage of the[3] contest. The Bebras contest ensures a variety of questions divided into five different categories such as abstraction, algorithmic thinking, decomposition, evaluation, and generalization. These tasks are snatched from real-life problems (Dagienė and Sentance, 2016) which provide a realistic visualization, ensuring curiosity and motivation even for the younger generation. They also emphasized that including Bebras tasks in the curriculum can promote students' CT skills to improve even better. Further studies (Hubwieser and Mühling, 2015; Csapó, 2019) also underlined the inherent potential of these problems with the help of which computer science can become an interesting and attractive field of science.

The Bebras challenge consists of tasks that require CT and a set of problem-solving skills. Consequently, for the CT assessment presented in this study, we decided to inspire and collect tasks from the[4] webpage which is maintained by the Hungarian partner of the Bebras initiative. Bebras contest items have been frequently used to measure CT. For example, based on a validation study of selective Bebras items, Tang et al. (2018)

1 https://psychometric-success.com/

2 https://psychometric-success.com/test-pdfs/PsychometricSuccessDiagrammaticReasoning-PracticeTest1.pdf

3 https://www.bebras.org/

4 http://e-hod.elte.hu/

TABLE 2 Assessment questions.

Metacode	Item
Questions 1–5	Gender, high school academic profile, university major, PPY, privacy policy
Task 1	Participants were asked to complete the DR test.
Questions 6–10	*Unknown operations: Determine the effect of the "operators" in order to produce the "output" for a given "input."*
Questions 11–15	*Known operations: Work from top to bottom. In order to produce the "output" apply the given rules for "operators."*
Task 2	Participants were asked to complete the CT test.
Task 2.1	**Math Machine task: Given the algorithm used by a calculator,**
	determine the correct output for each input value: calculate(n),
	where n is input, and the output is the result for function calculate(n).
Question 16	*Calculate(4): What is the output for input 4?*
Question 17	*Calculate(7): What is the output for input 7?*
Question 18	*Combining partial results into one: What is the output of the following sequence of operations: calculate(2) * calculate(6) - calculate(3) * calculate(4)*
Question 19	*Generalization formula for calculate(n): What is the output for input n?*
Task 2.2	**Heat Maps task: A letter machine can recognize five images,**
	which represent the letters I, T, O, C, and L.
	The letter machine uses heat maps in the recognition process.
	The heat map can be defined as the following: we assign a value to each of the pictures' pixels.
	The value shows how many of the other images have the same pixel in that location.
Question 20	*Pattern recognition: Which letter is illustrated by the following 3 × 3 dimensional heat map: {3, 3, 2 / 2, 2, 0 / 2, 4, 2}?*
Question 21	*Pattern recognition: Which letter is illustrated by the following 3 × 3 dimensional heat map: {3, 0, 1 / 2, 2, 3 / 2, 4, 2}*
Question 22	*Find the odd one: Which heat map does not belong to any of the images listed?*
Task 2.3	**Popularity task: Seven friends are in an online social network.**
	The network allows them to see posts on their own and their friends' timelines. Two friends are connected (are friends) with a line.
Question 23	*Direct connection: Who can see X's messages?*
Question 24	*Valency of vertex: How many friends are in the network whose messages can be seen by exactly four other people?*
Question 25	*Removing edges: How can we "hide" posts' from somebody?*

concluded that their results significantly correlated with another CT test.

The importance of Bebras and e-hód tasks was also emphasized by the managing research group of the Hungarian initiative (Pluhár and Gellér, 2017). Furthermore, researchers have also highlighted the fact that with the help of these problems teachers can be able to maintain students' engagement and motivation and they can also dissolve students' fears and negative feelings regarding computer science (Pluhár et al., 2019).

In this study, we included three tasks from the Bebras challenge in the interest of examining students' CT skills. For each task, we introduced further questions in order to provide complexity-related concepts as well.

4.3.2.1. Math machine task

The original problem of the Math Machine task was assigned in 2021 at the[5] which consists of one question (Q16). We extended this problem with three more questions: Q17 elementary question, similar to the main task of the problem, Q18 which included a sequence of operations and students needed to combine partial results into one, and Q19 where a general formula was required.

4.3.2.2. Heat maps task

In the case of the second task two-dimensional representation played an important role, where different letters were described with the help of Heat Maps. The initial task was also published in 2021 at the Bebras Australia Computational Thinking Challenge and it was constituted of one main question (Q20). We addressed two more questions to this: Q21 pattern recognition task, similar to Q20 and Q22 where students were asked to find the odd Heat Map (find the incorrect Heat Map of three given Heat Maps).

