Computational Thinking for Education Online

Project number: 2021-1-PL01-KA220-SCH-000024345

Computational Thinking
for education on-line

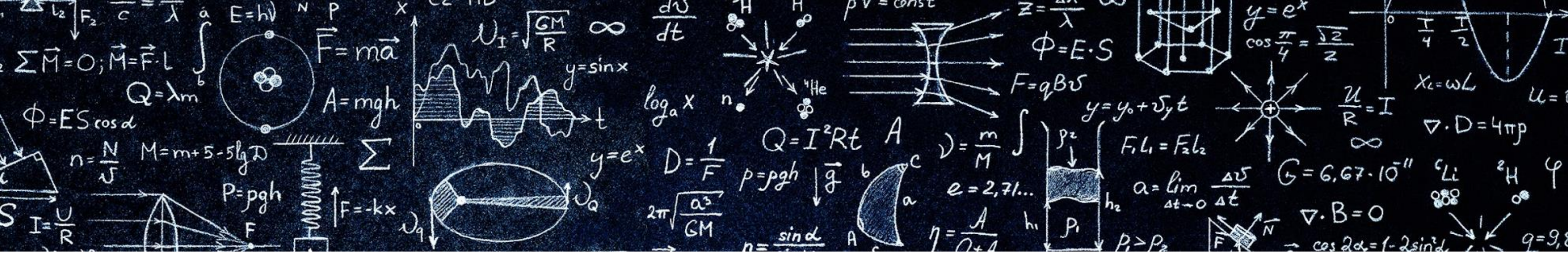
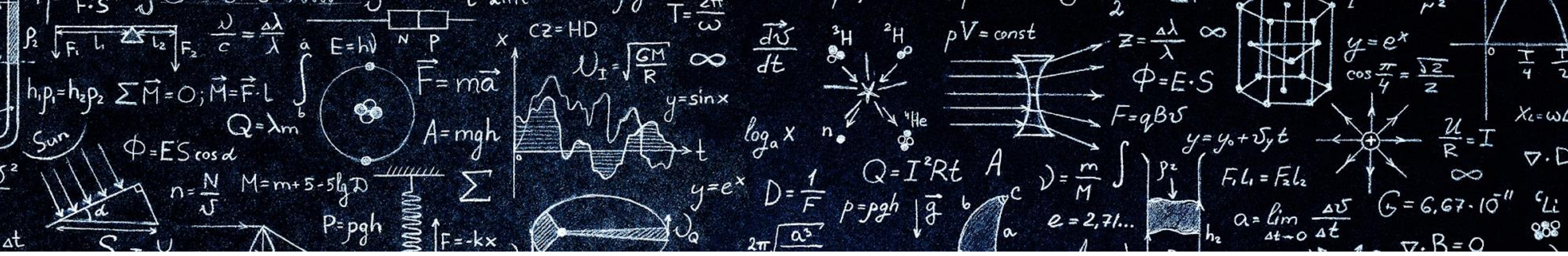


Table of contents



Publication title: Gamification For Intercultural Education

Date and year of publication: July, 2022

Book template attribution: Docs And Slides Website, https://docsandslides.com

More information about the project at http://edu-gamification.eu/

Organisations involved in the project:

Krakowskie Centrum Zarządzania i Administracji Sp. Zo.o.

Mediacreativa 2020 S.L.

2nd Peiramatiko Lykeio Athinas

32 Secondary Language School"St. Kl. Ohridski"

Authors and reviewers:

- Anna Raźny, Krzysztof Karkowski, Dorota Karkowska
- Leire Monterrubio, Jaione Santos, Johanna Helmes
- Anastasia Mylona
- Veneta Velkova
- Ioannis Dimos, Athanasios Dimos

This project is co-funded by the European Union

This publication has been produced with the financial support of the European Commission under the Erasmus+ program. This publication reflects the views only of the authors, and the European Commission is not responsible for the content contained therein.

1.1 Role of digital literacy in modern education

Modern information and communication technologies occupy an increasingly important role in the world around us and participate directly or indirectly in all areas of our lives. They are very attractive because, in whatever context they are used, they offer a new orientation in situations, a new way of thinking and a new view of the future. They are based on building a new global economic and social system – see Dufva (2019); Yates and Rice (Ed), (2020); Bikalenko, Vekua, Telegina, and Khabdaev (2021); Hilbert (2022); Nadoleanu (2022); Lynn (2022).

The idea of using computer technology in education is not new. For years, all experts in the field have suggested that when used intelligently, they can greatly improve learning outcomes and improve the quality of the process.

Due to globalisation and increasing international labour market connected mobility, technology is a priority for national education systems (Cuban, Kirkpatrick and Peck, 2001; Higgins and Packard, 2004). Ottestade and Quale (2009) find a positive relationship between ICT implementation and learning outcomes. Semenov (2005), Vrasidas and Glass (2005) and Roblyer (2006) provide evidence that integrating technology into curricula improves the learning process, increases student outcomes, and provides better opportunities for teaching and learning. The authors state that by effectively integrating ICT into curricula, educational institutions can provide higher quality services and strategies.

The dynamic development of technology in the 21st century and the emergence of a variety of digital learning resources have been accompanied by a rethinking of the nature of teaching and learning. They lead to a new understanding of the role of creating effective teaching materials, to new approaches to teaching teachers how to integrate these resources, and to rethink the relationship between the activities in which students are involved in class and their future implementation. All these mainstream trends provoke the development of a meta vision for education, significantly improved by integrating and using digital resources for teaching and learning, education that is more closely aligned with the requirements for the future literacy and realisation of young people, education that can create thinkers, able to work in a team and communicative personalities who will be part of an increasingly competitive global economy.

In order to integrate information technologies successfully in the educational process and to ensure their implementation in class, a process of careful planning and design is needed - planning training and teacher support, developing the necessary new digital learning materials, coordinating the elements between different levels of the education system, regular monitoring and analysis.

Coordinated national activities on the factors related to the successful integration of digital technologies and developing Computational Thinking (CT) (explained in the next section 1.2.) in education and training, as well as the need for comprehensive leadership in educational innovation, are important. Technology-based teaching and learning can lead to many changes in school, which will inevitably create the demand for new ways of planning the learning process. Dudeney (2010) noted that national ICT policies can serve several key functions. They provide a rationale, a set of goals and a vision for how education systems work if ICTs are integrated into the teaching and learning process, and are useful to students, teachers, parents and the population of a country.

Technology can be used to open the classroom to the wider world. Teachers can benefit from it by expanding and deepening their professional skills and diversifying their teaching style. Blackmore, Hardcastle, Bamblett and Owens (2003) argue that new technologies contribute to both teaching and learning by creating independent and motivated learners, encouraging the use of a variety of teaching methods. Roblyer and Edwards (2000) believe that ICT in teaching saves time and makes learning more flexible. Using computer technology, teachers can simulate the environment of physical characteristics of the objects studied (Capron & Johnson, 2004).

However, the integration of ICT in the educational process may face a number of barriers. Ertmer (1999) mentions two types of barriers that educational institutions face: external barriers related to access to and building an ICT environment and teachers' skills in using technology (digital skills). The second type of obstacles are defined as internal barriers and are related to the qualification of the teacher and their attitudes and beliefs. When external barriers are overcome, the decision and personal responsibility for whether and how to use ICT in the classroom depends on the beliefs, attitudes, motivation and competencies of the teacher.

The benefits of applying ICT and CT in education are many. It is important to acknowledge that people learn better from combining visuals with text and sound than through using either process alone, provided the design of learning resources follow certain multimedia principles (Mayer & Moreno, 2003). This set of seven principles related to multimedia and modality is based on the work of Richard Mayer, Roxanne Moreno, and other prominent researchers (Chan & Black, 2006; Ginns, 2005; Mayer, 2001; Mayer & Moreno, 2003).

1. **Multimedia Principle:** Student retention is improved through a combination of words (verbal or text) and visuals, instead of using words alone, provided it doesn’t introduce redundancy of content.
2. **Spatial Contiguity Principle:** Students learn better, when the text and visuals are physically integrated instead of being separated.

3. **Temporal Contiguity Principle:** Students learn better, when corresponding text and visuals are temporally synchronised instead of separated in time.
4. **Split-Attention Principle:** Students learn better when extraneous words, pictures, and sounds are excluded instead of included.
5. **Modality Principle:** Students learn better when text is presented auditorily as speech instead of as on-screen text.
6. **Individual Differences Principle:** Design effects from these principles are higher for low-knowledge learners than for high-knowledge learners, and that they are higher for high-spatial learners than for low-spatial learners.
7. **Direct Manipulation Principle:** As the complexity of the materials increases the impact of direct manipulation of the learning materials (animation, pacing) on transfer also increases.

As a result, students engaged in learning that comes with high-quality multimodal designs outperform, on the average, students who learn using traditional approaches with single modes.

It is worth stating that the digitalization of the educational process goes not only with the emergence of new methods and organisational forms of education but leads to a significant change in the role of teachers, and a requirement for continuous self-improvement and training. This tendency insists that teachers have a number of new skills for managing class work here to mention among them the ability to produce new knowledge, develop new ways to use information and communication technologies in enriching the learning environment, and keep a close eye on students' digital literacy development.

To summarise, professional development of teachers is the starting point in the algorithm of improving the education system and integrating new technologies into the educational process.

Digital literacy should be tightly developed and improved through the parallel application of *Inquiry based learning (IBL)*. Through the latter, students are engaged in real and authentic research. IBL aims to achieve a more realistic concept of scientific knowledge and provide a more motivating and learning-oriented environment.

Research-based learning (RBL) as a synonym of IBL is gaining increasing support (Polman, 1998). To be effective, research-based learning must include the basic steps of conducting research and understanding the way scientists make scientific discoveries and at the same time develop CT with its 4 cardinal pillars – decomposition, pattern recognition, abstraction, algorithm thinking (further explained in section 1.3.) Research-based training focuses on the importance of "mastering" processes in the natural sciences, as the formulation of empirically studied questions supported by claims and evidence" (Polman, 1998: 3). The effective use of this method engages students in independent research, provokes them to think deeply and scientifically realise the connection between evidence and theory, and develop CT. It is not so much the outcome of the study that is significant, but the process of the study and it is therefore important to provide time for discussion and encourage students to present their ideas (Watson, 2000).

1.2. What is Computational Thinking?

"Computational Thinking" (CT) as a concept has become popular in recent years; especially after being defined by Wing in 2006. Until recently, computing was considered a limited skill possessed only by computer scientists, engineers, mathematicians, and those from similar disciplines. However, nowadays almost everybody, irrespective of age, is expected to have some basic computing skills in parallel with the developments in technology.

In a seminal article published in 2006, Jeanette Wing described computational thinking (CT) as a way of "solving problems, designing systems, and understanding human behaviour by drawing on the concepts fundamental to computer science." She noted that computational thinking involves some familiar concepts, like problem decomposition, data representation, and modelling, also less familiar ideas, like binary search, recursion, and parallelization. She also argued, "Computational thinking is a fundamental skill for everyone, not just for computer scientists".

CT is a conception that has been growing for the last few years, used for the first time by Papert in 1996. In his paper, Papert didn't comprehensively define CT, but in 2006, Jeannette Wing presented CT and defined it as a skill for everyone, not just for computer scientists. This description seems general and abstract in terms of integrating CT into classes and how to observe scholars' CT ability (Zhenrong, Wenming, and Rongsheng, 2009). Companies like Google and Microsoft supported this idea, and several programs and projects have emerged to incorporate CT across different curricula. On the other hand, the International Society for Technology in Education (ISTE) and Computer Science Teacher Association (CSTA) published a functional description about CT; CT is a problem-based process that includes (not limited to) the following characteristics:

- Formulating problems in such a way that allows us to use a computer and other similar tools to assist us in solving them;
- Logically organising and analysing data;
- Representing data through abstractions like models and simulations;
- Automating solutions using algorithmic thinking (a sequence of ordered steps);
- Identifying, analysing, and implementing possible solutions with the goal of achieving the foremost efficient and effective combination of steps and resources;
- Generalising and transferring this problem-based solving process to a huge diversity of problems (CSTA and ISTE, 2011).

In addition to this definition, Mannila and her colleagues (2014) affirmed that CT could be a term covering a group of conceptions and thinking processes from computing that help in formulating problems and their solutions in several subject areas. Similarly, Riley and Hunt (2014) approached the cognitive strategies of thinking, as "the best way to characterise Computational Thinking is as the way that computer scientists think, the way in which they reason" (p.4).

> Wing views CT as process and logic rather than programming and coding. Two main aspects of CT based on Wing's perspective are:
>
> – thinking at multiple levels of abstraction during problem-solving processes;
> – communicating ideas and knowledge in computational terms during problem-solving and hands-on inquiry.

Sysł o and Kwiatkowska (2013) also emphasise that CT is a set of thinking skills that may not lead to computer programming. According to them, CT should "focus on the principles of computing rather than on computer programming skills" (p. 50). When examining the definitions within the literature, most of them linger over problem-solving, understanding problems, and formulating problems (Wing, 2006; Zhenrong, Wenming and Rongsheng, 2009; Liu and He, 2014; Barr, Harrison and Conery, 2011). To improve this ability, words like algorithm and precondition must be a neighbourhood of everyone's vocabulary (Zagami, 2013). Following the categorization proposed by Brennan and Resnick (2012), Lye and Koh (2014) suggested aspects of CT on concepts, practices and perspectives. With the dimension of computational terminology, they mentioned conceptions used by IT experts like variables. For the second aspect, CT application, they had in mind problem-solving practices encountered in computer programming processes such as loops and recursion. As the last dimension, namely computational perspectives, the authors mentioned the aforementioned (section 1.1.) self-improvement and self-realisation of scholars in terms of the technological world around them like abstracting, questioning, and debugging.

Computational thinking (CT) is a fundamental skill that is equivalent to reading, writing and arithmetic skills (National Research Council, 2005) CT involves problem-solving, systems design, and “understanding human behaviours” (Wing, 2006). CT is also a metacognitive process that consists of sub-skills and dispositions, which provide students the ability to analyse scientific patterns and model complex phenomena (C. P. Dwyer, M. J. Hogan, and I. Stewart, 2013). Thus, CT is the "third pillar” of scientific practice National Research Council, 2005) and plays a critical role in scientific inquiry and problem-solving (PCAST), (2010).

Computational thinking involves:

1. ***Algorithmic Thinking***

Algorithmic Thinking is the capability to create an ordered sequence of steps with the aim of solving a problem (further explained in the next section).

2. ***Evaluating / Debugging***

This is the ability to check if a prototype is working as planned, and if not, the ability to identify what needs to be improved. This is also the process a developer goes through to find errors in the programme and fix them

3. ***Abstraction***

Abstraction is the capability to explain a problem or a resolution by removing insignificant details.

Benefits of developing Computational Thinking in school:

- being able to conceptualise an idea.

A Process for Developing Computational Thinking Skills.

4. Planning

The students should spend some time imagining different solutions to the problem, and then make an in depth plan for executing one among their ideas. They will define the steps that they will need to go through in order to reach the solution. When they identify parts of the task that they may have seen before, they will develop a skill called 'Generalisation'.

The aim is to make students

- be able to make a list of actions to programme

- be able to identify parts of existing programmes that they could use

- be able to reuse parts of programmes

Students are then tasked with creating the ultimate version of their solution.

When the students code their ideas, they develop their Algorithmic Thinking skills.

5. Modifying

The students evaluate their solutions according to whether or not their program and model meet the success criteria. Using their evaluation skills, they will identify if they need to change, fix, debug or improve a part of their program.

The purpose is to make students:

- making iterations of their programme

- fixing problems in their programme

- able to assess whether the solution is linked to the problem

6. Communicating

Each student will present the final version of their solution to the class, explaining how their solution meets the success criteria. By explaining their solution with the proper level of detail, they are going to develop their Abstraction and communication skills.

The aim is to make students:

- explaining the most important part of their solution

- giving enough detail to enhance comprehension

- clarifying how their solution meets the criteria for success

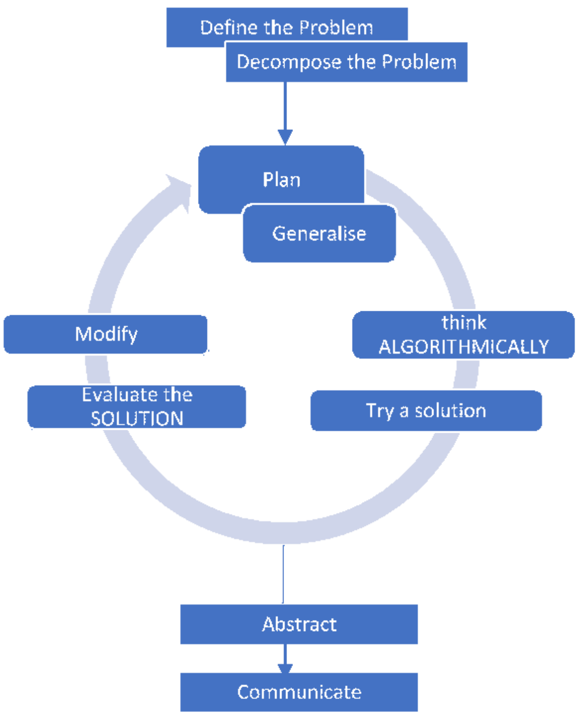


Chart.1. CT process

1.3 Key concepts of Computational Thinking

Computational Thinking can be further defined as a set of skills for problem-solving based on computer techniques. Computational thinking is required everywhere and goes to be a key to success in most careers, not just for a scientist, except for many professionals, like doctors, lawyers, teachers or farmers. To resolve a problem, it is a good idea to make a plan using some of the computer science techniques like: breaking down a complex problem into smaller parts that are more controllable and easier to understand, or solving - decomposition; identifying similarities between and within problems and other practices - pattern recognition; focusing only on important information, and drawing out specific differences to make one solution work for many different problems: abstraction; developing a step-by-step solution to the problem: algorithms. Everyone, no matter his or her area of expertise, task or age, often employs this plan. It is important to practice and develop these techniques very early. In recent years, we have to ascertain the proliferation of various projects with the precise objective of encouraging the study of Computational thinking. The projects of massification of computational thinking and coding are now beginning to be implemented in several education systems.

