Symbiotic Approach of Mathematical and Computational Thinking

Kristin Parve and Mart Laanpere(✉)

Tallinn University, Narva Road 25, 10120 Tallinn, Estonia
{kristin.parv,mart.laanpere}@tlu.ee

Abstract. Although CT is a rapidly expanding field of educational research, it is a relatively new concept in official national curricula. From the perspective of curriculum policy, CT is closest to two subjects taught in primary and secondary schools: computing/informatics and mathematics. Since informatics is not present as a separate subject in many countries, proponents of CT should find alternative routes for introducing this new body of knowledge in curricula. There are three main ways as to how it has been done in various countries: (A) adding CT into the existing informatics/computing curriculum, (B) integrating CT in the curriculum of some other subject – most likely, mathematics, and (C) introducing CT through cross-curriculum theme and interdisciplinary STEM/STEAM projects. This paper discusses the similarities and differences of computational and mathematical thinking that could potentially empower each other though meaningful integration in math lessons. Using the cases of Finland, Estonia, and Lithuania as examples, different approaches to integrating computational thinking into K-12 education will be contrasted and compared.

Keywords: Computational Thinking · Mathematical Thinking · informatics curriculum

1 Introduction

There is a well-known story about a philosophy professor who took an empty jar to his lecture. Standing in front of the class, he pulled some rocks out of his belongings and filled the jar with them. Everyone agreed that the jar was full. He then pulled out some pebbles and so filled the larger gaps between the rocks. The jar once again was full. Surprisingly, he then pulled out a box of sand and let it fill up the remaining spaces in the jar. The jar was full again. Even though this scene must have been unexpected, it proves the point that all the larger and smaller pieces you placed in the jar represent aspects of your life. It matters a lot if the jar is first filled with rocks or sand – if the sand, small unneccessary items are put in the jar first, you'll never have enough time for the big important things in your life.

This widely popular story has had its place as a reminder to set priorities in one's life and take time for important things. Yet, this paper won't try to argue or support that

Published by Springer Nature Switzerland AG 2023
T. Keane et al. (Eds.): WCCE 2022, IFIP AICT 685, pp. 184–195, 2023.
https://doi.org/10.1007/978-3-031-43393-1_18

matter. Computational thinking has been one of the most popular topics around computer science education in schools for more than 15 years, and many of the educators believe computational thinking to be one of those big rocks or at least some pebbles in the jar. But is it actually necessary or could we maybe solve the question of adding computational thinking teaching in K-12 differently?

First, we will provide an overview of two constructs – computational and mathematical thinking, then discuss the similarities and differences. Three cases from small European countries are used as examples to contrast and compare different approaches to introducing Computational Thinking in K-12 education. The metaphors of the rocks, pebbles and sand will help us to explain the challenges and solutions where identifying a suitable level of granularity of CT elements might avoid overwhelming the existing curricula.

2 Key Concepts

2.1 Computational Thinking

Computational Thinking (CT) as a term was coined by Wing [1], although it was not a totally new concept [2]. Its roots go back to the 1980s when Seymour Papert was first to mention it in his work about the LOGO programming language [3]. During the last 15 years, computational thinking has been at the centre of educational research as well as school innovations. Wing [1] described CT as a way of thinking that "*involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science*" [1, p. 33]. Over the years, there have been a number of attempts to operationalise the definition of CT, yet it has still remained a challenge. On a larger scale, proposed definitions of CT can be divided into two categories: ones that are more related to programming and computing concepts and the ones that see computational thinking as a broader problem solving skill [4]. First group of definitions value the computer science counterpart in them, adding skills like programming, debugging, computational models and solutions, and the use of software to them [2, 5–7]. The more universal definitions of computational thinking see it as a broader, transferable skill that could be addressed in the context of different subjects. Those definitions include CT components like problem identification, decomposition, algorithmic thinking, evaluation and generalisation as well as data practices [8–10].

Despite the ongoing discussion, studies mostly agree that integrating computational thinking in the school context will be beneficial. Weintrop *et al.* [6], who studied computational thinking in the context of mathematics and science, stated three main benefits of the integration. Firstly, it will develop the reciprocal relationship between learning mathematics, science and CT concepts; secondly, it will help to create a more accessible classroom environment; and finally, it keeps the mathematics and science classrooms aligned with current professional practices.