4.3.2.3. Popularity task

The third and the last problem illustrated a common everyday situation, which is based on communication between each other, on social networks. Networks and graphs are one of the best ways to illustrate problems regarded to social networks and friendships. The original problem was published in 2016

5 https://digitalcareers.csiro.au//media/Digital-Careers/Files/Bebras-Files/Bebras-Handbook-2021.pdf

at the[6] where one single question was addressed (Q23). In the CT test, we included two more questions: Q24 in the case in which students needed to specify the degree of a given vertex in the social network, and Q25 where participants were asked to determine which connections should be removed from the social network.

5. Results and discussion

The analysis was carried out in the R environment. The investigation included four factors: CT-test scores, DR-test scores, prior programming years (PPY), and students' CS1 marks. Obviously, comparing the means would be of no relevance, so we focused on examining possible correlations between the variables. Although a total of 137 students were present during the testing process, two of the participants did not complete their tests appropriately, consequently, their answers were excluded.

As a first step, we compared participants' CT and DR scores (RQ-1). Results of the Spearman correlation indicated that there was a significant positive association between the two test performances, [$rs(135) = 0.39$, $p < 0.001$]. Although previous studies did not address these abilities as intertwining skills, our findings are consistent with those studies that present them as related abilities. For example, Falkner et al. (2010) reported on a successful puzzle-based learning course initiative for engineering and computer science students. These authors emphasize that their approach has the potential to lay a basis for CT in the curriculum. On the other hand, the paper highlights that the implemented syllabus also included DR puzzles. Dodig-Crnkovic and Cicchetti (2017) connected CT and DR in the context of Model-Based Reasoning. They cite Giere (2002), who described DR as "the interaction between the diagram and a human with a fundamentally pattern-matching brain." With respect to generative computational methods, these authors underlined that computing enables a new kind of science (Wolfram, 2002), and supports what Jeannette Wing termed computational thinking (Wing, 2006). More recently, the (Wan et al., 2020) study aimed to maximize learning opportunities of machine learning for K-12 students with diverse STEM skills. Artificial Intelligence education traditionally assumes computational skills. In addition, the authors point out that the proposed method facilitates understanding, knowledge discovery, and sense-making of abstract and complex concepts through visual data exploration and diagrammatic reasoning.

As a next step, we examined students' CT and DR test performances from the perspective of their prior programming experience (PPY). It is found that the correlation coefficient for the PPY vs. CT comparison [$rs(135) = 0.44, p < 0.001$] was considerably higher than in the case of the PPY vs. DR comparison [$rs(135) = 0.17, p = 0.04$]. One possible explanation could be that the SD for the DR-test was higher than for the CT-test. This finding is in line with studies that reported a strong connection between learning programming (if properly taught) and students' CT. Tikva and Tambouris (2021) state that there is a dual association between CT and programming: "Programming supports the development of CT while CT provides to programming a new upgraded role." Accordingly, although programming is not the only approach for supporting CT development and the CT-programming relationship is not clear yet (Passey, 2017), these two concepts are widely associated (Voogt et al., 2015). For example, Grover and Pea (2013) refer to programming as a key instrument for supporting the cognitive tasks that CT also implies. Other researchers, in the same line of thought, emphasize that programming provides a proper environment for implementing CT related concepts and practices (Brennan and Resnick, 2012; Basogain et al., 2018).

Finally, we compared factors CT, DR, and PPY with students' CS1 scores. One hundred and twenty-three out of 135 students had all four types of test data. We found significant positive associations in all three cases [CT vs. CS1: $rs(123) = 0.47$, $p < 0.001$; DR vs. CS1: $rs(123) = 0.39$, $p < 0.001$; PPY vs. CS1: $rs(123) = 0.48$, $p < 0.001$]. These results provide further confirmation of the above-detailed ones, namely that there is an evident connection between the examined factors: learning programming, computational thinking, and diagrammatic reasoning. In a recent study, Gjelsten et al. (2021) also found a strong positive association between studying programming at high school and CS1 performance. In addition, these authors emphasize the importance of taking math and science-heavy courses in high school which is in line with prior STEM research regarding the association between mathematics, science, and CS (Uttal et al., 2013).

Additionally, we also compared female and male students' test results. The means and the corresponding SD (in brackets) were the following: CT [females: 7.35 (1.86); males: 7.51 (2.33)], DR [females: 6.47 (2.26); males: 7.42 (2.29)], and CS1 [females: 8.21 (2.03); males: 8.00 (2.01)]. No significant differences were found in the case of CT and CS1 scores but, interestingly, male students performed significantly better during the DR test. This result might harmonize with the gender gap phenomenon in STEM education (Wang and Degol, 2017).