Computational thinking transforms educators into innovators, to find ways to solve a problem, to organise and plan the resolution of a task. CT teaches us and gives us the courage, the methods and techniques to solve complex problems.

In recent years, part of the research related to the formation and development of the necessary competencies of teachers for the application of ICT is based on the conceptual framework of Technological Pedagogical and Content Knowledge TPCK (TPACK). Koehler and Mishra (Koehler & Mishra 2005, 2007; Mishra & Koehler, 2006) proposed it as an extension of the Pedagogical Content Knowledge model developed by Shulman (Shulman, 1986).

Mishra and Koehler propose this conceptual framework to describe the complex system of knowledge, skills and competencies, which must have a teacher who applies modern information technology in their work. Through it, they outline the complex "system of connections, interactions, mutual determination and limitations between the content of the subject, pedagogy and technology." In this model, knowledge of the subject, pedagogy and technology is essential for the formation of a good teacher. As much as they are considered as three separate parts of the teacher's knowledge, this model further emphasises the complex interaction between them (Mishra & Koehler, 2006).

Computational thinking has four tightly interrelated pillars that are usually systematically applied:

Decomposition refers to the operation of breaking down a complex system into smaller components that are easier to grasp, manage, explore, and understand. Each smaller problem is then solved on its own time and some tiny little details around it could be easily taken into account. Decomposition gives us the opportunity to examine a problem/situation, etc. from a wide scope of dimensions and angles, and to understand how it works from its innermost core. In the process of decomposing a larger and more intricate problem, we often detect/recognize patterns.

Pattern recognition (PR) could be described as the second pivotal feature of Computer Science. The patterns represent similarities or characteristics that are common for the problems/content the system processes (text, images, sounds, etc.). Patterns' nature could be described as sequential, or rhythm based. Thus, pattern recognition includes detecting these similarities/patterns among decomposed/segmented problems and enable us to solve the bigger problem more easily. There are two major PR operations:

- *explorative* - used to recognize commonalities in the data

- *descriptive* – used to categorise the commonalities in a certain manner (http://theapp solutions.com).

Moreover, the list of upcoming conferences of the International Association for Pattern Recognition (IAPR) provokes great interest – the thematic scope is wide and ranges from human computer interaction to environmental monitoring (see iapr.org).

Abstraction is a crucial and quite high conscious and cognitive ability that human beings predominantly possess. It provides us with the opportunity to filter the flow of hybrid data and erase the irrelevant amount of information (usually the details) to emphasise the one we want or need for the successful problem-solving process. This selective operation results in creating a pure representation, mental image, idea of what our problem consists in and to create a model or an algorithm to follow.

Algorithmic thinking consists in building up a series of concrete steps, rules, directions, instructions to solve a problem. AT could further be defined as elaborating a well-ordered strategy, an efficient easy-to-follow plan of action to cope with some complicated tasks. It is necessary to set a starting point, a finishing point, a spectrum of steps to follow in between. Algorithms are the leading tool for programming computers and identifying patterns.

ICT competence of teachers according to the conceptual framework of UNESCO

The framework proposed by a UNESCO team (UNESCO, 2008a) emphasises that it is not enough for a modern teacher to be scientifically literate and able to form relevant scientific skills and habits in his students. The modern teacher must be able to help students use ICT to work together successfully, to deal with emerging problems, to learn learning skills and, ultimately, to make contributing, responsible, and competitive students, citizens and employees.

The standards developed in this framework are related to the following aspects of the teacher's work - modules: understanding the role of ICT in education; learning content and assessment; pedagogy (didactic skills); ICT skills; organisation and management of education; Professional Development.

The main idea of the authors of this standardisation system is based on three approaches to learning adopted by them, which are related to the respective stages of professional development of teachers learning to work with ICT. The first approach - "Technology Literacy" - requires teachers to be able to help students use ICT to improve learning. The second - "Depth of Knowledge" - requires teachers to help students more fully master the curriculum in the subject and apply the knowledge gained to solve complex problems occurring in the real world. The third approach - "Creating knowledge" - requires teachers' skills to help students, future citizens and workers to produce (create, acquire) new knowledge that is necessary for the harmonious development and prosperity of society.

Computational Thinking is often viewed as explicitly related to programming, coding and CS (Computer Science), especially after the publication of Wing's seminal paper (Wing 2006). Efforts to develop students' CT thus tend to develop and use activities, tools or platforms associated with programming knowledge and skills in educational interventions and programmes (Barth-Cohen et al. 2018; Hsu et al. 2018). As an example, Barth-Cohen et al. (2018) investigated how fifth graders interpret and navigate information when participating in various coding and problem-solving activities during a programming environment. The school adopted and used a robotics curriculum in their study, had software installed in school-provided laptops for college kids, and had one physical robot for instructional use. Students' CT development was examined with a focus on their performance in formulating and solving problems during this robotic programming environment (Ji Shen, Guanhua Chen, Lauren

Barth-Cohen, Shiyan Jiang & Moataz Eltoukhy (2022) Connecting computational thinking in everyday reasoning and programming for elementary school students, Journal of Research on Technology in Education, 54:2, 205-225).

No doubt, digitalization possesses an extending impact on school education. Lately, in political contexts, this has been constantly outlined as the term "digital education". While this term is frequently associated with the use of technology, there are also approaches that concentrate additionally on Computational Thinking. CT describes patterns of coping with a problem as a computer scientist would (Wing, 2006). Although there is no comprehensive definition, there is mutual agreement that it includes a set of skills for thinking about and dealing with problems (Kalelioglu, Gülbahar, & Kukul, 2016). It's about applying expertise, like abstraction or breakdown, so that the resolution can be effectively carried out by a computer. Original approaches to include CT into secondary school teaching relate mainly to science education, e.g. by teaching science through simulation and modelling (Basu et al, 2013). But the approaches and achievements of the digital humanities show that teaching in other subject areas could be also strongly influenced by digitalization. Accordingly, some sources consider the embedding of CT across the different subject areas (Barr & Stephenson, 2011, Kale et al., 2018). Curve fitting or doing a verbal analysis of sentences are some examples (Barr & Stephenson, 2011). The model of Computational Learning (CL) emphasises the important role that "computer (and possibly its abstraction) can play in enhancing the learning process and improving achievement of students in the field of STEM and other courses" (Cooper, Pérez & Rainey, 2010). The model combines learning theories and the ability of the computer to deal with complexity and visualise results in a suitable way to improve comprehension as well as learning. While this concept involves the use of computers, e.g. for simulations and modelling, the authors emphasise that the model explicitly excludes non-cognitive uses of technologies such as clickers or blogs. Teachers can engage their students in CT and CL owing to computer science skills and knowledge. Some universities started to apply CS courses for many non-CS student groups, while there are a lot of pre-service teacher trainings concentrating primarily on developing and improving information and communication technology (ICT) skills (Goktas, Yildirim, & Yildirim, 2009).

Regardless of the importance of Computational Thinking in all subject areas throughout education, existent approaches explicitly including CT-related skills and competencies in non-CS teacher training are still uncommon. Existent CS courses for non-CS teachers tend to concentrate, for instance, on everyday phenomena that they analyse and explain from a computer science point of view (Müller, Frommer, & Humbert, 2013), on algorithmic conceptions (Yadav et al., 2011), or on focusing Computational Thinking in the context of Information & Media Literacy (Dengel & Heuer, 2018).

Computer science knowledge and skills are necessary to know the digital transformation and advances within the subject areas. In addition, the digital transformation also leads to new issues becoming relevant to be discussed in class (cf. Brinda & Diethelm, 2017). Due to their history of education, many prospective teachers still do not have a correct foundation in computing. Therefore, a course must be supported by the basic ideas of computer science that underlie the digital transformation and highlight impacts on society. In order to be suitable for prospective teachers of all disciplines, a corresponding course must also take the various levels of prior knowledge under consideration.

REFERENCES

Дамянова, А. (2005). Конструктивизмът - новата образователна парадигма? Български език и литература (ел. версия), (5). https://liternet.bg/publish3/adamianova/konstruktivizmyt.htm

Тафрова-Григорова, А. Кирова, М. & Бояджиева, Е. (2012). Учителите по природни науки – за конструктивистката учебна среда в българското училище. Химия, 21, 375-388.

Тошев, Б. В. (2007a). Ранното обучение по природни науки- интегрален или модулен подход. Bulgarian Journal of Science & Educational Policy, 1, 51-56.

Тошев, Б. В. (2009a). Метод на проектите в образованието. Химия, 18, 243-249.

Тошев, Б. В. (2012). Конструктивизъм – теория и практика. Химия, 21, 463-468. Холенбек, Д. Кирова, М. Бояджиева, Е. & Тафрова-Григорова, А. (2009). Ученици и учители за настоящата и предпочитана учебна среда – резултати от едно проучване. Химия, 18, 349-369

Цанова, Н. & Райчева, Н. (2012). Методика на обучението по биология. Теория и практика. София: Пенсофт.

National Research Council (NRC), *How Students Learn: History, Science, and Mathematics in the Classroom*. Washington, DC: National Academy Press, 2005.

J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33-36, 2006.

C. P. Dwyer, M. J. Hogan, and I. Stewart, "An integrated critical thinking framework for the 21st century," *Thinking Skills and Creativity*, vol.12, pp. 43-52, 2014.

President's Council of Advisors on Science and Technology (PCAST). (2010). *Prepare and Inspire: K-12 Education in Science, Technology, Engineering, and Math (STEM) for America's Future*. Washington, DC: White House Office of Science and Technology Policy (OSTP) 2010.

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. Learning & Leading with Technology, 38(6), 20-23.

Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? ACM Inroads, 2(1), 48-54.

Barth-Cohen, L.A., Jiang, S., Shen, J., Chen, G., & Eltoukhy, M. (2018). Interpreting and navigating multiple representations as central to computational thinking in a robotics-programming environment. Journal for STEM Education Research, 1 (1-2), 119-147.

Brinda T., & Diethelm, I. (2017). Education in the Digital Networked World. Proceedings of WCCE 2017. Cham: Springer, 653-657.

Cooper, S., Pérez, L., & Rainey, D. (2010). K-12 Computational Learning. Communications of the ACM, 53(11), 27-29.

Goktas, Y., Yildirim, S., & Yildirim, Z. (2009). Main Barriers and Possible Enablers of ICTs Integration into Pre-service Teacher Education Programs. Journal of Educational Technology & Society, 12(1), 193-204.

Ji Shen, Guanhua Chen, Lauren Barth-Cohen, Shiyan Jiang & Moataz Eltoukhy (2022) Connecting computational thinking in everyday reasoning and programming for elementary school students, Journal of Research on Technology in Education, 54:2, 205-225

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational What? Relating Computational Thinking to Teaching. TechTrends, 62(6), 574-584.

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking based on a Systematic Research Review. Baltic Journal of Modern Computing, 4(3), 583.

Müller, D., Frommer, A., & Humbert, L. (2013). Informatik im Alltag– Durchblicken statt Rumklicken [CS in Everyday Life - Understanding Instead of Clicking Around]. Proceedings of HDI '13, 98-104.

Wing, J. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35.

Mayer, R. (2001). Multi-media learning. Cambridge, UK: Cambridge University Press.

Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. In R. Bruning, C. A. Horn, & L. M. PytlikZillig (Eds.), Web-based learning: What do we know? Where do we go? (pp. 23–44). Greenwich, CT: Information Age Publishing.

Cuban, L., Kirkpatrick, H., & Peck, C. (2001). High access and low use of technologies in high school classrooms explaining an apparent paradox.

Blackmore, J., Hardcastle, L., Bamblett, E. & Owens, J. (2003*). Effective use of information and communication technology (ICT) to enhance learning for disadvantaged school students*. Canberra, ACT: Department of Education, Science and Training.

Davis, F.D. (1993). User acceptance of information technology: system characteristics, user perception and behavioral impacts. *Int. J. Man-Machine Studies*, *38*, 475-487

Davis, F. D., Bagozzi, R. P. & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, *35*, 982–1003.

Dirckinck-Holmfeld, L., Hodgson, V., Jones, C, de Laat, M., McConnell D. & Ryberg, T. (2010). Teacher Use of ICT: Challenges and Opportunities. In: Dirckinck-Holmfeld, L., Hodgson, V., Jones, C., de Laat, M., McConnell, D. &

Ryberg, T. (Eds). *Proceedings of the seventh International Conference on Networked Learning*. (pp 439-445). Adlborg Universitet.

Donnelly, D., McGarr, O. & O'Reilly, J. (2011). A framework for teachers' integration of ICT into their classroom practice. *Computers & Education*, *57*(2), 1469-1483.

Ertmer, B. (1999). Addressing first- and second-order barriers to change: Strategies for technology integration. *Educational Technology*, *Research and Development*, 47(4), 47-61.

Kumar, R. & Gupta V.K. (2009): An introduction to cognitive Constructivism in Education. *Journal of Indian Education. Nov.2009. New Delhi: NCERT.*

Mishra, P. & Koehler, M.J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, *108*, 1017-1054.

Polman, J.L. (1998). *Why train "little scientists": The purposes and practices of science education in today's democracy*. Paper presented at the Annual Meeting of the American Educational Research Association, San Diego, CA.

Roblyer, M. & Edwards, J. (2000). *Integrating educational technology into teaching* (second Ed.). Upper Saddle River, NJ: Prentice Hall.

Roblyer, M. (2006). *Integrating educational technology into teaching* (fourth Ed.). Upper Saddle River, NJ: Pearson Education.

Semenov, A. (2005*). Information and communication technologies in schools: A handbook for teachers, or how ICT can create new, open learning environments*. Paris, France: United Nations Educational, Scientific and Cultural Organisation.

Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, *15*, 4-14.

Singh, S. & Yadivashi, S. (2015). Constructivism in science classroom: Why and how. *International Journal of Scientific and Research Publications*, *5*(3), 1-5.

UNESCO. (2008). ICT Competency Standards for Teachers. Policy Framework. http://cst.unesco-ci.org/sites/projects/cst/The%20Standards/ICT-CST-Policy%20Framework.pdf

UNESCO. (2008a). ICT Competency Standards for Teachers. Paris: United Nations Educational, Scientific and Cultural Organization.

UNESCO. (2008b). ICT Competency Standards for Teachers. Implementation Guidelines.http://cst.unescoci.org/sites/projects/cst/The%20Standards/ICT-CS Implementationpercentage20Guidelines.pdf

Watson, R. (2000). The role of practical work. In: Monk, M & Osborne, J (Eds.). *Good practice in science teaching – What research has to say*. (57-71). Buckingham: Open University Press.

Chapter 2: Computational Thinking Frameworks and Education Online

2.1 Introduction

Computational Thinking (CT) is a part of a broader term of digital literacy, and it is considered to be an important *21st-century skill*. The biggest benefit of CT is how it enables real-world problem-solving, helps to break down big problems into small parts and offers solutions in a variety of fields. With knowledge of CT, students can learn how to solve maths problems or write book reports (Smith, 2014).

There are many theoretical frameworks related to CT, the ones presented in this chapter provide useful knowledge for a teacher whose task is to design online lessons to develop CT skills in students.

The following material will give the reader a better understanding of what CT is and what activities can be done to develop CT in students. The COVID-19 pandemic has accelerated the transition to the online teaching mode, this is why in this handbook, we will be focusing on developing CT in this context.

Theoretical frameworks of CT will be presented. Among them, the framework of Brennan and Resnick (2012) and the Computational Thinking Pedagogical Framework of Kotsopoulos et al. (2017). Then, it will be explained how to integrate CT across the curriculum.

2.2 Computational Thinking Frameworks

2.2.1 Framework for Studying and Assessing the Development of Computational Thinking by Brennan and Resnick (2012)

Brennan and Resnick (2012) developed a framework that can help better understand CT. They were interested in presenting how project-based activities support the development of CT in young people, and over the past several years they developed a framework to summarise their conclusions.

The authors based their theory on activities performed with CT tool - Scratch – a programming environment that enables young people to create their own interactive stories, games, and simulations, and then share those creations in an online community with other young programmers from around the world.

In their model, Brennan and Resnick (2012) distinguish:

- **COMPUTATIONAL CONCEPTS** - The concepts young people encounter as they program.
- **COMPUTATIONAL THINKING PRACTICES** - strategies and practices young people adapt to develop interactive media.

In regard to computational concepts, we distinguish the following:

- **Sequences:** a particular activity or task is expressed as a series of individual steps or instructions that can be executed by a computer.
- **Loops:** a mechanism for running the same sequence multiple times.
- **Parallelism:** sequences of instructions happening at the same time.
- **Events:** one thing causing another thing to happen.
- **Conditionals:** decision-making based on certain conditions, which supports the expression of multiple outcomes.
- **Operators:** enable programmers to perform numeric and string manipulations, including addition, subtraction, multiplication, division and so on.
- **Data:** involves storing, retrieving, and updating values.

Brennan and Resnick (2012) name the following computational practices:

- **Testing and debugging:** developed through trial and error, and transferred from other activities, or support from knowledgeable others.
- **Reusing and remixing:** is in other words, building on other people's work.
- **Abstracting and modularizing:** is creating something large by putting together collections of smaller parts, is an important practice for all design and problem-solving.

Young people are surrounded by interactive media. Websites they use in order to be created require Computational Thinking skills from their creators. The goal of teaching children CT should be changing their perspective: from a consumer to a person thinking "I can create something like that" or "I know how this was done".

Creativity and learning are part of the computational process. They are social practices that enrich the process and should be structured through online and face-to-face interactions. By creating with others, young people are able to do more than they could have on their own.

The framework of Brennan and Resnick (2012) offers an understanding of the capabilities of CT and illustrates the avenues for knowledge acquisition it opens up for students. The authors stress the importance of assessing the skills acquired during learning activities. The theoretical framework is a map showing us the way, but the key stage of working with students must be to determine if they acquired the necessary skills after the learning is over.

In order to determine if the learning activity was a success, teachers needs to evaluate students by seeking answers to the following questions:

- Can they put to practice the knowledge they acquired during the lesson?
- Can they remix or reuse concepts that were employed by others?
- Are they capable of evaluating their codes and the codes of their peers?
- Do they know how to debug problematic codes?