Integrating CT in the school context can be done on different levels. It can be 1) unplugged, meaning no computers are used, 2) include digital gadgets like programmable robots, or 3) be screen-based, with visual- or text-based programming environments [11]. Kotsopoulos *et al.* [12] proposed a four-phased Computational Thinking Pedagogical Framework (CTPF) that consisted of unplugged, tinkering, making and remixing stages. They stressed that a sequential approach may be helpful to understand the idea of computational thinking for novice learners.

From the perspective of policy makers, there are overall three different approaches to implement computational thinking to school education: 1) adding computational thinking across the curriculum in different subjects; 2) computational thinking is taught as a separate subject and 3) implementing computational thinking ideas into subjects that already exist in the school curricula [13].

2.2 Mathematical Thinking

Mathematical thinking does not refer solely to a specific subject, rather to a larger set of mathematical processes and operations that could generally be applied to any field [14]. As Polya [15] stated, the most important part of mathematical education is that it should teach students to think. Harel and Sower [16] stressed that mathematical understanding and thinking should be kept apart. Mathematical understanding refers to making sense of a particular mathematical problem, while mathematical thinking is something more universal, the key to understanding.

Habits of Mind were introduced by Cuoco, Goldenberg, and Mark in 1996 [17] for rethinking and reorganizing high school mathematics learning and teaching. They argued that high school mathematics teaching should provide students with real mathematical methods in order to help them think about mathematics in the way that mathematicians do. They suggested that those habits of mind could be divided into two: ones that are not limited to mathematics only, but cut across other disciplines, and the ones that are related to mathematics, so-called content specific habits. Mathematics related habits include skills like "thinking big and talking small" meaning generalizing and abstracting, thinking in ways of functions, using multiple points of view and mixing deduction and experiment. General habits include skills like finding patterns, experimenting, formulating written and oral descriptions, tinkering, inventing, using visualization, and conjecturing and guessing [17].

Mathematical literacy is defined in PISA as "*…an individual's capacity to formulate, employ and interpret mathematics in a variety of contexts. It includes reasoning mathematically and using mathematical concepts, procedures, facts and tools to describe, explain and predict phenomena. It assists individuals to recognise the role that mathematics plays in the world and to make the well-founded judgements and decisions needed by constructive, engaged and reflective citizens*" [18, p. 65]. This definition emphasises the need to use mathematics in context and to develop a deeper understanding of mathematics concepts because more and more daily-life situations and problems require some basic level of mathematical reasoning [19].

Mathematical literacy in the new PISA 2022 Mathematics Framework Draft [20] consists of the relationship between mathematical reasoning and problem solving skills (Fig. 1). Firstly, one has to be able to notice the mathematical nature in a (real life) situation and formulate it in the correct mathematical terms. The employment stage refers to the need to use the mathematical tools taught in school to solve the problem. Lastly, the outcome has to be evaluated in the context of the (real-life) problem. All those steps mentioned above are supported by mathematical reasoning skills.

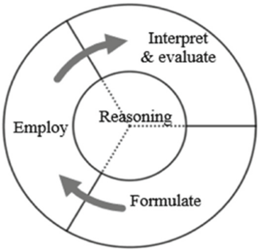

Fig. 1. Mathematical thinking process illustrated in PISA 2022 Mathematics Framework Draft [20]

2.3 Similarities between Computational and Mathematical Thinking

In the organising principle for mathematics curricula *Habits of Mind*, Couco and his colleagues stated that high school students should be helped to "learn and adopt some of the ways that mathematicians think about problems" [17]. About a decade later when teaching informatics was at a crossroad, Wing dreamed that everyone should be taught the basics of computational thinking – the ways and tools computer scientists use when solving problems [1]. Although those concepts differ from each other, they also share similar traits, not to mention the similarities in how the problems were addressed.

When trying to understand and compare the real heart of the two concepts, mathematical and computational thinking, one notices a similar aim of those two ways of thought processes. Going back to the 80's, Halmos [21] outlined that the existential reason for mathematics or as Stanic and Kilpatric [22] said – the real heart of it, is to solve problems. Winding 20 years ahead, similar ideas are used to describe the essence of computational thinking. Wing [1] said that "computational thinking involves solving problems" [1, p. 33], later computational thinking as an activity was seen as something "associated with, but not limited to, problem solving" [8]. Therefore, computational thinking not only shares ways with mathematical thinking [23], but also has similar overall aim. While going more in depth with the comparison of those concepts, several authors have made their statements about that.