6. Limitations

One of the limitations of this study is that the two tests were not aligned in the sense that the exact values of the scores could be compared. Therefore, we could not consider the difference between the averages of the test results, we only examined the correlation between them. As a next step, we plan to develop

6 https://digitalcareers.csiro.au/media/Digital-Careers/Files/Bebras-Files/2016-Bebras-Solution-Guide-AU.pdf

aligned CT and DR tests which would allow for a more thorough joint examination of these two competencies.

7. Conclusion

Computational thinking and diagrammatic reasoning can be considered essential factors in every person's life in our digital era. With the help of these key abilities, students can learn to think in a different way and they can also gain a better experience in problem-solving and abstract reasoning. The present research supports the fact that computational thinking is about developing a full set of mental tools in order to solve complex everyday problems (Wing, 2006) and diagrammatic reasoning reflects fundamentally means of thought, understanding, and reasoning (Bakker and Hoffmann, 2005).

This study provided further evidence that computational thinking and diagrammatic reasoning are closely related abilities. Our findings also confirmed that proper programming education has the potential to contribute to students' computational thinking. In addition, we found that all three examined factors (CT, DR, and PPY) positively correlate with students' CS1 performance. These results emphasize the importance of K-12 STEM education, since STEM disciplines (if properly taught) provide an appropriate framework for developing students' computational thinking and diagrammatic reasoning. Teachers are encouraged to purposefully contribute to the development of these abilities.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

Ethics statement

Ethical approval was not provided for this study on human participants because, it was not applicable. Subjects were asked if their responses can be analyzed for research purposes. All data was analyzed anonymously. The patients/participants provided their written informed consent to participate in this study.

Author contributions

PO and ZK contributed to the conception and design of the study, wrote the first draft, and sections of the manuscript. PO organized the database. ZK and EO performed the statistical analysis. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This study was supported by the Collegium Talentum Programme of Hungary and the Sapientia Foundation—Institute for Scientific Research.

Acknowledgments

We would like to thank Mária Csernoch for the thoughtful comments.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author PO and handling editor declared their shared affiliation.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aho, A. V. (2012). Computation and computational thinking. *Comput J.* 55, 832–835. doi: 10.1093/comjnl/bxs074

Bakker, A., and Hoffmann, M. H. (2005). Diagrammatic reasoning as the basis for developing concepts: a semiotic analysis of students' learning about statistical distribution. *Educ. Stud. Math.* 60, 333–358. doi: 10.1007/s10649-005-5536-8

Baran, E., Bilici, S., C, M., and Ocak, C. (2016). Moving stem beyond schools: students' perceptions about an out-of-school stem education program. *IJEMST.org* 4, 9–19. doi: 10.18404/ijemst.71338

Basogain, X., Olabe, M. Á., Olabe, J. C., and Rico, M. J. (2018). Computational thinking in pre-university blended learning classrooms. *Comput. Hum. Behav.* 80, 412–419. doi: 10.1016/j.chb.2017.04.058

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., and Barone, D. (2017). "Development of computational thinking skills through unplugged activities in primary school," in *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, Nijmegen, 65–72. doi: 10.1145/3137065.3137069

Brennan, K., and Resnick, M. (2012). "New frameworks for studying and assessing the development of computational thinking," in *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vol. 1* (Vancouver, BC), 25.

Bressler, M. S. (2014). Building the winning organization through high-impact hiring. *J. Manage. Market. Res.* 15:1.

Chongo, S., Osman, K., and Nayan, N. A. (2021). Impact of the plugged-in and unplugged chemistry computational thinking modules on achievement in chemistry. *EURASIA J. Math. Sci. Technol. Educ*. 17:em1953. doi: 10.29333/ejmste/10789

Csapó, G. (2019). Placing event-action-based visual programming in the process of computer science education. *Acta Polytech. Hungar*. 16, 35–57. doi: 10.12700/APH.16.2.2019.2.3

Csizmadia, A., Curzon, P., Humphreys, S., Ng, T., Selby, C., and Woollard, J. (2015). Computational thinking-a guide for teachers.