Brennan and Resnick's Framework captured the what, how, and why in their CT framework, however they didn't fully address the actual teaching of CT. This is why the next framework will fill the gap in understanding of the pedagogical aspect of developing Computational Thinking with students.

2.2.2 Computational Thinking Pedagogical Framework (CTPF) proposed by Kotsopoulos et al. (2017)

Numerous frameworks for CT have been proposed, but the framework that describes pedagogy has been a unique, pioneering work by Kotsopoulos et al. (2017). The authors propose the CT Pedagogical Framework (CTPF), which includes four pedagogical experiences:

- **Unplugged** experiences focus on activities implemented without the use of computers. They are often the first and foundational experiences in learning CT because they require possibly the least amount of cognitive demand and technical knowledge. The purpose of unplugged experiences is to introduce preliminary and overlapping concepts related to CT. Nishida et al. (2009) advise teachers to focus on student-directed activities that are easy to implement. Teachers can use games to teach or use gamification methods to embed their teaching with challenges and rewards.

- **Tinkering** experiences involve activities that take things apart, change and/or modify existing objects. These objects can be building blocks, puzzles, digital or electronic simulations, programming code, and so forth. Students are not constructing an object, digital or otherwise, they explore changes to existing objects and then consider the implications of the changes. The goal is to provide a context for exploring incremental modifications, without the additional challenge of actually building the object.
- **Making** experiences involve activities where constructing new objects is the primary focus. In making experiences, students are required to: solve problems, make plans, select tools, reflect, communicate, prototype and test, make connections across concepts.
- **Remixing** refers to experiences that involve the appropriation of objects or components of those with the aim to make use of it for other purposes or in other objects. Remixing involves modifying or adapting objects in some way, and requires a significant level of proficiency to identify a usable object and then adapt and modify it.

In summary, the framework developed by Kotsopoulos et al. (2017) primarily describes what activities a student can undertake to learn computer thinking. The list ranges from the simplest - unplugged and tinkering experiences to those requiring more perspective, better processed CT knowledge - making and remixing.

However, the activities mentioned do not have to be applied only to Computer Science learning. As we will show in the following section, CT can be treated as a skill such as critical thinking or problem-solving which can be learned by applying it to various school subjects as part of a cross-curriculum approach.

2.3 Computational Thinking in Compulsory Education: How to include CT in school curricula?

"The underlying idea in computational thinking is developing models and simulations of problems that one is trying to study and solve."
- Dave Moursund

In the knowledge society, organisations operate in the context of a global economy characterised by intense competition, economic interdependence and collaboration (Van Laar, et al., 2017). Employees need sufficient skills to adapt to the changing requirements of their jobs. The mostly asked question is: Is the learner enabled to put the concepts into practice? (Ahmad, Karim, Din, & Albakri, 2013; Carnevale & Smith, 2013). This means that adaptation has a huge significance in the 21st century. Researchers, while naming such 21st-century skills, often include collaboration, communication, digital literacy, citizenship, problem-solving, critical thinking, creativity and productivity (Voogt & Roblin, 2012).

Computational Thinking is a relatively new term, but while researchers started paying more attention to it, it became abundantly clear that CT is the skill of the 21st century (Black et al., 2009). CT allows non-computer scientists to benefit from a computational approach to problem-solving (Cuny et al., 2010). It helps them understand problems that are computable, to determine the correct tools and methods for solving certain problems, as well as helps the exploration of method limitations. Almost all disciplines have now been influenced by computational thinking in some way, in both the sciences and humanities (Mohaghegh, McCauley, 2016).

We have to assume that many of today's students have daily contact with computers and the virtual world. Learning how to operate the devices and how to use them is a basic skill, CT is a more targeted approach where students are taught to operate in and outside the digital world. CT enables students to think in a different way, express themselves through various media, solve problems and analyse everyday issues in an organised manner.

Digital literacy should not be confused with Computational Thinking, however, those two terms are overlapping. Digital literacy is primarily the study of how students can competently and safely use digital tools and resources. In contrast, CT is a competence that can be acquired by those who are already digitally literate. It is a more complex concept that goes beyond using computers or programming. By developing CT skills, students can simultaneously develop logical thinking skills, and problem-solving skills, as well as learn how to code and program. Ioannidou, Bennett, Repenning, Koh, & Basawapatna (2011) claim that learning CT skills includes:

→ formulating problems for the usage of computers in order to facilitate the solution;
→ logically organising and analysing data;
→ representing data through abstractions;
→ automating solutions through algorithmic processes;
→ identifying, analysing, and implementing possible solutions as the most efficient and effective combination of steps and resources; and
→ generalising and transferring this process to a variety of problem areas. We have merged and emphasised these (computational) components in our learning progression

When we talk about including CT skills in the school curriculum, we can take two approaches:

→ placement of CT within a specific subject such as Mathematics or Computer Science as a standalone activity. CT is a goal of learning in this approach.
→ holistic approach that assumes CT has the potential to be used in an interdisciplinary practice across different scientific fields. The holistic approach requires teachers to compare and contrast CT applications across different situations. Learners then are pushed to anticipate other situations where CT can be applied.

As was said by Lockwood, Money, 2017:

"For some schools, it might not be possible to run a curriculum in which CT can be taught separately. As such it is important that efforts are made to ensure that CT is not only taught in regards to Computer Science but that it can also be incorporated into many other areas of education. Even if it is possible to teach CT outright it is important that students are made aware and shown how CT skills and knowledge can be applied to many areas".

By engaging students in computational thinking across multiple domains, while using the same programming constructs and modelling environment, teachers can enable students to realise the commonalities across domains in terms of the underlying computational/mathematical constructs and practices. The meaningful immersion in authentic learning experiences happens over a period of multiple years (Lehrer et al., 2008). Therefore, the use of computer thinking to solve and understand problems that are present in domains unrelated to Computer Science or Mathematics will only help students to get a better grasp of this concept and develop more meaningful skills.

According to Kolodner [p 57] "*...it is important not to fall prey to the mistaken notion that if one learns computational thinking skills in one context, one will automatically be able to use them in another context. Rather, it will be important to remember that one can learn to use computational thinking skills across contexts only if they might use the same skills (and how they would)."*

CT can deepen the understanding in various scientific fields such as chemistry, biology, social sciences, foreign languages, etc. As Resnick (2011) argued, most people work better on things they care about and that are meaningful to them, so embedding the study of abstraction in concrete activity helps to make it meaningful and understandable. Stewart and Golubitsky (1992) point out that regardless of the domain, scientists' work involves building and refining models of the world. Therefore, model building through programming corresponds to core scientific practice. According to Sengupta, Kinnebrew, Biswas, and Clark (2012) studies have shown that through programming, students can construct representations of physical concepts such as speed and acceleration, and achieve a more profound understanding of these concepts.

Examples of CT implementation across the subjects:

Physics	CT practices are central to the development of expertise in a variety of STEM disciplines. Sengupta, Kinnebrew, Biswas, and Clark (2012) designed a learning environment that supported the learning of CT through modelling and simulation. Their product focused on modelling Newtonian mechanical phenomena such as the trajectory, velocity, and acceleration of balls placed in a specific environment. Students could influence this environment and observe how it reacts.
Geography	Anuar, Mohamad, and Minoi (2021) showed that through art students can grasp essential concepts of computational thinking such as abstraction, decomposition and algorithms. The study utilised problem-based learning to design computational thinking activities for 22 students aged 10 to 11 years old. Abstraction skill was trained through drawing and colouring the travelling map with appropriate labelling. Decomposition skill was trained when students were asked to sketch a simplified travelling map, and algorithmic thinking was required when students had to draw a dot for each location on the grid and write each path as an algorithm. Participants had to logically arrange and analyse data, and then represent this data through abstractions (drawings of a building, road, etc.).

Literature	Cabo and Lansiquot (2016) created a course: *Programming Narratives: Computer Animated Storytelling (PN)*, designed to help college students develop computational thinking skills through computer programming and the development of writing skills. Two distinct disciplines of English and Computer Science were connected for the purposes of this course. The student's goal was to create a narrative-driven video game prototype. Students then created stories as a computer program. They were introduced to the use of flowcharting techniques and programming structures, such as sequencing, repetition loops, and decision statements, to solve a problem with an algorithm, to concepts of object-oriented programming, such as classes, objects, properties and methods.

Above, we demonstrated how computational thinking can be applied to various fields of learning. However, when thinking about activities using CT, it should not be forgotten that it is equally important to adapt them to the students' skill level, as well as their age. Designing a game or creating a simulation are tasks that require a certain level of skills and knowledge. In the following examples of how integrated computational thinking lessons can be introduced into school.

Milks, et al., (2021) gives examples of computing integrated into the K–12 curriculum sorted by age. They suggest following activities:

For 4 to 5 age group:

- Break down the task of creating an object into smaller steps. An object can be made out of modelling clay, paper, beads, etc.
- Perform a sequence of music notes or dance moves.

- Use dice to select an action or sound to make (e.g. frog sound if you roll a 3, chicken sound if you roll 5). Another dice can be used to determine the number of times to repeat that action or sound.
- Create graphs of data relevant to your students or classroom (e.g., birthdays).
- Sort objects according to shared attributes (e.g., animal kingdom, food groups).
- Draw what a tree looks like at different times of the year.
- Create a map and provide precise instructions for a robot to travel to different destinations on the map.

For 8-11 age group:

- Compose a song using a computational tool such as Scratch
- Identify the large task of making a playdough tree, identify bigger and smaller tasks, such as making a trunk, leaves, and apples.
- Put the plot of a story in the correct order.
- Break down an everyday routine into smaller parts and make a mini-book to tell the story of their very own routine.
- Use Scratch to create programs that calculate area and perimeter given user inputs.
- Develop stories of different historical perspectives using data from primary sources.
- Use a simulation to explore scientific phenomena (e.g., how mass affects force when two objects collide).

For 11 to 18 age group:

- Create a visualisation that emphasises bias or injustice with a dataset.
- Develop an algorithm for decomposing an essay prompt
- Use to decode or remix a computational model about a scientific phenomenon (e.g. weather).
- Analyse and evaluate bias in second-hand data about socio-scientific issues (e.g. cancer, pollution).

- Analyse data critically examining socio-political structures within society and technology (e.g., redlining and digital redlining).

In summary, computing permeates all disciplines and areas of knowledge, and so it could be argued that computational thinking—including the application of computing concepts and skills to solve problems, not only in Computer Science but also in other disciplines—should be a part of the twenty-first-century education used as a tool for a broad range of college students, including those not majoring in computing (Guzdial, 2008; Wing, 2006).

2.4 Model of Computational Thinking for Education Online

2.4.1 Introduction

Online learning was invaluable during the pandemic. Teachers have become familiar with online teaching tools, and we assume that they will use some of them in one form or another in the future. This is why – regardless of the epidemic situation – online learning is here to stay.

What form of learning a teacher chooses may be based on either school context or personal preference. The selected mode of teaching may determine the activities carried out in class and how progress is monitored and results assessed.

- **Online learning** is conducted with the help of the Internet in a fully online environment. Students and teachers from different physical locations interact with one another via an Internet connection. Webinars, virtual classroom sessions, and online whiteboard collaborations are forms of online learning.
- **On-site learning** requires teachers and students to be physically present for the lesson. It means that teachers communicate and share learning materials with students in a physical classroom. This mode of learning is limited geographically and less accessible than online learning.

- **Hybrid learning** is an approach to education that combines computer-mediated activities and online educational materials with traditional place-based classroom methods. In this setting, students learn through a mix of in-person and online activities. Hybrid learning supports and reinforces on-site learning. Students can access materials like lectures, readings, and discussion boards at any time and learn at their own pace. It is also a method implemented in a situation when some of the students cannot be present in the physical classroom for some reason (for example, health issues or travel issues).

Online learning is similar to on-site learning, but often requires the knowledge of different tools and engagement methods. We have included descriptions of various teaching modes in the context of our model of Computational Thinking for Education Online to show possibilities and differences between different environments of learning. In the following subchapters, we will describe how to facilitate learning in an online or a hybrid setting, how to select tools for online learning, how to plan and implement activities to develop CT skills.

2.4.2 Teachers roles and responsibilities in facilitation of online learning

Bosch (2016) points to six basic pedagogical goals of learning: contents, social and emotional support, questioning, encouraging reflection, fostering collaborative learning, and making an evaluation.

There are many ways in which content can be delivered and presented in an online setting or a hybrid setting. Teachers can use learning management systems (CMS/LMS) such as: Blackboard, Canvas, or Moodle. Once it's decided what tools to use to teach the lesson and share the learning materials with students, there's a need to focus on how the lesson should be taught.

Sousa (2016, p.17) describes student engagement as *"the amount of attention, interest, curiosity, and positive emotional connections that students have* *when they are learning, whether in the classroom or on their own"* (2016, p. 17).

As Bosch points out, it is important to support students socially and emotionally. Students need someone to speak to, whether to help them understand a complex concept or to provide advice which naturally leads us to discuss what is a learning community and why building it is important in an online setting.

The term "learning community" can mean a variety of things, but on the most basic level, the community is people who interact with each other. The community can be built in both face-to-face and online settings. With the emergence of the COVID-19, educators were required to engage students in online learning environments, and this situation separated students from their teachers and peers, creating a threat to emotional wellbeing and social development. This is why it is important to stress that community building is possible and necessary in an online setting, and it is a part of a teacher's role to facilitate interactions between students.

Stimulating intergroup discussion by asking the "right" questions can help students think critically about a topic or issue. The Socratic method can be particularly useful for engaging and challenging students intellectually in order for them to understand difficult issues. Pedagogical activities that require students to reflect on what they learn and to share their reflections with their teachers and fellow students extend and enrich the learning experience.

It is also encouraged to involve students in group projects, as collaborative learning is an essential tool for creating knowledge, as well as fostering relationships between students, generating peer review and evaluation (Fredericksen, 2015).

The Academic Communities of Engagement (ACE) framework (Borup et al., 2014; Borup, Graham, et al., 2020; Borup, Jensen, et al., 2020) the presence of three different community roles is taken into account. Teachers, peers and parents in a learning process as a part of the learning community. The framework authors assert that the more engaged and present community actors are the more students are engaged in the process of learning.

ACE Framework also identifies responsibilities of teachers that include:

- **Orienting**: helping students understand expectations, systems, and strategies for learning online
- **Instructing**: providing students with feedback and tutoring that directly impacts their understanding of the course curriculum.
- **Organising and Designing**: providing students with a learning environment and learning activities that foster learning
- **Nurturing**: establishing close, caring student-teacher relationships
- **Facilitating communication**: encouraging communication with and between students, parents, and other stakeholders
- **Monitoring and motivating**: tracking student progress and motivating them to be fully engaged in learning activities (Borup, Graham, & Drysdale, 2014; Borup & Stevens, 2016, 2017).

As for the evaluation process itself, there are many tools available online for creating tests, essays and surveys. However, there are also many artefacts that students can create and evaluate, examples include: classroom presentations, YouTube videos and podcasts, weekly class discussions on discussion boards or blog posts that can be reviewed over and over again to examine how students have participated and progressed over time. They are also most helpful to instructors to assess their own teaching and to review what worked and what did not work in a class. Unlike face-to-face group work that typically ended up on the instructor’s desk, online-generated content can be shared with others beyond the classroom (Bosh, 2016).

To summarise, in order to develop students' knowledge and skills, a teacher needs to choose the most optimal teaching strategies. It is important to create clear learning objectives and inform students about them. In addition, while working in an online environment, it is essential to know the possibilities of the tools that a teacher plans to use in the virtual classroom. Acquiring knowledge is a process with many obstacles, however, with good organisation of work and healthy, positive environment for learning, students can learn more effectively and better understand discussed contents. By building good student-teacher, student-peer relationships and involving parents in the learning process (especially with hybrid and online education), we are likely to see long-lasting and positive results.

Monitoring progress and providing feedback is also essential in designing an effective learning environment. Knowing how well students understand discussed concepts, and how effective are their implementation skills, allows the teacher to evaluate the educational strategy. In turn, a student who is able to monitor his progress is motivated to continue the improvement of their skills and knowledge.

2.4.3 Planning and implementing activities to develop CT skills

Knowing the responsibilities and functions of a teacher in an online environment, we can now move on to planning activities aimed at developing computer thinking skills.

Before we start discussing specific activities, we will briefly discuss the activity planning process, specifically the lesson plan design process. The lesson plan can be a useful guide for a teacher, it should reflect teaching philosophy, and most importantly, teacher's learning objectives (Jensen 2001; Nesari and Heidari 2014).

A lesson plan usually involves information such as: *grade*, *subject*, *lesson duration*, *date*, *lesson topic*, *learning objectives*, *teaching and learning activities*, *time allocation of activities*, *teaching resources and assessment activities*. It also reflects a teacher's beliefs, understanding of, and orientation towards the curriculum, the subject that is being taught, and the pedagogy (Li and Zou 2017; Shen et al. 2007). At the beginning of this chapter, we presented the framework of Brennan and Resnick. This framework serves as the basis for the curriculum, or long-term action plan. A lesson plan, however, aims to set short-term, measurable goals that will allow us to have control over the students' learning process and the quality and fit of the educational methods used. Objectives or learning goals are crucially important for the process of designing educational activities. Clear goals help the learners understand the purpose of the learning. They help the teacher plan and deliver instruction at an appropriate level, and they also help with assessments and evaluation of learning activities later on.

Planning the lesson beforehand:

- increases productivity
- allows avoiding teaching out of context/time-wasting
- enables teachers to organise information to be presented to learners
- allows setting clear goals on what the teacher wants to achieve
- students know what they are going to be assessed on
- allows teacher to assess if learning objectives were met
- allows placing the lesson within curriculum policy

To create a lesson plan, consider:

1. How do CT activities fit into the curriculum?
2. How to assess what knowledge and skills a student needs to practice CT?
3. How much time each activity will tak?
4. What teaching tools will you use, how (you can use programs like Scratch, Alice, AgentSheets, etc.?
5. How to design activities that correspond with a student's current skills and knowledge?
6. How to make evaluation and assessment?