Sneider and his colleagues [24] tried to describe the connection between mathematical and computational thinking using a Venn diagram (Fig. 2). Mathematical skills are related strictly to the subject, like counting, arithmetic, algebra, geometry and others. Computational thinking involves skills like simulation, algorithmic reasoning, gaming, programming and others. But these two ways of thinking also share a number of similar capabilities like problem solving, modeling, analyzing and interpreting data and statistics and probability.

Swedish researchers Bråting and Kilhamn [25] have studied the connections between algebraic and computational thinking in the context of changes in the local curriculum. Teaching mathematics includes fostering algebraic thinking. They stated that at least on the theoretical level, both algebraic and computational thinking value the process of problem-solving more than the result, although the domains themselves are rather different from each other [25, 26].

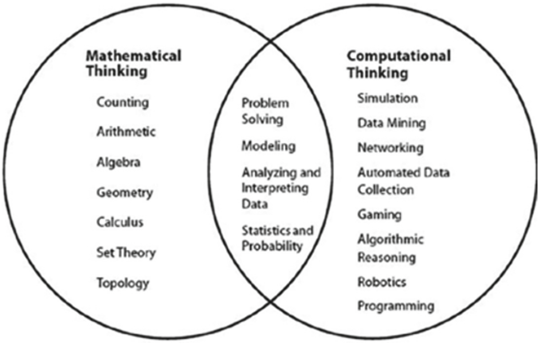

Fig. 2. The overlap of Computational and Mathematical thinking illustrated by a Venn diagramm [24]

Pei *et al.* [27] also see the overlap between computational and mathematical thinking. They describe how computational thinking and mathematical habits of mind are strongly related and mutually supportive. Therefore, adding a computational aspect to the mathematics classroom will create a larger and more meaningful mathematics learning experience. Weintrop and his colleagues [6] also saw computational thinking as a beneficial component of mathematics classrooms. They formulated a taxonomy where the integration of CT can be divided into four categories: data practices, modeling and simulation practices, computational problem-solving practices and systems-thinking practices [6]. More recent research on integrating computational thinking into mathematics education was conducted by Kallia and her colleagues [28], who described computational thinking as an'umbrella' concept that does not depend on the context, therefore, is adaptable to different situations. It was agreed that while talking about computational thinking in the context of mathematics education, three main aspects should be considered: *problem solving* as a fundamental part of mathematics education in which computational thinking can be taught, *cognitive process* meaning the different thinking processes that mathematical and computational thinking share, and *transposition* or the ability to phrase the solution of a mathematical problem [28].

The overlap between those concepts has been noticed and acted on already at a larger scale. The need to "encompass the synergistic and reciprocal relationship between mathematical thinking and computational thinking" is clearly stated in the draft of the PISA 2022 Mathematics framework [20, p.7]. Enriched with many examples, the main emphasis is on how those two concepts complement each other, giving out endless possibilities to deepen the understanding of mathematics while interacting more effectively with new technologies. More precisely, students should be able to show their computational thinking skills in the three parts of mathematical literacy described above [20]. In addition to the named benefits of integrating CT into mathematics classes, Stephens and Kadijevich [29] also described examples of integrating CT into mathematics classes in several countries. Since the integration of CT into mathematics is complex, countries have chosen different levels of integration (cross-curricular integration vs a separate subject) or fundamentally different approaches (more of a gradual introduction vs having a formal subject for everyone).

3 CT-Related Curriculum Policies in Three Countries

Computational thinking has been added to national curricula in various countries. While some of the countries have chosen to teach computational thinking as a separate discipline, other countries have divided it into several subjects as a cross-curricular approach or added it into the curricula of the subjects students are already familiar with [13]. An overview of the three different approaches in Lithuania, Finland and Estonia is introduced in the next sections.

3.1 Lithuania

The Lithuanian education system is free of charge and is compulsory until the student is 16 years of age. School education consists of mainly three parts: primary, basic and secondary education, all together 12 years. In addition to that, youth schools are an alternative to basic education that offer pre-vocational training during the studies. Secondary education curricula consist of compulsory and optional modules and this level of education can be acquired in either gymnasiums, pre-gymnasiums, full or short secondary, vocational or other schools [30].