Dagienė, V., and Sentance, S. (2016). "It's computational thinking! Bebras tasks in the curriculum," in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (Cham: Springer), 28–39. doi: 10.1007/978-3-319-46747-4_3

Dodig-Crnkovic, G., and Cicchetti, A. (2017). "Computational aspects of model-based reasoning," in *Springer Handbook of Model-Based Science*, eds L. Magnani and T. Bertolotti (Cham: Springer), 695–718. doi: 10.1007/978-3-319-30526-4_32

Falkner, N., Sooriamurthi, R., and Michalewicz, Z. (2010). Puzzle-based learning for engineering and computer science. *Computer* 43, 20–28. doi: 10.1109/MC.2010.113

Fernandés, O. (2011). Psychometric assessment and its impact on human resource practices.

Giere, R. N. (2002). "Models as parts of distributed cognitive systems," in *Model-based reasoning* (Springer). 227–241.

Gjelsten, B. K., Bergersen, G. R., Sjøberg, D. I., and Cutts, Q. (2021). "No gender difference in cs1 grade for students with programming from high school: an exploratory study," in *21st Koli Calling International Conference on Computing Education Research*, Joensuu, 1–5. doi: 10.1145/3488042.3488071

Gonzalez, H. B. and Kuenzi, J. J. (2012). *Science, Technology, Engineering, and Mathematics (stem) Education: A primer*. (Washington, DC: Congressional Research Service, Library of Congress Washington).

Grover, S., and Pea, R. (2013). A review of the state of the field–a computational thinking in K. *Educ. Res.* 42, 38–43. doi: 10.3102/0013189X12463051

Hoffmann, M. H. (2011). Cognitive conditions of diagrammatic reasoning. *Semiotica* 2011, 189–212. doi: 10.1515/semi.2011.052

Holmlund, T. D., Lesseig, K., and Slavit, D. (2018). Making sense of "stem education" in k-12 contexts. *Int. J. STEM Educ*. 5:32. doi: 10.1186/s40594-018-0127-2

Hubwieser, P., and Mühling, A. (2015). "Investigating the psychometric structure of Bebras contest: towards measuring computational thinking skills," in *2015 International Conference on Learning and Teaching in Computing and Engineering*, 62–69. doi: 10.1109/LaTiCE.2015.19

Hutchins, N. M., Biswas, G., Maróti, M., Ákos Lédeczi, Grover, S., Wolf, R., et al. (2020). *C2stem: A System for Synergistic Learning of Physics and Computational Thinking, Vol. 29*. Springer, 83–100. doi: 10.1007/s10956-019-09804-9

Katai, Z., Osztian, E., and Lorincz, B. (2021). Investigating the computational thinking ability of young school students across grade levels in two different types of Romanian educational institutions. *J. New Approach. Educ. Res.* 10. doi: 10.7821/naer.2021.7.640

Kiernan, N. A., Manches, A., and K, S. M. (2021). *The Role of Visuospatial Thinking in Students' Predictions of Molecular Geometry*. Available online at: pubs.rsc.org.

Latour, B. (1990). Technology is society made durable. *Sociol. Rev*. 38(1 Suppl.), 103–131. doi: 10.1111/j.1467-954X.1990.tb03350.x

Lodi, M., and Martini, S. (2021). Computational thinking, between papert and wing. *Sci. Educ*. 30, 883–908. doi: 10.1007/s11191-021-00202-5

Lye, S. Y., and Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for k-12? *Comput. Hum. Behav*. 41, 51–61. doi: 10.1016/j.chb.2014.09.012

Lyon, J. A., and Magana, A. (2020). Computational thinking in higher education: a review of the literature. *Comput. Appl. Eng. Educ*. 28, 1174–1189. doi: 10.1002/cae.22295

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., et al. (2014). "Computational thinking in k-9 education," in *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, Uppsala, 1–29. doi: 10.1145/2713609.2713610

Martín-Páez, T., Aguilera, D., Perales-Palacios, F. J., and Vílchez-González, J. M. (2019). What are we talking about when we talk about stem education? A review of literature. *Sci. Educ*. 103, 799–822. doi: 10.1002/sce.21522

Mirabueno, J. A. S., and Boyon, M. C. L. (2020). Senior high school academic progression in mathematics. *Int. J. Soc. Sci.* 5, 840–849. doi: 10.20319/pijss.2020.53.840849

National Academies of Sciences, Engineering, and Medicine (2018). *Graduate STEM Education for the 21st Century*. National Academies of Sciences, Engineering, and Medicine. Washington.

Newton, P., and Bristol, H. (2011). Psychometric success. *Spatial ability*.

Next Generation Science Standards (2013). *Next Generation Science Standards: For States, by States*, Vol. 1. Washington, DC: Next Generation Science Standards.

Novak, G. S. (1977). *Representations of Knowledge in a Program for Solving Physics Problems*. Available online at: ijcai.org.