In order to effectively plan learning objectives for computational thinking, we need to consciously plan easy activities at the beginning and slowly move to increasingly difficult ones. In understanding this process and planning learning objectives, we can rely on Bloom's taxonomy, which divides learning objectives into levels of complexity and specificity.

Bloom mentions six educational objectives that describe the cognitive processes that students are required to use in order to learn, these are: Remember, understand, apply, analyse, evaluate, create. Learners must approach a topic or subject from the lowest level (remember) before moving on to higher levels of thinking.

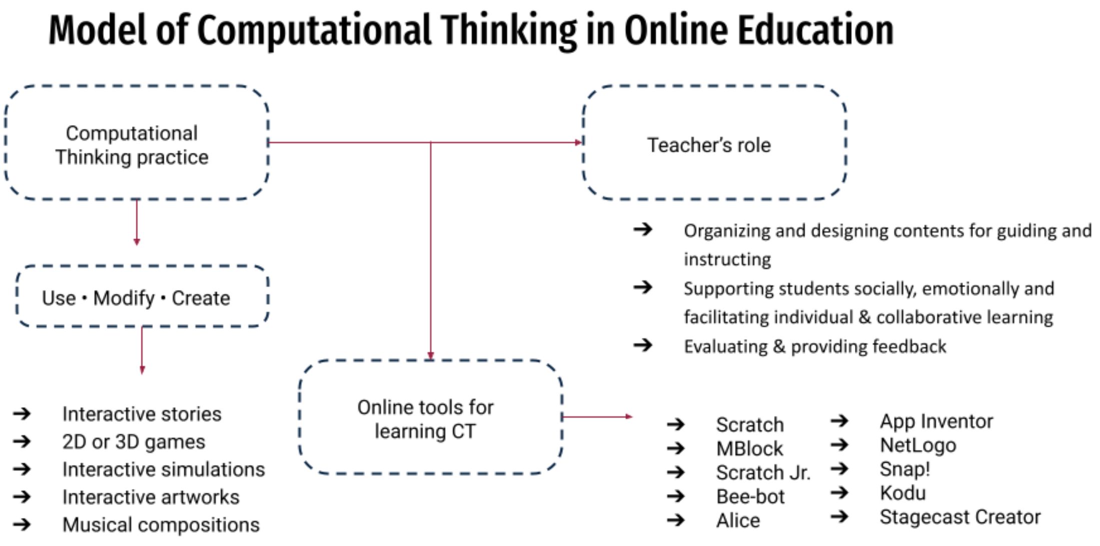
Remembering involves recognizing and recalling as well as describing, identifying, listing. Understanding requires actions such as interpreting, paraphrasing, explaining, classifying, summarising or comparing. Applying happens when students are asked to use, solve, predict, apply or demonstrate a task or action, they would likely be working at this level of thinking. Analysis happens when students can draw connections between ideas and utilise their critical thinking skills. Teachers can ask students to differentiate, compare, deconstruct facts or data. Evaluation happens when learners test, check, or critique artefacts. Finally, creating is the highest level of learning in Bloom's taxonomy, it requires planning and producing something. Students may have to write, design, combine, devise or modify something.

As postulated in Bloom's Taxonomy, in order to create effective learning activities it is necessary to deepen the student's understanding, then slowly and gradually engage students in increasingly complex activities. We can see the same philosophy in the USE-MODIFY-CREATE Framework (Lee et al., 2011), which aims to demonstrate how computational thinking should be taught. The framework USE-MODIFY-CREATE as the name suggests, consists of three phases, which we will briefly describe below:

- **USE**: During this phase, students learn how to use existing programs and projects that others have made. This may involve performing scripted tutorial operations and exploring new software.
- **MODIFY**: As comfort is gained in using the tools, students can begin to experiment by modifying existing programs or projects, making increasingly original contributions. During this phase, students begin to understand how they can control underlying mechanisms to bring about different results.
- **CREATE**: In this phase, students apply their growing skills to create an original product. This work will show increasing levels of abstraction and automation than may have been present in earlier activities.

To summarise, in the above-mentioned theory we move from the simplest activity (using a ready-made tool) to the most difficult activity: creating artefacts for assessment and evaluation. For example, the teacher can ask students to create a simulation, interactive story, game, or animation.

2.5 Summary



As Lee et al. (2021) previously stated, computational thinking can be taught through three activities: using, modifying, creating.

Students can create various artefacts that will later be evaluated. These artefacts can take the form of interactive *stories*, *games*, *simulations*, *works of art or musical compositions*. Students can also use and modify codes that others have made.

A significant number of programs for CT learning are available online. Amongst the most popularised ones are Scratch, Scratch Jr, Alice, AgentCubes, App Inventor. We discuss the functions of these programs in the third chapter of this Handbook.

In addition to programs aimed at developing computational thinking, the teacher also has at his disposal a number of programs that will enable him to provide online instruction, organise online group activities, and gather feedback from undertaken activities in a digital form. There are also tools for communication between participants, e.g. forums, social media and chat rooms that enable students to have a discussion and share reflections. They're vehicles for group discussions and sharing reflections. Finally, there are tools for quizzes and surveys that are useful for feedback and evaluation purposes.

In our model, we try to combine the online teaching model with the model of teaching computer thinking. Aware that the nurturing of computational thinking is possible primarily through the use of tools available in digital form, we described the specifics of the work of a teacher in an online environment and the specifics, activities and products generated in the learning of computational thinking.

Knowledge check

Answer the following questions to your best ability:

- How can we define CT as a key 21st-century skill for school children?

....

....

....

- How can CT be applied in a subject that is not Mathematics or Computer Science?

....

....

....

- What is needed to further develop the CT agenda in compulsory education settings?

....

....

....

- What does it mean to assess CT?

....

....

....

References:

Ahmad, M., Karim, A. A., Din, R., & Albakri, I. S. M. A. (2013). Assessing ICT competencies among postgraduate students based on the 21st century ICT competency model. Asian Social Science, 9(16), 32e39.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K- 6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Journal of Educational Technology and Society, 19(3), 47-57. http://A K-6 Computational Thinking Curriculum Framework

Anuar, N. H., Mohamad, F. S., & Minoi, J.-L. (2021). Art-Integration in Computational Thinking as an Unplugged Pedagogical Approach at A Rural Sarawak Primary School. International Journal of Academic Research in Business and Social Sciences, 11(14), 21–39.

Allan, W., Coulter, B., Denner, J., Erickson, J., Lee, I., Malyn-Smith, J., & Martin, F. (2010). Computational thinking for youth. White Paper for the ITEST Small Working Group on Computational Thinking (CT).

Borup, J., Graham, C. R., & Drysdale, J. S. (2014). The nature of teacher engagement at an online high school. British Journal of Educational Technology, 45(5), 793–806. https://doi.org/10.1111/bjet.12089

Borup, J., West, R. E., Graham, C. R., & Davies, R. S. (2014). The Adolescent Community of Engagement: A framework for research on adolescent online learning. Journal of Technology and Teacher Education, 22(1), 107–129.

Borup, J., & Stevens, M. A. (2016). Parents' perceptions of teacher support at a cyber charter high school. Journal of Online Learning Research, 2(3), 227–246.

Borup, J., & Stevens, M. A. (2017). Using student voice to examine teacher practices at a cyber charter high school. British Journal of Educational Technology, 48, 1119-1130. https://doi.org/10.1111/bjet.12541

Bosch, C. (2016). Promoting Self-Directed Learning through the Implementation of Cooperative Learning in a Higher Education Blended Learning Environment. Johannesburg, SA: Doctoral dissertation at North-West University.

Branch, R. M., & Kopcha, T. J. (2014). Instructional design models. In Handbook of research on educational communications and technology (pp. 77-87). Springer, New York, NY.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at the American Educational Research Association. Canada: British Columbia.

Carnevale, A. P., & Smith, N. (2013). Workplace basics: The skills employees need and employers want. Human Resource Development International, 16(5), 491e501. http://dx.doi.org/10.1080/13678868.2013.821267

Cabo, C., & Lansiquot, R. (2016). Integrating creative writing and computational thinking to develop interdisciplinary connections.

Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why

Curzon, P., Black, J., Meagher, L. R., & McOwan, P. (2009). cs4fn. org: Enthusing students about Computer Science. Proceedings of Informatics Education Europe IV, 73-80.

Fredericksen, E. (February 4, 2015). Is online education good or bad? And is this really the right question? The Conversation. Retrieved from: https://theconversation.com/is-onlineeducation-good-or-bad-and-is-this-really-the-right-question-35949

Guzdial, M. (2008). Paving the way for computational thinking. Communications of the ACM, 51(8), 25-27.

How Online Education Aids Distance Learning. (2022, January 27). Infinity Learn. https://infinitylearn.com/surge/blog/general/how-online-education-aids-distance learning/

How Computational Thinking Sets Kids Up for Success In and Outside of the Classroom. (2021, June 14). CodeWizardsHQ. https://www.codewizardshq.com/computational-thinking/

Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., & Basawapatna, A. (2011). Computational Thinking Patterns. Online Submission.

ISTE. (2016). ISTE (International Society for Technology in Education) Standards for Students. Eugene, OR.

ISTE, CSTA. (2011). Computational Thinking in K–12 Education Leadership Toolkit.

Jensen, L. 2001. "Planning Lessons." In Teaching English as a Second or Foreign Language, edited by M.M. Celce. 3rd ed, pp. 403–409. Boston: MA: Heinle & Heinle.

Koc, S., & Goodell, J. E. (2018). Preparing teachers for mobile learning applications grounded in research and pedagogical frameworks. In Handbook of Research on Mobile Devices and Smart Gadgets in K-12 Education (pp. 103-114). IGI Global.

Kola, M. (2021). Pre-service teachers' action research: technology education lesson planning in a South African University. Educational Action Research, 29(1), 99-117.

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. Digital Experiences in Mathematics Education, 3(2), 154-171.

Lehrer, R., Schauble, L., and Lucas, D., (2008). Supporting development of the epistemology of inquiry. Cognitive Development, 23 (4), 512-529.

Li, W., and W. Zou. 2017. "A Study of EFT teacher expertise in lesson planning." Teaching and Teacher Education 66: 231–241.

Mills, K., Coenraad, M., Ruiz, P., Burke, Q., & Weisgrau, J. (2021). Computational Thinking for an Inclusive World: A Resource for Educators to Learn and Lead. Digital Promise.

Mohaghegh, D. M., & McCauley, M. (2016). Computational thinking: The skill set of the 21st century.

Nesari, A.J., and M. Heidari. 2014. "The Important Role of Lesson Plan on Educational Achievement of Iranian EFL Teacher Attitudes." International Journal of Foreign Language Teaching & Research 3 (5): 25–31.

NRC 2011. Report of a Workshop on the Pedagogical Aspects of Computational Thinking. 2011, National Research Council.

Sengupta, P., Kinnebrew, J., Biswas, G., Clark, D. (2012). Integrating Computational Thinking With K-12 Science Education - A Theoretical Framework. 4th International Conference On Computer Supported Education (CSEDU), 40-49.

Stewart, I. and Golubitsky, M. (1992). Fearful Symmetry: Is God a Geometer?, Blackwell Publishers, Oxford.

Van Laar, E., Van Deursen, A. J., Van Dijk, J. A., & De Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. Computers in human behaviour, 72, 577-588.

Voogt, J., & Roblin, N. P. (2012). A comparative analysis of international frameworks
for 21st century competencies: Implications for national curriculum policies. Journal of Curriculum Studies, 44(3), 299e321. http://dx.doi.org/10.1080/00220272.2012.668938.

Smith, J. (2021, June 14) How Computational Thinking Sets Kids Up for Success In and Outside of the Classroom. From: CodeWizardsHQ. https://www.codewizardshq.com/computational-thinking/
Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

3.1 Current Pedagogical Approaches and Strategies to Teach Computational Thinking

Computational thinking is one of the most important skill sets educators should teach their young learners. It's a cornerstone of early childhood development, giving kids an approach to problem-solving that develops a solution and is seen as one the critical competence to adapt to the future (Hsu et al., 2018). In the last decade, CT has obtained popularity, both in the academic field as well as in practice. CT has been studied in different countries worldwide. Consequently, the subjects, research issues and teaching tools in the field of CT have been further developed (Hsu et al., 2018).

Nevertheless, despite the growing academic attention to the topic, the reality in many schools is that teachers are not specialised in applying CT in their classrooms. This is mainly due to the lack of teacher´s training in CT and lack of knowledge about how to implement the methodology because there is still limited evidence around the several challenges someone needs to be aware of to design appropriate learning experiences and a curriculum based on CT (Angeli & Giannakos 2020). Therefore, there is an increasing demand for supportive material that helps teachers to teach their students different subjects based on the CT framework. This section aims to present current pedagogical approaches and strategies to teach CT. CT is still mainly used in STEAM subjects but the aim of the project is to extend CT to other subjects and thus make CT accessible to more teachers and students and especially due to its adaptation to online education.

The aim is to give an overview of the currently available strategies and methodologies that enable the CT in education. Since the focus of our approach is to offer a framework for the implementation of CT in online education, chapters 3.1.1 and 3.1.2 will present approaches to combine these CT and online courses. Therefore, the structure of this section is the following:

First, the Engineering Design Process Model in CT education will be explained. Later, the Flipped Learning Methodology will be described.

3.1.1 Engineering Design Process Model in CT education

The Massachusetts Curriculum Framework for Science and Technology (2016) is a document that addresses the mandatory educational guidelines for Science and technology/engineering (STE) subjects in this state. It includes the vision of Science and Technology/Engineering Education from their authors - a teacher and policymakers Review Panel and the Next Generation Science Standards (NGSS) Advisory Group - as well as an overview of the practices from each grade and curriculum/standards-related appendices and it highlights the importance of mathematics for measuring, predicting and solving problems.

This framework comes from the need to educate citizens capable of participating actively in a technological world, as the goal of STE education is *to develop scientifically and technologically literate citizens who can solve complex, multidisciplinary problems and apply analytical reasoning and innovative thinking to real-world applications needed for civic participation, college preparation, and career readiness*.

Teaching and learning are at the heart of quality science and technology/engineering education. The vision of the Massachusetts STE standards is to engage students in the core ideas through the integration of science and engineering practices, while making connections to what they know and the world they live in. The goal of the Guiding Principles is to help educators create relevant, rigorous, and coherent STE programs that support student engagement, curiosity, analytical thinking, and excitement for learning over time. Educators, administrators, and curriculum designers can refer to the Guiding Principles to develop effective pre-K–12 STE programs.

Emphasis in STE Standards	Implication for Curriculum and Instruction

Relevance: Organised around core explanatory ideas that explain the world around us	The goal of teaching focuses on students analysing and explaining phenomena and experience
Rigour: Central role for science and engineering practices with concepts	Inquiry- and design-based learning involves regular engagement with practices to build, use and apply knowledge
Coherence: Ideas and practices build over time and among disciplines	Teaching involves building a coherent storyline over time and among disciplines

Figure 1. DOE (2016). Qualities of Science and Technology/Engineering Education for All Students.

The guiding principles of the Massachusetts Curriculum Framework for Science and Technology are:

Relevance

1. An effective science and technology/engineering program develops students' ability to apply their knowledge and skills to analyse and explain the world around them
2. An effective science and technology/engineering program addresses students' prior knowledge and preconceptions.

These ideas focus on the importance that students have of getting to know surroundings for interesting connections and work on previous misconceptions going to the core of concepts.

Rigour

3. Investigation, experimentation, design, and analytical problem solving are central to an effective science and technology/engineering program.
4. An effective science and technology/engineering program provides opportunities for students to collaborate in scientific and technological endeavours and communicate their ideas.
5. An effective science and technology/engineering program conveys high academic expectations for all students.

This vision tries to involve students in the whole process for engagement and success by active collaboration with professionals and working on activities and concepts that will be important after school life.

Coherence

6. An effective science and technology/engineering program integrates STE learning with mathematics and disciplinary literacy.
7. An effective science and technology/engineering program uses regular assessment to inform student learning, guide instruction, and evaluate student progress.
8. An effective science and technology/engineering program engages all students, pre-K through grade 12.
9. An effective science and technology/engineering program requires coherent districtwide planning and ongoing support for implementation.

The authors manifest the idea that, if all these points are respected, the engagement of students will increase and step forwards towards a better citizenship.

The framework is based on learning standards, which are outcomes, or goals, which reflect what a student should know and be able to do. Every subject in every grade has standards that all students have to meet, for example:

Example: Grade 1: Earth and Space Sciences

Subject: Earth's Place in the Universe

1. Standard: Use observations of the Sun, Moon, and stars to describe that each appears to rise in one part of the sky, appears to move across the sky, and appears to set.
2. Standard: Analyse provided data to identify relationships among seasonal patterns of change, including relative sunrise and sunset time changes, seasonal temperature and rainfall or snowfall patterns, and seasonal changes to the environment.

Clarification Statement:

Examples of seasonal changes to the environment can include foliage changes, bird migration, and differences in amount of insect activity.

3.1.2. The 5E-Based Flipped Classroom Model in CT education

During the last ten years, flipped methodology has grown to become one of the major trends in innovative education. The flipped learning model promotes student-centred learning which is the process of using different instructional methods to target individual needs, motivations, engagement, and supports that will allow a student to succeed on their own (Carnevale, 2017). This new way of teaching has been defined by their main ambassadors,
Jonathan Bergmann and Aaron Sams (2014), as:

Basically, the concept of a flipped class is this: that which is traditionally done in class is now done at home, and that which is traditionally done as homework is now completed in class.

Nevertheless, the idea of flipped lessons goes much further and deeper, as they have been able to test later on:

Flipped Learning is a pedagogical approach in which direct instruction moves from the group learning space to the individual learning space, and the resulting group space is transformed into a dynamic, interactive learning environment where the educator guides students as they apply concepts and engage creatively in the subject matter. (Flipped Learning Network, 2014)

Flipped methodology is based on the four pillars:

F Flexible Environment

Flipped Learning allows for a variety of learning modes; educators often physically rearrange their learning spaces to accommodate a lesson or unit, to support either group work or independent study. They create flexible spaces in which students choose when and where they learn. Furthermore, educators who flip their classes are flexible in their expectations of student timelines for learning and in their assessments of student learning.