Information technology (IT) courses are part of compulsory education in Lithuania. At primary level, informatics is taught as a part of other subjects. In the lower secondary school (grade 5–10) IT is taught for one hour a week, emphasising students to see the integrative nature of IT and how it benefits their overall study process. While IT is compulsory at the lower secondary level, it is an elective course at upper secondary school (grade 11–12). The level of IT studies in schools at both lower and upper secondary school level depends heavily on the level of skills and knowledge of the teachers [30–32].

Teaching and learning computational thinking (lithuanian "*informatinio mąstymo*") concepts are also part of the compulsory IT course. The subject includes five areas of knowledge: information; digital technologies; algorithms and programming; virtual communication; security, ethics and legal principles. The studies in upper secondary school include topics related to electronic publishing, database design and management, and programming [30, 31]. In 2023 a new curriculum will come into effect that will change the matter of IT education. More of a cross-curricular approach at the primary school level is being introduced, including studying CT components in other subjects [32]. The new curriculum, that is most probably going to be accepted in the summer of 2022, continues with a compulsory computer science education as a separate subject from the 5th grade, setting teaching computational thinking skills as one of the main learning outcomes in the basic school as well as secondary school level.

3.2 Finland

The Finnish educational system also consists of basic education (grades 1–9) and upper secondary education. Upper secondary education can be divided into two: more general and rather academically oriented education, and vocational education which aims is to prepare students for direct employment or further studies in the polytechnics [33]. Up until recently, only basic education was compulsory. Continuing studies in the upper secondary or vocational level was rather popular with more than 90% of young people electing to do so, but it was not officially compulsory. Starting from August 2021, the compulsory education was extended, now students have to complete an upper secondary qualification (either from the general or vocational education) or attend school until 18 years of age [34, 35].

	Years 1–2	Years 3–6	Years 7–9
Digital competence	using digital media, technological fluency	impact of technology, tech-integration	
Math	step-by-step instructions	visual programming	algorithmic thinking, good computing conventions
Crafts		robots, automation	embedded systems, own artifacts

Fig. 3. Computer Science related topics in Finnish basic school (Niemelä *et al.*, 2017)

Finland has approached teaching CT as a more cross-curricular activity. The division of computer science related topics in basic school level can be seen above (Fig. 3). Starting from 2016, they were one of the first countries in the European Union to set "algorithmic thinking" and programming as a mandatory part of the curriculum starting from the 1st grade [31]. In the basic education, *i.e.*grades 1–9, learning and teaching CT is integrated in different subjects from arts to environmental studies, but mainly in the mathematics lessons. Algorithmic thinking is stated as one of the 20 objectives for Maths. Later on in upper secondary education, different courses related to programming, computer science and CT are offered [32]. The introduction to CT skills is a continuous process where in every grade some additional skills are taught. It starts with learning to give step-by-step instructions in the first two school years that is followed by using visual programming tools. During the years in the basic education, they are gradually introduced to more and more complex concepts. In whatever subject the programming tasks are used, they always serve a higher purpose for the learning process and are also aligned with transversal competences in the national core curriculum [31].

3.3 Estonia

The Estonian education system also consists of nine years of basic education that can be followed by general upper secondary or vocational education. Education is provided free of charge and the studies are meant to support the lifelong learning process [36, 37].

The informatics related subject has been on and off the national curriculum in Estonia. From the mid 90s to the early 2000s, it was part of the national curriculum as an elective course named informatics. In the first decade of the 2000s, informatics was not a separate subject in the curriculum, but students had to be introduced to compulsory information and communications technology (ICT) skills by the end of basic school as a part of other subjects. From 2011, informatics or subjects related to the concept are back as elective subjects in basic as well as upper secondary education [38].