Papert, S. (1980). *Children, Computers, and Powerful Ideas, Vol. 10*. Unitend Kingdom: Harvester Press, 978–983.

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: implications for future research. *Educ. Inform. Technol.* 22, 421–443. doi: 10.1007/s10639-016-9475-z

Peirce, C. S. (1868). Questions concerning certain faculties claimed for man. *J. Specul. Philos*. 2, 103–114.

Pluhár, Z., and Gellér, B. (2017). "International informatic challenge in Hungary," in *International Conference on Interactive Collaborative Learning* (Springer), 425–435. doi: 10.1007/978-3-319-73204-6_47

Pluhár, Z., Torma, H., and Törley, G. (2019). Hallgatói teljesítményértékelés az algoritmikus gondolkodás tükrében.

Shute, V. J., Sun, C., and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educ. Res. Rev*. 22, 142–158. doi: 10.1016/j.edurev.2017.09.003

Sochański, M. (2018). What is diagrammatic reasoning in mathematics? *Logic Logic. Philos*. 27, 567–581. doi: 10.12775/LLP.2018.001

Sowa, J. F. (2020). Natural logic is diagrammatic reasoning about mental models. *Proc. Comput. Sci*. 169, 31–45. doi: 10.1016/j.procs.2020.02.217

Stieff, M. (2011). When is a molecule three dimensional? A task-specific role for imagistic reasoning in advanced chemistry. *Sci. Educ*. 95, 310–336. doi: 10.1002/sce.20427

Stieff, M., Hegarty, M., and Dixon, B. (2010). "Alternative strategies for spatial reasoning with diagram," in *International Conference on Theory and Application of Diagrams* (Berlin: Springer), 115–127. doi: 10.1007/978-3-642-14600-8_13

Tang, X., Yin, Y., Lin, Q., and Hadad, R. (2018). "Making computational thinking evident: a validation study of a computational thinking test," in *AERA Online Paper Repository*.

Tikva, C., and Tambouris, E. (2021). Mapping computational thinking through programming in k-12 education: a conceptual model based on a systematic literature review. *Comput. Educ*. 162:104083. doi: 10.1016/j.compedu.2020.104083

Tóth, P., Horváth, K., and Kéri, K. (2021). Development level of engineering students' inductive thinking. *Acta Polytech. Hungar*. 18, 107–129. doi: 10.12700/APH.18.5.2021.5.8

Uttal, D. H., Meadow, N. G., Tipton, E., Hand, L. L., Alden, A. R., Warren, C., et al. (2013). The malleability of spatial skills: a meta-analysis of training studies. *Psychol. Bull*. 139:352. doi: 10.1037/a0028446

Voogt, J., Fisser, P., Good, J., Mishra, P., and Yadav, A. (2015). Computational thinking in compulsory education: towards an agenda for research and practice. *Educ. Inform. Technol*. 20, 715–728. doi: 10.1007/s10639-015-9412-6

Wan, X., Zhou, X., Ye, Z., Mortensen, C. K., and Bai, Z. (2020). "Smileycluster: supporting accessible machine learning in K-12 scientific discovery," in *Proceedings of the Interaction Design and Children Conference*, London, 23–35. doi: 10.1145/3392063.3394440

Wang, M.-T., and Degol, J. L. (2017). Gender gap in science, technology, engineering, and mathematics (stem): current knowledge, implications for practice, policy, and future directions. *Educ. Psychol. Rev*. 29, 119–140. doi: 10.1007/s10648-015-9355-x

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining computational thinking for mathematics and science classrooms. *J. Sci. Educ. Technol.* 25, 127–147. doi: 10.1007/s10956-015-9581-5

Wilkerson, M., and Fenwick, M. (2017). “Using mathematics and computational thinking,” in *Helping Students Make Sense of the World Using Next Generation Science and Engineering Practices,* eds C. V. Schwarz, C. Passmore, and B. J. Reiser (Arlington, VA: National Science Teachers’ Association Press), 1–19

Wing, J. M. (2006). Computational thinking. *Commun. ACM* 49, 33–35. doi: 10.1145/1118178.1118215

Wing, J. M. (2016). Computational thinking, 10 years later. *Micros. Res. Blog* 23.

Wolfram, S. (2002). *A New Kind of Science, Vol. 5*. Wolfram Media Champaign.

Xie, Y., Fang, M., and Shauman, K. (2015). Stem education. *Annu. Rev. Sociol.* 41, 331–357. doi: 10.1146/annurev-soc-071312-145659