F.1	☐ I establish spaces and time frames that permit students to interact and reflect on their learning as needed.
F.2	☐ I continually observe and monitor students to make adjustments as appropriate.
F.3	☐ I provide students with different ways to learn content and demonstrate mastery.

L Learning Culture

In the traditional teacher-centered model, the teacher is the primary source of information. By contrast, the Flipped Learning model deliberately shifts instruction to a learner-centered approach, where in-class time is dedicated to exploring topics in greater depth and creating rich learning opportunities. As a result, students are actively involved in knowledge construction as they participate in and evaluate their learning in a manner that is personally meaningful.

L.1	☐ I give students opportunities to engage in meaningful activities without the teacher being central.
L.2	☐ I scaffold these activities and make them accessible to all students through differentiation and feedback.

I Intentional Content

Flipped Learning Educators continually think about how they can use the Flipped Learning model to help students develop conceptual understanding, as well as procedural fluency. They determine what they need to teach and what materials students should explore on their own. Educators use Intentional Content to maximize classroom time in order to adopt methods of student-centered, active learning strategies, depending on grade level and subject matter.

I.1	☐ I prioritize concepts used in direct instruction for learners to access on their own.
I.2	☐ I create and/or curate relevant content (typically videos) for my students.
I.3	☐ I differentiate to make content accessible and relevant to all students.

P Professional Educator

The role of a Professional Educator is even more important, and often more demanding, in a Flipped Classroom than in a traditional one. During class time, they continually observe their students, providing them with feedback relevant in the moment, and assessing their work. Professional Educators are reflective in their practice, connect with each other to improve their instruction, accept constructive criticism, and tolerate controlled chaos in their classrooms. While Professional Educators take on less visibly prominent roles in a flipped classroom, they remain the essential ingredient that enables Flipped Learning to occur.

P.1	☐ I make myself available to all students for individual, small group, and class feedback in real time as needed.
P.2	☐ I conduct ongoing formative assessments during class time through observation and by recording data to inform future instruction.
P.3	☐ I collaborate and reflect with other educators and take responsibility for transforming my practice.

Figure 2. Flipped Learning Network (FLN). (2014) The Four Pillars of F-L-I-P™

As addressed by both teachers and researchers, this model increases active participation, engagement, interaction and autonomy in students, who enjoy a more personalised education and the possibility of dedicating as much time as they need for understanding new concepts.

This approach turns especially interesting when combined with a step by step inquiry based model. The 5 Es method gives students a way to connect scientific ideas to their experiences and apply their learning. These are the 5 Es:

Engage

The teacher uses short activities to promote curiosity. The activity must connect prior knowledge to new learning experiences in order to expose any misconceptions and prepare students for new learning.

Explore

A lab investigation or hands-on activities are usually introduced in this phase as students attempt to investigate a problem. Conflicting ideas, questions, and confusion are common and help students identify what they need to know before new terms or concepts are introduced in the Explain phase.

Explain

With the teacher's guidance, students explain the concepts they explored in the previous phase and demonstrate their understanding of the new terms that were introduced. Depending on the topic and the grade level, teacher-led instruction might be necessary to address any confusion and questions that come up in the Explore phase. Questions can make learning more meaningful, interactive, and participatory.

Elaborate

Students apply their knowledge to new experiences and extend their conceptual understanding as they solve a problem in a new context before evaluation in the last phase of the 5E model. Elaboration activities can take place during classroom time, or they can be a homework assignment.

Evaluate

Students evaluate their learning and demonstrate their understanding and mastery of key concepts. Evaluation doesn't have to be limited to a quiz or test. It can be a product such as a presentation, a poster, a pamphlet, a journal article, or a final paper.

So, what are the outcomes of combining these two approaches? Goa and Foon Hew (2022) have conducted a study based on the 5-E framework in a flipped classroom questioning how well elementary students can understand CT concepts. They tested 125 students in the experimental group and 122 students in the control group and it resulted that the students that joined the class based on the 5-E Flipped classroom model significantly improved their understanding of CT concepts and problem-solving compared with the other group that joined a traditional classroom approach. To be more concrete, Goa and Foon Hew stress in their study that the 5-E framework combined with the flipped methodology helps to foster active student participation, for example: The elaboration phase during the in-class activities helps students to find solutions on their own. "Thinking about a task without immediate answers from the instructor can lead to deeper student cognitive processing, which in turn can help students to understand the content better." (Deslauriers et al., 2019 as cited in Goa and Foon Hew, 2022). Moreover the state that the flipped group was in general more active and explored more solutions to the problems presented.

All in all, it shows that combining these both models helps students to get a deeper understanding of the concepts of CT and it helps them to actively practice CT in class and enhances their learning experience.

3.2 Current pedagogical approaches and strategies for online teaching and learning

The situation caused by the outbreak of COVID-19 forced teachers to adapt to the online learning environment. More teachers than ever had to look for a way to efficiently use online tools in order to carry on their work. Many teachers were not prepared for this situation and had to start teaching without appropriate skills.

Although the situation was a challenge, it was also an opportunity for teachers to find the vast amount of online resources created in recent years with the purpose of making the teaching process easier and more entertaining.

The pandemic not only forced teachers out of physical classrooms but also brought researchers' attention to concepts related to the XXI century competencies such as digital literacy, ICT literacy and Computational Thinking. Teachers who were forced to overcome barriers caused by a lack of experience in the online teaching environment are now more than ever motivated to learn how to provide good quality training for students with the use of online tools.

After the literature review, we realise the important role of CT in the future development of social, economic, and technological systems. Change is often a complex process with challenges to overcome. The process of change increases opportunities for errors, it creates upheaval in the psychosocial state of people who are affected by those changes, but without change - growth is impossible.

Over the last two decades, schools, training centres and universities have begun to address the challenges and opportunities created by many online learning resources. The shift from the traditional face-to-face learning model to distance learning became more and more visible, or even impossible to ignore.

However, teachers whose teaching competencies were developed in a traditional classroom setting may find the shift to online pedagogy daunting. Online learning, especially designing full online courses, requires a coordinated effort of many players involved. That's why there's a need for teaching resources that address this issue and provide the concrete knowledge necessary to carry on the teaching process online.

Rogers' Diffusion of Innovation theory (2003), stresses that innovation is communicated through certain channels over time among members of a social system. Some innovations are successful, others are never widely accepted. According to Papert (1980) "schools cannot get ahead of society and the development of a digital literacy essentially requires time".

Modern-day teachers not only have to adapt to the dramatic changes brought by the rise of the web and mobile communication tools, they also need to meet the needs of students who use the internet regularly and effectively (Johnson et al. 2012). In order to digitise their classroom, teachers must incorporate the best traditional classroom pedagogy to address the needs of new online learning environments.

Complex social systems benefit from a clear change-implementation strategy, and CT can offer a necessary method for developing such a strategy. According to Wing (2006) CT involves "taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing".

CT offers a construct for solving problems related to social change through the use of algorithms, patterns, parallels, and abstraction (Voskoglou, Buckley 2012). By providing a clear articulation of the problems affecting change from one mode of pedagogy to another, CT can help educators build strategies for overcoming barriers to change.

When discussing CT teaching strategies, educational experts often propose activities based on constructivism perspective. Constructivism is the cognitive theory invented by Jean Piaget, it differs from traditional teaching approaches in which the teaching process happens through reading, listening or memorising certain knowledge.

(Social Constructivism & Social Media | Socialmedia4444, n.d.)

Piaget believed that to learn, a child must consciously make an effort to derive meaning from the concept, and through that effort, meaning is constructed. He liked to emphasise learning through play, as constructivism’s main principle is: "you have to make it to learn it". In the context of teaching CT students are often asked to create or play video-games or create and play with interactive models (Beheshti, 2017). CT teaching strategies will be described in the next section of this chapter.

3.2.1 Online Teaching Tools

Through this subchapter, we are focusing on the current pedagogical approaches to teaching Computational Thinking online. After the literature review, we categorised the following categories of tool types: visual and other programming languages, video games and collaboration tools.

Visual Programming Language (VPL) use in CT learning process

In computing, a visual programming language (visual programming system, VPL, or, VPS) is any programming language that provides graphical or iconic elements which can be manipulated by users in an interactive way. In other words, any language that lets users create programs by manipulating program elements graphically rather than by specifying them textually. Because VPL languages don't require a specific coding knowledge, they're the most popular approach to teach programming and CT skills in school. Writing code is in general a very creative procedure and many studies have shown that students who experienced code writing improve their problem-solving skills and system design skills (Krugel, Hubwieser, 2017).

A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notation. For example, many VPLs (known as dataflow or diagrammatic programming) are based on the idea of "boxes and arrows", where boxes or other screen objects are treated as entities, connected by arrows, lines, or arcs which represent relations.

However, when using a visual programming language, we need to think about how to introduce it to students so that they can use it effectively and, above all, want to develop their skills in using it.

Individual effort

One method is to encourage students to work individually. In this case, the student is given a task to perform or a problem to solve and works on it. For example, Ahmadi and Jazayeri (2014), tried to see if a complete novice could independently learn CT skills by asking participants of their study to explore the proposed problem and then gain the programming skills that are required to solve it. In result - 90% of tasks were completed in full by the participating students.

Collaboration and remixing

Remixing has been defined as the reworking and combination of existing creative artefacts, usually in the form of music, video, and other interactive media. Lessig (2008) has suggested that remixing reflects a broad cultural shift spurred by the Internet and a source of enormous creative potential. Manovich (2020) has called remixing "a built-in feature of the digital networked media universe.

The importance of remixing and reusing has been recognized in the sphere of developing Computational Thinking skills. Students - other than creating their own code - can also remix a code created by someone else. In the context of CT:

- **Reusing** means taking pieces of code created by others and using it to solve a problem or meet a need, rather than creating it from scratch.
- **Remixing** involves putting together or 'mashing up' code for video, sound, text, etc. created by other people to make something new and original.

An example is a Scratch project that has been created by a pair of people, and then passed on to another pair to extend and reimagine.

Another example is given by Repenning et al. (2009). The authors of this study believed that programming is taught today inefficiently and tends to discourage students' peer-to-peer interaction. In their study they employed an online homework submission system called the Scalable Game Design Arcade (SGDA). SGDA provided a supplementary way of teaching Computational Thinking using peer-to-peer interactions. Students learned from each other, by viewing and running each other's code. It was done in order to implement the Flow of Inspiration Principles in Educational Game Design.

The collaborative method they promoted focused on the Flow of Inspiration Principles, which were the result of 10 years of observation of middle school students in computer science clubs who freely shared programming ideas, codes, and advice on problem-solving solutions. The researchers' intention was to transfer that interactions present in the computer club environment to the classroom. The principles, they named, include:

- Displaying projects in a public forum
- Viewing and running fellow students' projects
- Providing feedback on fellow students' projects
- Downloading and view code for any project
- Providing motivation for students to view, download, and give feedback on fellow classmates' projects

The Scalable Game Design Arcade (SGDA) they used was based on the idea of scalable Game Design. This approach combines motivation and competency frameworks, standards for technology, and computational thinking authoring tools. It shows that designing and playing video games can be a method of teaching CT principles (Wing, 2006; Ioannidou et al., 2011). The Scalable Game Design curriculum balances design challenges and design skills to keep students in optimal flow. This can be achieved through various forms of scaffolding methods, such as explicit just-in-time instruction and social learning support from interactions with instructors and other students.

Now that we have explained what VPL is, described CT teaching methods based on solving problems independently or collaborating and remixing with stressed peer to peer interaction it is now time to move on and talk about the tools a teacher can use to teach CT.

Visual Programming Language Tools

Online Teaching Tools

Ioannidou et al., (2011) claim that Computational Thinking Authoring Tools are the kinds of tools that are essential to enable children to acquire Computational Thinking skills. CT authoring tools that are of high quality and enable CT learning:

have a low threshold: a student can produce a working game quickly.

have high ceiling: a student can make a real game that is playable and exhibits sophisticated behaviour, e.g., complex AI.

have scaffolding flow: the tool + curriculum provides stepping stones with managed skills and challenges to accompany the tool.

enable transfer: tool + curriculum must work for both game design and subsequent computational science applications as well as support transfer between them. It should also facilitate transition to professional programming languages.

support equity: game design activities should be accessible and motivational across gender and ethnicity boundaries.

are systemic and sustainable: the combination of the tool and curriculum can be used by all teachers to teach all students (e.g. include opportunities for teacher training and implementation support; align with standards and work in STEM content areas).

The following table contains a list of notable CT tools. Many of which enable game design:

Name	**Description**	**Program name/ Authors, developers/ Licence**
Kodu	- allows developing 3D games of a different type: e.g., adventure, arcade, racing - for children young as 9 to 10	Microsoft's FUSE Labs/ Microsoft Open Source License
Greenfoot	- is an integrated development environment using Java or Stride - allows developing two-dimensional graphical applications, such as simulations and interactive games	M. Kölling/ Free and open-source
NetLogo	- is a programmable modelling environment for simulating natural and social phenomena - students can open simulations and “play” with them, exploring their behaviour under various conditions - it enables students, teachers and curriculum developers to create their own models	U. Wilensky, Northwestern University/Free and open-source

Scratch	- allows developing interactive stories, animations, and simulations - developing 2D games of a different type: clicker games, platform games, maze game, pong game -the games developed with Scratch can be played locally or online once the games have been uploaded on the Scratch website - recommended for ages 8 to 16	MIT Media Lab/Free and open-source
ScratchJr	- is an interpretation of Scratch designed primarily for younger audiences (5-7-year-old children).	MIT Media Lab/Free and open-source
AgentCubes	- it's an educational programming language - allows creating 3D and 2D online games and simulations - recommended for ages 10-12	A. Repenning/ Proprietary licence
Snap	- D&D interface - snap allows you to create games, animations, apps, presentations, etc. - recommended for ages 12-20	B. Harvey and J. Mönig/Free and open-source
App Inventor	- uses a visual, blocks language - allows building Android Apps - recommended for ages 9 - 12	H. Abelson, M. Friedman, MIT Media Lab / Free and open source
VTS Editor	- VTS Editor allows users to develop simulation games	Serious Factory/Paid with free trial

AgentSheets	- block based programming environment - allows developing 3D games and publishing them on the web - for children, age not specified	A. Repenning/ Proprietary licence
Alice	- block based programming environment - allows developing 3D games and publishing them on the web - recommended for ages 13 to 18	Saarland University/ MIT Licence
GameMaker	- uses GameMaker Language - D&D interface - allows developing 2D games of a different type, for example: shoot them up games - recommended for ages 11-to-14	M. Overmars, YoYo Games/ Proprietary licence
Bubble	- used to create websites and web applications	J. Haas E. Straschnov/ Commercial
GDevelop	- uses JavaScript - 2D game development engine - user can develop all kinds of games, For example, platform games, puzzles, bullet shooter games - recommended for ages 7-to-19	F. Rival, V. Levasseur, A. Vivet, A. Pacaud, Franco M., T. Imreorov /Free and open-source

Godot	- uses GDScript, C#, VisualScript, and C++ and C - an incredibly versatile engine for 2d and 3d games	J. Linietsky, A. Manzur/Free and open-source
Celestory	- D&D interface - allows developing 2D games of a different type: Playing cards, escape game, platform, interactive, movie, quiz	Celestory 2022/freemium
Catrobat	- a block-based visual programming language - allows the creation of games, stories, animations, and many types of other apps directly on smartphones	W. Slany/Free and open-source
GameSalad Creator	- D&D interface - allows developing 2D games of a different type like puzzle or arcade -enables the user to create applications for iOS, Android, HTML5, and the Mac Platform - recommended for ages 12 and beyond	GameSalad/Proprietar y licence
Gamestar Mechanic	- point and click interface - allows developing 2D games of a different type: adventure, platform, action, and experimental - developed game uploaded in the Game Alley and shared with the online community - designed especially for ages 14 to 17	E-Line Media/ Free and open-source

Stagecast Creator	- point, and click interface. - allows developing 2D games of a different type: action and adventure games. - allows users to save the games developed on a local disk or on the internet. - designed for children as young as 8	A. Cypher and D. Canfield Smith, ATG/Freeware
Unity	- uses C# language - allows users to design both 2D and 3D games -different type of games can be created: First person, Flying, Puzzle, Rolling	Unity Technologies/Propriet ary licence
Blender Game Engine	- python as a programming language - 3D creation suite	Blender Foundation/Free and open-source
Babylon.js	- uses JavaScript; TypeScript - it's a 3D engine for games and other 3D visualisations. It allows building animated 3D computer graphics	D. Catuhe, Microsoft and contributors/Free and open-source
Construct/ Construct 2	- HTML5-based - allows developing 2D games	Scirra/Proprietary licence
RPG Maker VX Ace Lite	- uses Ruby language - it's a game engine designed to make 2D Roleplaying Games	Enterbrain Inc./Free and paid version

Flowgorithm	- uses C# language - tool which allows users to write and execute programs using flowcharts - can interactively translate flowchart programs into source code written in other programming languages	D. Cook / Freeware
Hopscotch	- drag-and-drop interface - allows to create games, animations and other colourful interactive programs - designed for ages 10-16	Hopscotch Technologies 2021 / Free to download app. Paid upgraded version.
Kojo	- a point-and-click interface - allows creating drawings, animations, games - designed for children as young as 8	Lalit Pant / Free and open-source
Mblock	- based on MIT's visual programming language - drag and drop interface - allows creating games and animations - designed for children as young as 8	Ja. Wang / Free and open-source
Open Roberta	- enables children and adolescents to program robots - designed for children as young as 8	B. Jost, R. Budde, T. Leimbach, A. Kapusta. Fraunhofer IAIS / Free and open-source

Raptor	- designed specifically to help students visualise their algorithms	M. C. Carlisle, T. Wilson, J. Humphries and Jason Moore/ Free and open-source
Starlogo	- use 3-D graphics to make games and simulation models	M. Resnick, E. Klopfer, D. Wendel, MIT / Free to use non-commercially
ToonTalk	- allows building programs and games - designed for children as young as 5 or 6	K. Kahn / Versions 1.0 and 2.0 had commercial licenses version 3.0 is now free.
Visual Logic	- allows students to write and execute programs using flowcharts	© PGS Systems/ Proprietary licence

Playing Video Games

The use of VPL is a good CT teaching method because it enables students to construct artefacts such as video games or interactive stories. But we also have other methods that are worthy of consideration (Weintrop et al. 2016).