More of an holistic approach of informatics education throughout the twelve years of school has been offered to be in the new updated national curricula and is described in the figure above (Fig. 4). New versions of the syllabuses were sent to the government in January 2022 and hopefully will be in action from 2023. In the basic education curriculum, the central concepts that the curriculum is based on are 1) design thinking for

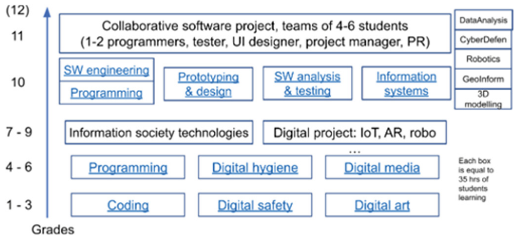

Fig. 4. Holistic view of K-12 informatics curriculum (Niemelä *et al.*, 2021)

creative and collaborative learning and 2) computational thinking for a more thorough way of solving real world problems. As seen in Fig. 4, during grades 1–6, students are introduced to basic concepts of different fields, whereas in grades 7–9 the knowledge will be brought together to solve real-life problems in a collaborative way. During the upper secondary school level, the informatics related subjects stress the importance of developing practical ICT, creative thinking and collaboration skills. Therefore, the outcomes of the elective courses can be put into action during the collaborative software project. Students in basic and in upper secondary school level have to conduct an empirical research or practical project to graduate, therefore adding digital project in the lower secondary school and collaborative software project in the upper secondary school level gives students more possibilities to pass the mandatory part of curriculum in more collaborative and 21st century self-directed learning ways.

4 Discussion

As shown above, there is no one way to integrate computational thinking ideas into school curriculum. Several examples of integrating CT into mathematics have been introduced by Stephens and Kadijevich [29], but in general three main ways of integration have been identified [13]: (1) CT as a cross-curricular theme; (2) CT as a separate subject; (3) selected CT ideas integrated in a few chosen subjects. As our study showed, often some kind of mix of these approaches is used. Lithuania has a long history of computer science education [38], therefore it is no surprise that the main way to bring computational thinking ideas to K-12 education is through a separate computer science curriculum, but pilot studies have already been conducted to introduce more of a cross-curriculum approach in primary schools in the future [32]. Finland has also worked on a mix of cross-curricular integration and single-subject model, although the main responsibility for teaching CT lies on mathematics teachers [13, 32]. Estonia falls somewhere between these two approaches. Computational thinking is mentioned in the "not yet official but already in use" documentation for the upcoming elective informatics subject both in basic and secondary school. Although elective courses have many benefits, such as having a wide range of topics to cover and an unlimited number of lessons that the school themself can agree on, it also brings up some problems. Dagiene and Stupuriene [30] mentioned that although informatics is a compulsory subject in Lithuania, the level of teaching depends heavily on the knowledge and skills of the teachers. Estonia is facing a similar problem – although teaching ICT skills is set as a priority in Estonia, there is a lack of competent IT teachers [39], and the actual level of IT education varies between schools. It has been calculated that only a quarter of teachers actually have qualifications to teach IT. Although having computational thinking skills integrated into the next curriculum is already a step ahead, it might still not be enough.

Now, imagine the jar full of rocks, pebbles and sand again. School curricula are mostly full and those already existing subjects like mathematics can be seen as rocks in the jar, filling biggest parts of the curricula. This also means that adding new courses or subjects automatically refers to the need to discard some of the existing learning outcomes. Therefore, the integration of CT or anything else new has to be thought through, adding them in smaller pieces, or as pebbles and sand to the jar.

Some countries see teaching computational thinking as a quite a pebble in the jar of curriculum – it has its place as a separate subject like Lithuania does. As shown before, they soon will have an additional way of cross-curricular approach at the primary level. This could be seen as some sand in the existing slots in the jar. Finland has taken another path, adding computational thinking as smaller pieces mainly in the context of mathematics to the curriculum. While trying to visualise it, mathematics could be seen as a bigger rock and the computational thinking counterpart as a sand to fill in the gaps around it. Estonian approach is more difficult to picture. Is computational thinking added as a sand to the learning outcomes of informatics lesson that could easily fill the voids or is it just adding additional pebbles to the jar that are already falling out of the jar?