Many studies demonstrate how simply playing a video game can be an effective way to practice CT. This method may be especially attractive for out-of-classroom learning experiences. The right method of teaching CT will always be determined by a variety of factors i.e. students' age, teachers' preferences, time constraints, etc.

The majority of the games are designed with a "constructionist" activity as core to the gameplay. The other theory to note is Computational Encoding (Holbert & Wilensky, 2011), and there are also various techniques of learning

CT such as collaboration remixing, and immersion (Pellas and Peroutseas 2016; Dhatsuwan and Precharattana 2016; Debabi & Bensebaa, 2016).

Below is a list of the most well-known and free video games that transparently help users to enhance CT principles.

- **AlgoGame:** Futschek (2006), proposed the AlgoGame as a method of teaching CT. Students were asked to focus on solving problems rather than learning the syntax of a programming language. Futschek reported that after playing the AlgoGame, an experimental group had better results in writing a selection sort algorithm than a control group.
- **CodeCombat:** The game called CodeCombat offers an ideal platform for teaching object-oriented programming (OOP) concepts. Teaching OOP helps to prepare students to write their first program by using classes and objects. CodeCombat has multiple levels, and each level covers several concepts of the computer sciences curriculum. For instance, the first level covers introductory concepts such as syntax, sequence, objects, and methods. The game illustrates objects as building blocks, they are things or characters that can perform actions. In the game, a hero is an object. The actions an object performs are called methods. This is the way in which students have to interact with the coding blocks (Karram, 2021).
- **Rob-Bot:** is a computational thinking card game consisting of a pack of 57 playing cards, each of which contains 8 different characters or objects. Each card contains one character or object which matches to one in every other card in the pack. The authors claim that the game develops CT thinking because it requires identifying specific characters and objects on the cards (decomposition), searching for and finding the same character or object (pattern recognition), abstraction through dismissing and filtering out characters and objects that don't match and finally - algorithm design - meaning a technique of searching for and identifying characters and objects (Rob-Bot Resources, 2018).

3.2.2 CT in online education

So far, we have presented a consistent and comprehensive pedagogical approach to using methods and tools for online education. This section is dedicated to the role of online teaching methods in order to boost and support Computational Thinking development.

According to Voskoglou and Buckley (2012), *"CT develops a variety of skills (logic, creativity, algorithmic thinking, modelling/simulations), involves the use of scientific methodologies and helps to develop both inventiveness and innovative thinking"*.

Computational Thinking offers teachers not only the opportunity to explore new problem-solving processes but also to unleash their own creativity in designing courses, research and policies within their areas of specialisation. One way to offer teachers a chance to learn Computational Thinking would be to provide an opportunity to undertake online course development, using Computational Thinking as a model. Teachers would be able to choose any subject of interest; participatory design and development of online course content would both promote an understanding of new course technology and allow teachers to apply their expertise in course content areas.

Teachers can find one of the models in Soh et al. (2015), who created a series of courses that supported both analysis (decomposition, pattern recognition) and reflection (abstraction, algorithm design), and used simple description exercises to promote collaborative problem-solving. Smidt et al. (2014) also offered a possible framework for development starting with a face-to-face conversation and then transferring communications to an online discussion board. Masterman, Walker, & Bower, (2013) aimed to promote opportunities in using technology tools to guide and support pedagogy while also promoting collaborative teachers learning in designing online courses. They concluded *"[the] challenge to embedding computational support for teachers' thinking in a manner that takes into account all these factors are to position it within the design of a program, department, and faculty where it is used by school on a regular basis."*.

In practice, there are two barriers to integrating Computational Thinking into the curriculum

- the lack of consensus regarding a definition of Computational Thinking;
- the shortage of qualified teachers that can teach this skill (Chi and Menekse 2015; Curzon and Dorling 2014).

This lack of agreement was highlighted in 2009 when a workshop organised by the US National Research Council with the goal of establishing *"The Scope and Nature of Computational Thinking"* failed to reach a consensus among its participants concerning the content and structure of Computational Thinking (Committee for the Workshops on Computational Thinking et al. 2010). Computational Thinking has its beginning with Seymour Papert (1980) and his much cited book *"Mindstorms: Children, Computers, and Powerful Ideas"* but it was Jeannette Wing (2006), who popularised this concept. She defined it as follows: *"Computational thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science"*. This definition of Wing's has significance for compulsory education in that it states that Computational Thinking is fundamentally a thought process, i.e., independent from technology and that its solutions can be executed by either a human or computer, or both (European Commission. Joint Research Centre. 2016).

Wing's 2006 paper was very specific about this stating that Computational Thinking was about conceptualization, not programming. It was concerned with ideas, not artefacts.

However, this is not a universal belief. For example, (Brennan and Resnick (2012) proposed a definition of Computational Thinking that revolved around the Scratch language. At the aforementioned workshop organised by the US National Research Council in 2009 (Brennan and Resnick 2012; Wolz et al. 2010) all voiced opinions concurring that programming is essential to Computational Thinking (Committee for the Workshops on Computational Thinking et al. 2010).

The above discussion highlights that there is no simple answer to the question of what is Computational Thinking, thus illustrating one of the challenges to integrating Computational Thinking into the classroom and online environment.

3.3 Computational Thinking: Teaching methods

3.3.1 Programming and mathematical computing training as a Computational Thinking teaching methods

Thinking is a human brain's indirect and abstract reflection of the essence of objects, their inner connections and it is also the main form of human intellectual activities. How to explicitly express the thinking activities internalised in the human brain and to make learners see and touch the abstract computer knowledge with non-physical properties is a very important question.

CT is a thinking method and while teaching CT teachers' basic goal should be to develop learners' thinking abilities to solve real problems by applying CT consciously. Edsger Dijkstra, one of the famous computer experts who published the book "The Teaching of Programming i.e., the Teaching of Thinking" in 1976, said: "*the tools we used had affected the ways and habits of our thinking, thus they will also profoundly affect our thinking ability"*. Therefore, CT and programming are closely related concepts. CT provides some efficient thinking methods to programming courses and programming courses provide a practical carrier for the cultivation of CT skills.The programming course's goal is to make learners understand how to solve real problems with the use of computers, which is the embodiment of CT training. CT is implemented by automatic and mechanical calculation and the contents of a programming and/or mathematical course can include, for example: abstracting problems, decomposing problems, creating models, designing algorithms, and verifying results.

Thanks to the problem-driven teaching method, teaching activities include finding a real problem, mapping solving methods into a computer model and designing algorithms.

In traditional programming courses, students are being taught how to program with a specific language. Applied methods lack universality, the ability to program in one language doesn't transfer to being able to program in another. In consequence, teaching effects were very different from the educational goals

As we know, the programming languages are different from each other, but the problem-solving and thinking methods are the same or similar. The introduction of CT forces students to pay more attention to the ideas behind the teaching and methods of thinking.

The teaching methods based on CT can help students to understand the collaborative relationship between human thinking and computing, it can also make them master the general methods of thinking summarised by CT. CT cultivates the learners' abilities to find problems, analyse problems, design algorithms, compare performance and solve problems by applying specific methods and principles.

Students can take what they learned during the programming course and apply computational thinking skills to different areas of life. Therefore, a programming course can most and best reflect the thinking process of CT.

1. Is programming the most appropriate way for expressing CT?

CT is described through some languages or words. And it is meaningless without explicit expression. Further, the expression of CT must follow some strict rules and specific language formats, otherwise, it cannot be understood. Programming language itself is a kind of formal expression with the characteristics of determination, finiteness and mechanisation. Programming language can express CT accurately and the expression can be understood by computer easily.

2. Does programming teaching include the thinking methods of CT?

Programming course converts the unknown problems into the known problems by applying CT's methods of reduction, transformation, simulation, etc. Programming course simplifies the complex problems into simple ones by applying structural programming and functions based on the CT's Separation of Concerns approach. Programming course solves the uncertainty problems based on the CT's heuristic approach. Programing course improves the calculation speed by using CT's parallel approach and the evaluation of time and space proposed by CT is also the important index for measuring the performance of algorithms; Programing course focuses on the robustness and reliability and algorithms can be recovered in worst-cases through CT's methods of prevention, protection, redundancy, fault-tolerant, damage containment and error-correction, etc.

3. Is the practice in a programming course an important way for training CT ability?

The cultivation of CT skills requires experience. Students need to be able to test the concepts they learned about and see if they can apply them in practice.

In the case of CT learning, the experience can be in the form of writing a computer program or designing a game.

Through lots of practice, programing courses can transform superficial and abstract knowledge into applying in practice the appropriate CT methods and effective computer knowledge. In conclusion, with the application of CT, the programming knowledge ran throughout the whole process of solving problems for the ultimate goal of forming a complicated thinking network, which took the knowledge as nodes and the CT as connectors. CT makes the contents of the programing course approach the real problem as far as possible and it expands human thinking and computer knowledge from computer to the real world by breaking the limitations between humans, the real world and computer science. CT and programming have a natural relationship, which indicates the cultivation of CT ability needs the support of programming and programming needs the guidance of CT. Therefore, it makes sense to introduce CT into programming. It was necessary to teach knowledge along with to teach its contained thinking. In this way, it can strengthen knowledge through thinking and can cultivate thinking through knowledge. Ultimately, it can both help learners learn programming languages and cultivate their CT ability.

3.3.2 Experiential learning: learning by doing in CT education

This section will present different approaches of experiential learning as teaching methods for the use of Computational Thinking in an online classroom. The aim is to present the methods that fall under the experiential learning approach in the context of the computational thinking approach and to provide practical guidance for online teaching.

Experiential learning is defined below. In the following the different methods that fall under it are presented: Project-based, Problem-based, Cooperative learning and Game-based learning. The focus lies on their connection with CT and the implementation of this combination into online teaching.

What is experiential learning (ExL)? How can it be defined?

"Experiential [learning] is a philosophy and methodology in which educators purposefully engage with students in direct experience and focused reflection in order to increase knowledge, develop skills, and clarify values" (Association for Experiential Education, para. 1).

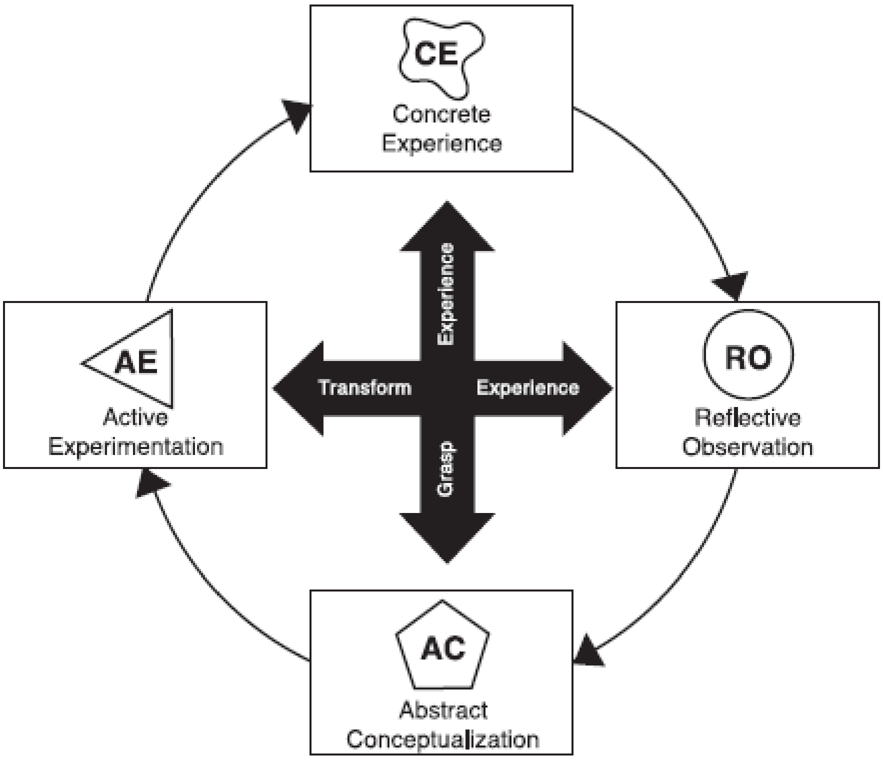
Generally speaking, ExL is a process that describes learning through experience. It is about practical hands-on learning. This means that ExL takes a different perspective on the learning process compared to didactic approaches and other learning theories. By placing experience at the centre of learning (Kolb, 2014), ExL is characterised by certain features that are contrary to traditional learning methods.

Within ExL theories there are different approaches. Different authors have attributed a central role in learning to experience. What Kolb (2014) calls the "foundational scholars of experiential learning" are the following authors: Wiliam James, John Dewey, Kurt Lewin, jean Piaget, Lev Vygotsky, Carl Jung, Mary Parker Follett, Carl Rogers and Paulo Freire.

They have all produced different emphases for the theory. Nevertheless, it has been the educational psychologists John Dewey (1859-1952), Carl Rogers (1902-1987), and David Kolb (b. 1939) that set the groundwork of ExL theories with the focus on "learning through experience or "learning by doing." In general, ExL is a dynamic, holistic theory and multi-dimensional theory. It addresses students in their entirety and are viewed as valuable resources in the educational cycle (Carver, 1996). This means they are important factors for their own educational journey as well as the one of the other students. Moreover, the educational staff is also seen as a member of this learning community. During the process the teacher constantly learns from his or her students (Carver, 1996). This means that ExL is a learner-centred approach. Due to that experience being the centre of the learning process, the traditional way of instruction is interjected. ExL approaches design instructions to engage students with the topic through direct and hands-on experiences. where the instruction is highly structured, students in experiential learning situations cooperate and learn from one another in a more semi-structured approach with the aim to find solutions for real-world problems. Also, the role of the teacher changes in the ExL approach. The teacher is seen as a facilitator and guide and not just the primary information source that gives direct instructions where the students only task is to listen and absorb the information while the teacher is explaining (Northern Illinois University, 2012).

The approach offers numerous advantages. Among other things, its proponents argue that it can lead to a better and deeper understanding of the course and the concepts. Students learn better when they apply and experience the concepts themselves. Thus, it is not the reproduced word that is stored in the short-term memory, but the information is stored in the autobiographical memory and there it is stored more sustainably. Through the process described below, students learn to reflect critically (Slavich & Zimbardo, 2021).

To apply the approach in the classroom, teachers can use Kolb´s concept of the "learning cycle". The cycle is based on four stages that divides the learning experience into the following steps: action, reflection, experience and abstraction.



This means the Kolb's learning cycle combines the grasping of experience— Concrete Experience (CE) and Abstract Conceptualization (AC)— with the transformation of these experiences into knowledge through Reflective Observation (RO) and Active Experimentation (AE).

Moreover, the experiential approach places change the interaction between students and teachers. Using Kolb's cycle all participants are included into the learning experience, students and teachers are both the receivers of information and the creators of information (Kolb, 2017). It is a student-centred and flexible approach.

How can ExL be combined with CT? Looking at the fundamental theoretical approaches, one can see how these methodologies can work hand in hand.
The main pedagogical approach guiding computational thinking is constructionism. Piaget argues that learning is an internal process, which takes place through interaction with the environment (Piaget, 1969). Hence, as opposed to seeing or listening, thinking, and reflecting improves our ability to learn. Consequently, active learning methodologies are a tool to develop

computational thinking (either one or a combination of several). Seymour Papert, the driving force behind programming for children and creator of the Logo language, and considered as the pioneer of "computational thinking", introduced the idea that programming can provide children with a way to think about their own thinking and learn about their own learning. In the context of active learning methodologies, programming becomes a tool for planning, conducting, and evaluating problems and solutions.

As previously mentioned, the academic research on CT increased in the last decade. Many authors adopted Project-Based Learning, Problem-Based Learning, Cooperative Learning, and Game-based Learning in CT. Moreover, these methodologies are all based on the ExL theory. Therefore, the following sections will be discussing first the different methodologies and approaches and their connection and implementation in CT.

3.3.2.1 Project-based learning in the CT activities

Project-based learning is a student-centred and dynamic classroom methodology. PBL is oriented towards solving specific topics or projects taken from real life, so that, as with other active methodologies, the student acquires the ability to build his or her own learning process.

The students actively engage in real-world projects. This is done to these five key features:

1. Teachers provide their students with a driving question. This question should initiate the process to solve the problem.
2. Students start to explore this driving question by participating in authentic, situated inquiry.
3. Students, teachers and community members collaborate in activities to find a solution to the problem.
4. Students will be equipped with learning technologies to help them to achieve results and participate in activities.

5. Students create a set of tangible products that address the problem (Krajcik, 2006).

The approach lies on the assumption that learners obtain deep understanding of subjects when they interact with the world and experience the topics. The approach counteracts the traditional way of teaching where information is presented and transmitted by the teachers. The advocates of this approach argue that only superficial learning can occur in these traditional ways of teaching (Krajcik, 2006).

A class that makes use of PBL should include the following steps:

1. Generate and stimulate: Teachers should discuss topics with their students and activate their curiosity.
2. Define and refine: From this point on, the teacher should define a driving question that is built on the interests and environment of his or her students. These are questions that can not be answered with a yes or no. In general, they must be formulated in a way that they initiate deeper research, curiosity and engagement.
3. Design and collaborate: At this stage the project is designed with the help of the SMART principles (Specific, Measurable, Achievable, Relevant, Timely). Different tasks will be assigned to the students. Nevertheless, there should be a strong focus on collaboration.
4. Compare and share: The project-based methodology stresses the importance of feedback. The feedback should come from the teacher and peers as well this is done by creating groups to compare and share their ideas.
5. Enhance and Advance: During the learning experience learners enhance skills and knowledge they already have.
6. Review and Revise: Students have a retrospective of what they have done. They are asked to evaluate their work.
7. Produce and present: The final product should be presented (Harding da Rosa, 2018).

As a reminder, CT is a problem-solving approach using some of the computer science techniques like breaking down big problems into smaller parts, seeking for similarities between and within problems, focusing only on important information and so on.

Integrating the Project Based Learning model in Computational Thinking involves taking a project as a reference. The student will work through a process of acquiring knowledge and thereby creating new knowledge as a result of a decision-making process within the framework of a trial/error model. This is an eminently constructionist experience in which the student interacts with the environment to tackle real-world projects.

Working on computational thinking through project-based learning will allow us to work on dimensions such as the capacity for abstraction or the identification of patterns and variables. In addition, it will allow us to express ideas and encourage creativity, developing skills for prototyping or defining the project.

As an essential part of the learning process, this knowledge must be evaluated. The teacher becomes a facilitator who guides the process by proposing different scenarios that contextualise the learning process.

How can CT be combined with Project-Based Learning in an online class? Therefore, Shin et al. (2021) propose to use the five steps presented above and adapt them to the CT context:

1. Focusing on learning goals. These goals should enable the students to demonstrate mastery of both the chosen topic and CT practice.
2. Starting with a driving question grounded in CT. This question must have an adaptation to the lives and environment of the students.
3. Exploring the driving question by participating in group works and other activities that intersect with CT (e.g., asking questions, developing and using models, analysing data, constructing an explanation and

designing a solution). It is crucial that students are able to transfer it to their everyday life.

4. Creating a set of tangible CT products.
5. Provide students with CT learning technologies.

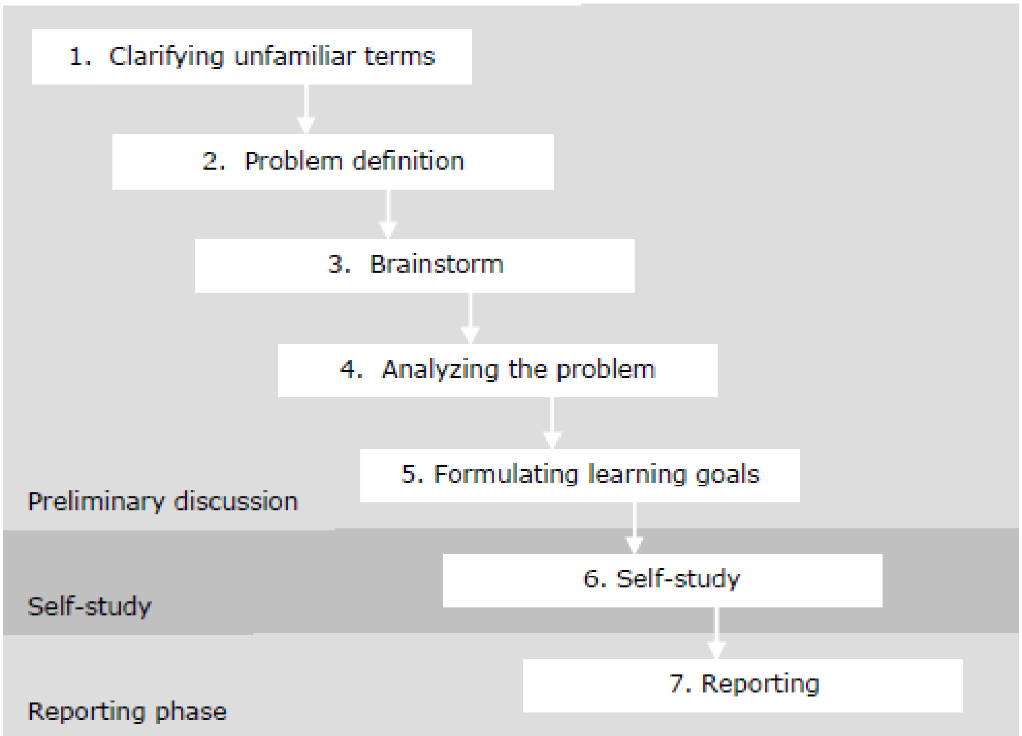
To implement a project-based approach in an online class in conjunction with CT, project management tools for team leadership can be used. Like this the teacher can monitor the process of each group. These tools can be Project Pals, Headrush or Student corner.

All in all, keep the steps of the project-based learning approach and try to design the tasks and the Guiding Question in a way that the principles of CT can be used. This means that in the second step of exploring the question, the children should follow the principles of CT: Breaking big problems into small ones, finding equalities and so on.

3.3.2.2 Problem based learning in the CT activities

Problem based learning (PBL) is a variant of project-based learning. It is based on the same principles and key features as the aforementioned. Nonetheless, there are some differences. In order to initiate a PBL class students are confronted with "triggers" from a problem case. Nevertheless, it is the teacher that presents the problem but he or she shifts from a presenter of information to a facilitator of a problem-solving process. This means, problem-based learning aims that students become self-directed learners, but teachers still guide them by monitoring discussion and intervening when appropriate (Allen et al., 2011). So, Problem-based learning is not an approach that focuses on problem solving per se, but on the increase of knowledge. The main difference is that in a problem-based learning approach, education does not take place in classrooms but in small student groups where students discuss problems using the seven-steps approach to find a solution (Wood 2003). Problem based learning has its origin in medical education. Here, the approach was invented as a tutorial process because the intensive pattern of basic science lectures followed by an exhausting clinical teaching program, was becoming ineffective (Savery, 2006). So, problem-based learning was originally an approach for medical students in universities and is based on a tutoring program where the teacher guides the students through the lecture and where students and teacher obtain a different role than in a traditional system.

Problem based learning is based on a seven-step approach:



(Camp et al., 2014)

According to these seven steps approach the teacher prepares an authentic, ill-structured and real-world problem and establishes the learning objectives. Then, the students work in small groups. They have the responsibility of their own learning experience, but they are guided by the tutor. There are different problem-based learning approaches but most of them define different roles that the students have to obtain in the sessions with the teacher. Students can obtain roles like project leader, recorder (takes notes of each meeting), team member. The first step is to clarify unfamiliar terms. The aim is that everyone is on the same page. Then the Problem has to be defined in the second step. The group has to agree upon the phenomenon that they want to explain. In the next part the group shifts to a brainstorming session where preexisting knowledge is activated and determined and then they will analyze the problem in the next step. Here, students have to discuss explanations and hypotheses. The goal of all is, that the student group formulates learning goals at the end of the preliminary discussion. From this point on, they enter the self-study phase. The group members search individually for relevant literature that can answer the questions in the learning goals and prepare reporting for the next tutorial meeting. At the end of every session, the tutor and the other group members have to give feedback about the accomplishments (Camp et al., 2014).

The advantages of the problem based learning approach are that it promotes self learning and it is highly engaging. Moreover, it helps students to develop transferable skills and improve teamwork, communication and research skills and abilities.

Using PBL for the development of computational thinking means understanding the problem, facilitating solutions and driving decision making, using a sequence of phases that would translate into algorithmic thinking. The main phases of the process would be the following:

- Breaking down the problem into parts
- Extract key information
- Develop descriptive models to understand the problem and search for a solution
- Create and test automated solutions

Automaticity will allow all knowledge to be focused on finding the solution to the problem.

Activation will allow us to identify the main elements / decompose the problem, analyse their interactions and isolate patterns of behavior of the problem.

Causality facilitates the establishment of a sense of the relationship between the knowledge we are acquiring and our previous knowledge. From there it is time to make connections between what we know and what we learn.

The **Generation of Connections**. Active methodologies allow the student, who is at the centre of the learning process, to generate the connections that arise from the analysis of the problem. The student is enabled to connect the acquired knowledge to the existing knowledge he or she gathered beforehand. We are codifying learning and the teacher, who, in a context of active methodologies, takes on the role of facilitator, supports the process by allowing it to be broken down into smaller processes and components.

Coding is understood in this context as a programming sub-task that involves the task of understanding a problem.

The teacher is the facilitator of the process, he or she decompose the problem through questioning

Students create connections

The learner performs the analysis process by generating a coding process for solving the problem through analysis.

Discovery. The teacher guides the students in the process of exploring the problem which allows them to identify relationships and patterns, discovering the rules and principles underlying the problem, allowing the discovery of patterns.

Metacognition. The aim is to reflect on the learning process. Studying the problem-solving process will allow us to share the learning process, generating codes and detecting possible errors, analysing how we have developed the process (Mayer & Wittrock 1996, 2006).

In both Problem-Based Learning and Project-Based Learning, it is the students who, normally organised in groups, set out their order of tasks, make timetable proposals, search for information and apply it in their final product, disseminating the possible solutions. All this makes use of creativity, critical thinking, and collaboration.

3.3.2.3 Cooperative learning in the CT activities

As the aforementioned, Cooperative learning is an approach which is student centred, instructor-facilitated instructional method in which students work in groups with the guidance of a teacher. It also focuses on the responsibility of the students of their own learning. Also, Cooperative learning has its origin in social constructivism that stresses that the roles of culture and society, language and interaction are important for the way individuals learn (Vygotsky cited by Li & Lam, 2013). Social-constructivism states that knowledge is cultural, and the development of every individual is a cultural process. The abilities that the individual gains are a result of social interactions with others (Li & Lam, 2013).

Cooperative learning therefore focuses on the arrangement of students into groups. While grouping the students the teacher has to focus to offer some structural elements that help the students to see the benefits of their group work (Mourtos, 1994):

Forming Teams: While grouping students into working units there are two factors that need to be considered: a. the size of the teams, and b. the members of the teams. The size of the team is a critical factor for the collaboration of the group. On the one hand, more students in the group means more information and knowledge. But on the other hand, larger groups require a lot of social skills, and it can cause problems in the interaction. Therefore, it is crucial that the teacher chooses the size of the group wisely but having in mind the characteristics of the members of the group. Teachers must know their students well in order to group the teams with a mix of abilities that the different students have to offer (Mourtos 1994).

Positive Interdependence: "For CL to work, students must understand that either they swim together, or they sink together" (Mourtos, 1994). Positive Interdependence means that students have the responsibility for their own learning but also for the development and the learning of their group mates. These two statements have to be clear to the students.

Individual Accountability: Nonetheless, the approach favours and promotes group works students have to have accountability for themselves. To avoid that any student becomes a "free rider" and avoid participating, teachers should to the following:

- If students are still no literate in cooperative skills, keep the group small
- Give individual tests
- Give individual tasks to the group members. So, they are responsible for this part, and they have to teach about this part of the project. This approach is the so-called "jigsaw" approach.
- Students should have a personal notebook (Mourtos, 1994).

As well as in the other methodologies, the role of the teachers in Cooperative Learning shifts from professor to facilitator. The teacher obtains the management of the class and the learning process, but the students are responsible for the learning outcomes. They are the sources of ideas and solutions.

How to combine cooperative learning with CT? In the implementation of a CT learning project according to the principles of collaborative learning, teachers and students work together working on the dimensions of the process, finding solutions and understanding the process. The aim is to:

- Automate solutions
- Collect information/data
- Establish patterns between data
- Using parallel thinking to make sense of the information gathered

- Modelling through data to allow experimentation through trial/error modelling
- Evaluate the process as a group in terms of effectiveness and efficiency

This can also be done online through different learning management systems and platforms that offer video conferencing tools.

3.3.2.4 Game-based learning in the CT activities

Once again, the essential elements of game-based learning will be explained before explaining how game-based learning and CT can be linked.

Game based learning is an educational method that uses certain game principles and transfers them to non-game contexts and real-life settings with the aim to engage the student (Trybus, 2015). The term was coined at the of the millennium through the work of authors such as James Paul Gee (2007), Diana Oblinger (2006), Richard Van Eck (2006), Steven Johnson (2006) and Marc Prensky (2007). But looking back into history, games have always been a part of teaching. Chess was used to teach strategic thinking. Game-based learning is one of the methodologies with the greatest impact in the current context of active learning methodologies.

The core concept of the game-based methodology is teaching through a different way of the interpretation of failure. The students work towards a goal, but they are learning and studying while experiencing the topics through games or game elements. They have to choose their actions and can repeat them as often as they want. The game-based methodology is about repetition, failure and accomplishments of goals (Cahill, n.d).

The advantages of this methodology are that it makes learning more fun, therefore the engagement of the students increases. Moreover, it offers the students the freedom to fail and experiment. Mistakes and failure are part of the learning process and are not interpreted as something bad, but much more as something necessary and as an elementary part of learning.

As there are a huge variety of games, there are also many different ways and possibilities how teachers can use games in their classroom: Board games, Card games, Video games, Simulations, Role-playing games and so on. Moreover, the methodology is especially fruitful for online classes due to the high offer on the internet. There are many different platforms that offer game-based elements. One example is Kahoot. Here teachers can prepare quizzes where students can participate with their device, and they get point for every right answer.

Obviously, the effectiveness of the game-based methodology in the classroom depends on the design of the games in order to achieve different pedagogical objectives, as well as on the different learning theories included in each of them.

How to combine CT and game-based learning? To answer this question, we will examine one of the above-mentioned possibilities to use games in the classroom: Simulation.

Simulation games involve an inductive learning process in which the student actively investigates to acquire the learning objectives.

From a specific framework or scenario, the teacher introduces elements with the aim of making the process attractive and being able to apply deductive thinking through the resolution of tests, which is key, for example, in the teaching of STEM subjects.

Once again, the aim is to establish patterns and look for solutions using the simulation that the game allows. The simulation will allow the teacher to represent the data and processes of the real world, with which he/she wants the student to work, using different models to experiment with. The game-based learning methodology will provide the teacher with the possibilities to work with different scenarios resulting from the combination of different parameters, which will make it possible to combine the diversity of the students and the different learning environments.

In this way, the teacher promotes learning by discovery in a context of "Learning by Doing" that promotes student motivation, allowing the construction and deconstruction of situations.

The game-based methodology in the CT contexts is developed around these main dimensions:

- Simulations
- Experimentation
- Investigation
- Modelling
- Predictions

3.3.3 The challenges of CT education and how to overcome them

Challenges

According to many experts, students' demand for online education increased over the twenty-year period between 2002 and 2022, but teachers' acceptance of online education remained low, at 32% (Allen & Seaman, 2013, 2). Lloyd, McCoy, and Byrne (n.d.) performed a survey study on barriers to online education and found concerns about pedagogy and teacher's support, as well as the relative quality of face-to-face vs. online course offerings (Smidt et al. 2014). The top emergent challenges are seen below:

Q: Can online delivery offer the same educational quality as face-to-face learning?

A: We can not totally support that traditional methods of online learning offer the same educational results as face-to-face learning. But with CT, the situation might be different. CT offers teachers not only the opportunity to explore new problem-solving processes but also helps to unleash creativity in designing courses, research and policies within their areas of specialisation. One way to offer teachers a chance to learn CT would be to provide an opportunity to undertake online course development, using CT as a model.

That’s why we believe that in the CT case, designing an online course offers better results than face-to-face learning.

Q: Does the school policy support both teachers and students in online delivery?

A: Schools authorities promote opportunities in using technology tools to guide and support pedagogy and promoting collaborative learning in designing online courses. We are witness to the above during the pandemic period where school authorities, teachers and students adequately synchronised on the direction of providing e-learning material.

A common problem example is technology implementation and training. Training many teachers at once may compress the time needed but explode the cost of technology implementation in support of online courses. Categorising these and similar problems can offer clarity and permit an effective strategy to emerge. In the example of a technology and training problem, at the institutional level, it may be possible to parse the problem into categories such as decision-making problems and policy problems, while at the school level technology implementation and training may emerge as a strategic development issue. Once these and similar problems are explored and defined, application of CT strategies may permit further refinement of potential solutions.

Applying Computational Thinking Solutions

Wing (2006) outlined how and why Computational Thinking (CT) is an important skill set for problem solving. According to Wing, CT is a way of conceptualising problems to be solved by humans through integrating fundamental methods derived from computer science. In the case of change management related to adopting online learning, processes to overcome barriers can be derived by breaking down specific problems and identifying patterns in both the problems and any known solutions. Abstracting ideas that form principles can guide solutions and create algorithms or step-by-step solutions that offer logical clarity. By thinking differently about how problems are structured and solutions strategized, learning is grounded in theory but applied to a relevant and useful process.

Decomposition and Recursive: Thinking By breaking problems down into smaller components, it is possible to focus attention on the type of problem and its component issues. Using decomposition to break down the interpersonal concerns cited by teachers in relation to online learning can offer clarity in sequencing and considering each as parts of the whole. Lloyd et al. (n.d.) determined that teachers consider a potential loss of interpersonal interaction with students to be an important barrier to online learning adoption. They decomposed this result into five specific categories of concern related to specific social interaction changes through a weighted factor analysis.

Patterns: Human intuition drives pattern recognition. Common patterns begin to emerge in our awareness as we learn any new skill. Moldoveanu, (2009) discusses how recognizing patterns in problems can promote predictions and strategic solutions. For example, teachers learning to develop online courses don't need to learn how to code each course in HTML. Rather, they should be able to recognize the patterns in the various structures that make up an online learning management system (LMS) such as Canvas, D2L, Moodle or Blackboard. While each LMS has its own unique proprietary specifics, any pedagogically sound course content can be patterned into the system. Students also respond to repetitive patterns in course organisation within a program. Human pattern recognition is the result of abstraction and recursive thinking (thinking about thinking). We think computationally in several instances each day without distinct awareness of doing so.

Abstraction: Czerkawski and Lyman (2016) describe abstraction as *"a tool that permits the creation of large and complex systems of information by defining and generalising from simpler components"*. This can be useful when one encounters the challenges described by (Silber 2007) in designing educational environments and methods when the design is guided by *"ill-structured"* or poorly defined problems at the outset. Abstraction allows us to examine the structure and complexity of problems before focusing on the details. "The most important and high level thought process in Computational Thinking is the abstraction process. Abstraction is used in defining patterns, generalising from instances and parameterization" (Wing 2011). A common process in academic course development is the use of curriculum mapping. Komenda et al. (2015) discuss the use of curriculum mapping through spatial representations of the curriculum so that interconnections can be visualised graphically using learning analytics, algorithms, and models to fully understand learning outcomes. While secondary education curriculum is viewed at the school, program and course level, teachers can engage with the curriculum at each level of abstraction, by viewing a map or web of interconnectivity. Voskoglou and Buckley (2012) describe abstraction as a way of mapping from a complex representation to a simpler one.