As discussed above, computational thinking shares many similarities with mathematical thinking, sharing similar sub-skills [6, 24] and a ground-base [1, 17, 24, 25]. Using the same metaphor, the integration of CT into mathematics classes could also be illustrated with rocks, pebbles and sand. The smallest granularity of CT elements in the mathematics classroom could be computational tasks meant only for the gifted students as a way to enrich the school-level mathematics or prepare them for the mathematics competitions. The next level of granularity, pebbles, could address some specific learning outcomes of mathematics curricula and engage all students in widening their understanding of abstract mathematical concepts in computational context that might be more interesting, closer to real life. The largest granularity level of CT elements would bring in another "rock" in the jar by introducing CT as a separate course – if the existing curriculum allows it. On the other hand, the "CT rock" could be also introduced as a cross-curricular theme, in the form of interdisciplinary project-based learning as it is done in Finnish case of phenomenon-based learning.

5 Conclusions

This paper considers the similarities and differences of computational and mathematical thinking, two concepts that could potentially empower each other in school curricula. The three main ways to introduce how computational thinking has been integrated in the K-12 education was illustrated by the examples of three close countries: Finland, Estonia and Lithuania. While introducing computational thinking in K-12 education is seen as important and beneficial by many authors, making changes to the existing curriculum is never easy. Designing school curricula is a highly contextualized and politicized process, which is why every country has to find their own way for introducing computational thinking in schools. We have provided some arguments for adding computational thinking in everyday school-life as smaller pieces in the context of similar concepts like mathematics could be a solution for some countries, to avoid overwhelming the existing curricula.

References

1. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
2. Grover, S., Pea, R.: Computational thinking in K–12: a review of the state of the field. Educ. Res. **42**(1), 38–43 (2013)
3. Papert, S.: "Mindstorms" Children. Computers and Powerful Ideas.Basic books, Inc., Publishers/New York (1980)
4. Tang, X., Yin, Y., Lin, Q., Hadad, R., Zhai, X.: Assessing computational thinking: a systematic review of empirical studies. Comput. Educ. **148**, 103798 (2020)
5. Brennan, K., Resnick, M.: New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada, pp. 1–25 (2012)
6. Weintrop, D., et al.: Defining computational thinking for mathematics and science classrooms. J. Sci. Educ. Technol. **25**(1), 127–147 (2016)
7. Denner, J., Werner, L., Ortiz, E.: Computer games created by middle school girls: can they be used to measure understanding of computer science concepts? Comput. Educ. **58**(1), 240–249 (2012)
8. Selby, C., Woollard, J.: Computational Thinking: the Developing Definition. University of Southampton (E-prints), UK (2013)
9. ISTE, CSTA. Operational Definition of Computational Thinking for K-12 Education (2011)
10. Yadav, A., Hong, H., Stephenson, C.: Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. TechTrends **60**(6), 565–568 (2016)
11. Gadanidis, G., Clements, E., Yiu, C.: Group theory, computational thinking, and young mathematicians. Math. Think. Learn. **20**(1), 32–53 (2018)
12. Kotsopoulos, D., et al.: A pedagogical framework for computational thinking. Dig. Experiences Math. Educ. **3**(2), 154–171 (2017)
13. Bocconi, S., Chioccariello, A., Earp, J.: The Nordic approach to introducing Computational Thinking and programming in compulsory education. Report prepared for the Nordic@ BETT2018 Steering Group, pp. 397–400. National Research Council of Italy, Institute for Educational Technology (CNR-ITD), Palermo, Italy (2018)
14. Burton, L.: Mathematical thinking: The struggle for meaning. J. Res. Math. Educ. **15**(1), 35–49 (1984)
15. Polya, G.: Mathematical Discovery: On Understanding, Learning and Teaching Problem Solving, 2 vol. combined, 1981st edn. John Wiley & Sons, New York (1965)
16. Harel, G., Sowder, L.: Advanced mathematical-thinking at any age: its nature and its development. Math. Think. Learn. **7**(1), 27–50 (2005)
17. Cuoco, A., Goldenberg, E.P., Mark, J.: Habits of mind: an organizing principle for mathematics curricula. J. Math. Behav. **15**(4), 375–402 (1996)
18. OECD: PISA 2015 Assessment and Analytical Framework: Science, Reading, Mathematic, Financial Literacy and Collaborative Problem Solving, revised edition. PISA, OECD Publishing, Paris (2017)

19. OECD: PISA 2018 Assessment and Analytical Framework. PISA, OECD Publishing, Paris (2019)
20. PISA, OECD: Mathematics Framework (Draft). Retrieved from PISA (2022). https://pisa2022-maths.oecd.org/files/PISA%202022%20Mathematics%20Framework%20Draft.pdf. Last accessed 22 May 2023
21. Halmos, P.R.: The heart of mathematics. Am. Math. Mon. **87**(7), 519–524 (1980)
22. Stanic, G., Kilpatrick, J.: Historical perspectives on problem solving in the mathematics curriculum. In: Charles, R., Silver, E. (eds.) The teaching and assessing of mathematical problem solving, pp. 1–22. National Council of Teachers of Mathematics, Reston, VA (1988)
23. Wing, J.M.: Computational thinking and thinking about computing. Phil. Trans. R. Soc. A: Math. Phys. Eng. Sci. **366**(1881), 3717–3725 (2008)
24. Sneider, C., Stephenson, C., Schafer, B., Flick, L.: Exploring the science framework and NGSS: computational thinking in the science classroom. Sci. Scope **38**(3), 10 (2014)
25. Bråting, K., Kilhamn, C.: Exploring the intersection of algebraic and computational thinking. Math. Think. Learn. **23**(2), 170–185 (2021)
26. Malara, N.A., Navarra, G.: New words and concepts for early algebra teaching: sharing with teachers epistemological issues in early algebra to develop students' early algebraic thinking. In: Kieran, C. (ed.) Teaching and Learning Algebraic Thinking with 5-to 12-Year-Olds, pp. 51–77. Springer, Cham (2018)
27. Pei, C., Weintrop, D., Wilensky, U.: Cultivating computational thinking practices and mathematical habits of mind in lattice land. Math. Think. Learn. **20**(1), 75–89 (2018)
28. Kallia, M., van Borkulo, S.P., Drijvers, P., Barendsen, E., Tolboom, J.: Characterising computational thinking in mathematics education: a literature-informed Delphi study. Res. Math. Educ. **23**(2), 159–187 (2021)
29. Stephens, M., Kadijevich, D.M.: Computational/algorithmic thinking. In: Lerman, S: (ed.) Encyclopedia of Mathematics Education, pp. 117–123. Springer, Cham (2020)
30. Dagiene, V., Stupuriene, G.: Bebras – a sustainable community building model for the concept based learning of informatics and computational thinking. Inform. Educ. **15**(1), 25–44 (2016)
31. Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K.: Developing computational thinking in compulsory education-Implications for policy and practice (No. JRC104188). Joint Research Centre, Seville (2016)
32. Bocconi, S., et al.: Reviewing Computational Thinking in Compulsory Education, JRC128347. In: Inamorato Dos Santos, A., Cachia, R., Giannoutsou, N., Punie, Y. (eds.). Publications Office of the European Union, Luxembourg (2022)
33. Kupiainen, S., Hautamäki, J., Karjalainen, T.: The Finnish education system and PISA. opetus- ja kulttuuriministeriö (2009)
34. Maaranen, K., Stenberg, K.: Teacher effectiveness in finland: effectiveness in finnish schools. In: Grant, L.W., Stronge, J.H., Xu, X. (eds.) International Beliefs and Practices That Characterize Teacher Effectiveness, pp. 125–147. IGI Global, Hershey, PA (2021)
35. Eurydice: Finland: Compulsory education extended until the age of 18. https://eacea.ec.europa.eu/national-policies/eurydice/content/finland-compulsory-education-extended-until-age-18_en. Last accessed 26 Mar 2022
36. Estonian Education System: https://www.educationestonia.org/about-education-system/, Last accessed 26 Mar 2022
37. Preschool, basic and secondary education: https://www.hm.ee/en/activities/pre-school-basic-and-secondary-education. Last accessed 26 Mar 2022
38. Niemelä, P., Pears, A., Dagienė, V., Laanpere, M.: Computational thinking–forces shaping curriculum and policy in Finland, Sweden and the Baltic Countries. In: Passey, D., Leahy, D., Williams, L., Holvikivi, J., Ruohonen, M. (eds.) Digital Transformation of Education and Learning – Past, Present and Future. OCCE 2021. IFIP Advances in Information and

Communication Technology, vol. 642, pp. 131–143. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-97986-7_11

39. Haaristo, H.-S., et al.: Elukestva õppe strateegia vahehindamine. Tallinn: Poliitikauuringute Keskus Praxis, Rakendusuuringute Keskus CentAR (2019)