Lu and Fletcher (n.d.) assert that problems can be understood and resolved more effectively by encouraging multiple levels of abstraction. For example, if the policy concerns are viewed from both the top-down and bottom-up perspectives, then school administrative and student concerns can be addressed. Focusing on a specific level of abstraction within a problem can yield clarity and more efficient solutions.

Parallels: In computer science, parallel processing is used to accomplish many computing tasks synchronously; parallel thinking lends itself to further defining problems and making sense of them cognitively. Instructional design projects often use both sequential and parallel approaches. For example, Alsofyani and Baharuddin (2012) described their experiences guiding faculty through change using the Technological, Pedagogical, and Content Knowledge (TPACK) model for building competencies in online education. They found that though prior studies have explored a hybrid model of online learning in teaching with technology, participants rated a fully online faculty development experience favourably. Similarly, (Rienties et al. 2013) examined the impact of collaborative teacher training across different institutions and disciplines outside of their own in a parallel teacher training in technology course.

Algorithm Design: Basic algorithms are step-by-step processes for resolving a specific challenge. We use algorithms each day in our thinking in simple human interchanges such as driving. If the light is red, I must stop, if it is yellow, I have choices. With respect to faculty forming algorithms to solve problems, Moldoveanu, (2009) discusses how algorithms can provide logical steps, and also several different models in choosing a viable solution. For example, with regard to technical training, not all participants learn technology at the same pace. Offering a self-paced step-by-step approach in an online learning forum can promote faculty awareness of how students may perceive learning in an online format and build comfort with their own experience in a learning management system.

References

Ahmadi, Mohammad Reza and Shahid Beheshti University, Tehran, Iran. 2017. “The Impact of Motivation on Reading Comprehension.” *International Journal of Research in English Education* 2(1):1–7. doi: 10.18869/acadpub.ijree.2.1.1.

Ahmadi, Navid, and Mehdi Jazayeri. 2014. “Analyzing the Learning Process in Online Educational Game Design: A Case Study.” Pp. 84–93 in 2014 23rd *Australian Software Engineering Conference*. Milsons Point, NSW, Australia: IEEE.

Aho, A. V. 2012. “Computation and Computational Thinking.” *The Computer Journal* 55(7):832–35. doi: 10.1093/comjnl/bxs074.

Allen, E., & Seaman. (2013). Changing course: Ten years of tracking online education in the United States. *CA: Babson Survey Research Group & Quahog Research Group, LLC*. http://ezproxy.library.arizona.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=ED 541571&site=ehost-live

ALLEN, D. E., DONHAM, R. S., & BERNHARDT, S. A. (2011). Problem- based learning. New directions for teaching and learning, 2011(128), 21-29.

Alsofyani, Mohammed Modeef, and Aris Baharuddin. 2012. “A Preliminary Evaluation of Short Blended Online Training Workshop for TPACK Development Using Technology Acceptance Model.” *Turkish Online Journal of Educational Technology*.

ANGELI, C., & GIANNAKOS, M. (2020). Computational thinking education: Issues and challenges. Computers in Human Behavior, 105, 106185.

Barr, Valerie, and Chris Stephenson. 2011. “Bringing Computational Thinking to K-12: What Is Involved and What Is the Role of the Computer Science Education Community?” *ACM Inroads* 2(1):48–54. doi: 10.1145/1929887.1929905.

Benkler, Y. (n.d.). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press. 10.12987/9780300127232

BERGMANN, J. & SAMS, A. (2012). Before you flip, consider this. Phi Delta Kappa International. Vol. 94 (2), p. 25.

BERGMANN, J. & SAMS, A. (2012). Flip Your Classroom: Reach Every Student in Every Class Every Day.

Brennan, Karen, and Mitchel Resnick. 2012. "New Frameworks for Studying and Assessing the Development of Computational Thinking." 25.

CAHILL, G. (N.D.). Why Game-based Learning?, online URL: https://thelearningcounsel.com/article/why-game-based-learning.

CAMP, G. ET AL (2014): PBL: step by step. A guide for students and tutors, online URL: https://teach.its.uiowa.edu/sites/teach.its.uiowa.edu/files/docs/docs/Steps_of_PBL_ed.pdf.

CARVER, R. (1996). Theory for practice: A framework for thinking about experiential education. *Journal of Experiential Education*, *19*(1), 8-13.

Cheliotis, G., Hu, N., & Yew, J. (2014). *The Antecedents of Remix. In ACM 17th Conference on Computer Supported Cooperative Work*, 38. http://dx.doi.org/10.1145/2531602.2531730

Chi, Michelene T. H., and Muhsin Menekse. 2015. "Dialogue Patterns in Peer Collaboration That Promote Learning." Pp. 263–74 in *Socializing Intelligence Through Academic Talk and Dialogue*, edited by L. B. Resnick, C. S. C. Asterhan, and S. N. Clarke. American Educational Research Association.

Committee for the Workshops on Computational Thinking, National Research Council (U. S.)., Computer Science and Telecommunications Board, National Research Council (U.S.), and *Air Force Science and Technology Board*. 2010. Report of a Workshop on the Scope and Nature of Computational Thinking. Washington, D.C.: National Academies Press.

Curzon, Paul, and Mark Dorling. 2014. "DevelopingComputationalThinkingInTheClassroomaFramework.Pdf."

Czerkawski, Betul C., and Eugene W. Lyman. 2016. "An Instructional Design Framework for Fostering Student Engagement in Online Learning Environments." *TechTrends* 60(6):532–39. doi: 10.1007/s11528-016-0110-z.

Dhatsuwan, Apinya, and Monamorn Precharattana. 2016. "BLOCKYLAND: A Cellular Automata-Based Game to Enhance Logical Thinking." *Simulation & Gaming* 47(4):445–64. doi: 10.1177/1046878116643468.

European Commission. Joint Research Centre. 2016. Developing Computational Thinking in Compulsory Education: *Implications for Policy and Practice*. LU: Publications Office.

EFSTRATIA, D. (2014). Experiential education through project based learning. *Procedia-social and behavioral sciences*, *152*, 1256-1260.

FLIPPED LEARNING NETWORK (FLN). (2014) The Four Pillars of F-L-I-P™.

GENTRY, J. W. (1990). What is experiential learning. *Guide to business gaming and experiential learning*, *9*, 20.
GAO, X., & HEW, K. F. (2022). Toward a 5E-Based Flipped Classroom Model for Teaching Computational Thinking in Elementary School: Effects on Student Computational Thinking and Problem-Solving Performance. *Journal of Educational Computing Research*, *60*(2), 512-543.

HARDING DA ROSA, J. (2018). Promoting Project-Based Learning, online URL: https://oupeltglobalblog.com/tag/jane-maria-harding-da-rosa/.

HSU T.-C., CHANG S.-C. & HUNG Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature, Computers & Education.

Ioannidou, Andri, Vicki Bennett, Ashok Basawapatna, Alexander Repenning, and Kyu Han Koh. 2011. *"Computational Thinking Patterns."* American Educational Research Association.

Jenkins, H., Clinton, K., Purushotma, R., & Robinson, A. (2006). Confronting the Challenges of Participatory Culture: Media Education for the 21st Century. MacArthur Foundation. http://digitallearning.macfound.org/ site/apps/nlnet/content2.aspx?c= enJLKQNlFiG&b=2108773&content_id=

Johnson, Terri, Mary Ann Wisniewski, Greg Kuhlemeyer, and Jamie Krzykowsk. 2012. "TECHNOLOGY ADOPTION IN HIGHER EDUCATION: OVERCOMING ANXIETY THROUGH FACULTY BOOTCAMP." *Carroll University 16(Journal of Asynchronous Learning Networks):*63–72.

Karram, O. (2021). The Role of Computer Games in Teaching Object-Oriented Programming in High Schools-Code Combat as a Game Approach. WSEAS *Transactions on Advances in Engineering Education*, 18, 37-46.

Koh, Kyu Han, Ashok Basawapatna, Vicki Bennett, and Alexander Repenning. 2010. "Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning." Pp. 59–66 in 2010 IEEE *Symposium on Visual Languages and Human-Centric Computing.* Leganes, Madrid, Spain: IEEE.

KOLB, D. A. (1984). *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall.

Komenda, Martin, Martin Víta, Christos Vaitsis, Daniel Schwarz, Andrea Pokorná, Nabil Zary, and Ladislav Dušek. 2015. "Curriculum Mapping with Academic Analytics in Medical and Healthcare Education" edited by K. E. Vrana. PLOS ONE 10(12):e0143748. doi: 10.1371/journal.pone.0143748.

KRAJCIK, J. S., & BLUMENFELD, P. C. (2006). *Project-based learning* (pp. 317-34).

Krugel, Johannes, and Peter Hubwieser. 2017. "Computational Thinking as Springboard for Learning Object-Oriented Programming in an Interactive

Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge University Press.

Lessig, L. (2008). Remix: Making Art and Commerce Thrive in the Hybrid Economy. Penguin Press.

LI, M. P., & LAM, B. H. (2013). Cooperative learning. The Hong Kong Institute of Education, 1-33.

Lloyd, Steven, Tami McCoy, and Michelle Byrne. n.d. "Faculty-Perceived Barriers of Online Education." MERLOT *Journal of Online Learning and Teaching*.

Lloyd, S., McCoy, T., & Byrne, M. (n.d.). Faculty-Perceived Barriers of Online Education. MERLOT *Journal of Online Learning and Teaching*.

Lu, James J., and George H. L. Fletcher. n.d. "Thinking About Computational Thinking." 5.

MASON, S. L., & RICH, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. Contemporary Issues in Technology and Teacher Education, 19(4), 790-824

MASSACHUSETTS DEPARTMENT OF ELEMENTARY AND SECONDARY EDUCATION (DOE). (2016). *Massachusetts Curriculum Framework for Science and Technology/Engineering*.

Manovich, L. (2020, April 15). Remix and remixability. Manovich. Retrieved July 15, 2022, from http://www.manovich.net/DOCS/Remix_modular.doc

Moldoveanu, M. 2009. "Thinking Strategically about Thinking Strategically: The Computational Structure and Dynamics of Managerial Problem Selection and Formulation." *Strategic Management Journal* 30(7):737–63.

MOOC." Pp. 1709–12 in 2017 IEEE *Global Engineering Education Conference* (EDUCON). Athens, Greece: IEEE.

MOURTOS, N. J. (1994, November). The nuts and bolts of cooperative learning in engineering. In Proceedings of 1994 IEEE Frontiers in Education Conference-FIE'94 (pp. 624-627). IEEE.

NATIONAL RESEARCH COUNCIL (NRC) (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. Washington, DC: The National Academies Press.

Online Learning Statistics*: Is Online Education the Future?* (2021). Bay Atlantic University. Retrieved July 17, 2022, from https://bau.edu/blog/online-learning-statistics/

Papert, Seymour. 1980. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books.

Pellas, Nikolaos, and Efstratios Peroutseas. 2016. "Gaming in Second Life via Scratch4SL: Engaging High School Students in Programming Courses." *Journal of Educational Computing Research* 54(1):108–43. doi: 10.1177/0735633115612785.

Rienties, Bart, Natasa Brouwer, Katerina Bohle Carbonell, Danielle Townsend, Anne-Petra Rozendal, Janneke van der Loo, Peter Dekker, and Simon Lygo-Baker. 2013. "Online Training of TPACK Skills of Higher Education Scholars: A Cross-Institutional Impact Study." *European Journal of Teacher Education* 36(4):480–95. doi: 10.1080/02619768.2013.801073.

Rob-Bot! The Computational Thinking Card Game! — Rob-Bot Teaching Resources - Teaching Resources for Computing. (2022). Retrieved 26 July 2022, from https://robbotresources.com/blog/2019/2/1/computational-thinking-skills

SAVERY, J. R. (2015). Overview of problem-based learning: Definitions and distinctions. Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows, 9(2), 5-15.

Shute, Valerie J., Chen Sun, and Jodi Asbell-Clarke. 2017. "Demystifying Computational Thinking." *Educational Research Review* 22:142–58. doi: 10.1016/j.edurev.2017.09.003.

SHIN, N., BOWERS, J., KRAJCIK, J., & DAMELIN, D. (2021). Promoting computational thinking through project-based learning. Disciplinary and Interdisciplinary Science Education Research, 3(1), 1-15.

Silber, K. 2007. "A Principle-Based Model of Instructional Design: A New Way of Thinking about and Teaching ID." *Educational Technology*, 5–19.

SLAVICH, GEORGE M. & PHILIP G. ZIMBARDO. 2012. "Transformational Teaching: Theoretical Underpinnings, Basic Principles, and Core Methods." *Educational Psychology Review* 24, no. 4: 569–608.

Smidt, Esther, Brian McDyre, Jennifer Bunk, Rui Li, and Tanya Gatenby. 2014. "Faculty Attitudes about Distance Education." IAFOR *Journal of Education* 2(2). doi: 10.22492/ije.2.2.06.

Social Constructivism & Social Media | socialmedia4444. (n.d.). socialmedia4444. Retrieved July 15, 2022, from https://socialmedia4444.wordpress.com/social-constructivism-social-media/

Soh, Leen-Kiat, Duane F. Shell, Elizabeth Ingraham, Stephen Ramsay, and Brian Moore. 2015. "Learning through Computational Creativity." *Communications of the ACM* 58(8):33–35. doi: 10.1145/2699391.

TRYBUS, J. (2015). Game-based learning: What it is, why it works, and where it's going. New Media Institute.

Voskoglou, Michael Gr., and Sheryl Buckley. 2012. “Problem Solving and Computational Thinking in a Learning Environment.” doi: 10.48550/ARXIV.1212.0750.

Weintrop, David, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. “Defining Computational Thinking for Mathematics and Science Classrooms.” *Journal of Science Education and Technology* 25(1):127–47. doi: 10.1007/s10956-015-9581-5.

Wing, Jeannette M. 2006. “Computational Thinking.”

Wing, Jeannette M. 2011. “Research Notebook: Computational Thinking--What and Why?” 8.

WOOD, D. F. (2003). Problem based learning. Bmj, 326(7384), 328-330.

Wolz, Ursula, Meredith Stone, Sarah M. Pulimood, and Kim Pearson. 2010. “Computational Thinking via Interactive Journalism in Middle School.” P. 239 in Proceedings of the 41st ACM technical symposium on Computer science education - SIGCSE ’10. Milwaukee, Wisconsin, USA: ACM Press.

Yu, L., & Nickerson, J. V. (2011). Cooks or cobblers?: crowd creativity through combination. In ACM 2011 annual conference on Human factors in computing systems (CHI ’11)., 1393–1402. http://dx.doi.org/10.1145/1978942.1979147

Chapter 4: Training Teachers in Computational Thinking: Elements of Innovation and Expected Impact

4.1. Effective teacher professional development (TPD)

Teaching is a very special professional commitment. As the role of the teacher is considered multidimensional, a teacher contributes to the spread of knowledge and empowers students' growth, helping them to develop the spiritual, social and psychological aspect of their personality which makes their presence in a student's life crucial. Teachers act as role models. Many students are influenced by them and often rely on them more than they rely on their family. This increases the teacher's responsibility towards students.

On the other hand, we must not overlook the difficulty of teaching. Its most fundamental aspect lies in the creating a harmonious and effective educator-learner interaction between different people of different ages (adults and minors), of different personalities with different experiences. Moreover, successful teaching basically requires the cooperation of different teachers, each of whom has his/her own temperament, views and philosophy concerning education. Mainly, though, the difficulty increases considering the different cognitive background of each student, their specific social environment, unpleasant experiences they may have in their life and other particular features that determine their attitude and academic performance (Knight & Wiseman, 2005)

All these factors make it clear that teaching could not and should not have a static character. The learning process must be ongoing and endless, a lifelong procedure (Knight, 2002).

It's worth stressing that student-centred teaching models have been very effective so far. Shifting the centre of teaching from teachers to students helped children become more self-confident, encouraged them to discover knowledge and showed them how to construct new information based on their prior experience. Students stopped being passive learners and teachers stopped being the experts of the class (Young & Paterson, 2007). Besides, active learning gives educators the opportunity to apply cooperative teaching methods and help all children engage in the teaching procedure. With the teacher's support, the learning experience became enjoyable and students' interest in the school was enhanced.

As a result of this paradigm shift in our days teachers have acquired new roles .... The educator as a researcher has the flexibility to innovate, inspired by the individuality of the students, on their talents, their desires as well as on their weaknesses. This is "the ultimate professional development", according to Dorothy Suskind (Suskind, 2016). Instead of being limited by the curriculum, teachers "shape" the curriculum themselves, starting their day with the question "What will my students teach me today?" and becoming "directors of their own professional development".

The recent pandemic of Covid-19 established a totally computer mediated educational reality all over the world.

According to UNICEF, "since the outbreak of Covid-19 to date, the pandemic had a devastating impact on the education of millions of children worldwide, making it the largest global crisis for children" in the 75-year history of this organisation.

During this pandemic, education of all grades was relying totally on e-learning methods. The classroom became digital, and its members were interacting only through different digital devices. Thus, technology mediated learning (TML) and communication was widely spread as the sole and unavoidable interaction tool of students and educators. Therefore, we should consider orientating professional development of teachers according to current students' needs and social demands. However even though online TML appears with advantages, like time saving, flexibility and independence of learning pace (Postholm, 2007) it is not yet clear whether learning outcomes are desirable and more effective for students. Distance learning lacks the needed face to face close interaction between students and teachers but on the other hand, offers alternative routes of teaching through online technology tools that make the lesson friendly and fascinating. For example, in synchronous learning environments, students have the capability to communicate and cooperate in smaller groups exploiting the offered digital tools (Center for Academic Innovation, 2020).

Thus, the situation clearly confirms the growing tendency for e-learning strategies that the TPD should inevitably address. However, there is also another issue to be taken into consideration: either our classroom is conventional or digital, it is obvious that educators have to teach their students how to learn on their own. Students need to acquire metacognitive skills that will raise their awareness, internal motivation, and personal responsibility in the educational process. They should be able to connect their prior knowledge with new acquired information to reach deep comprehension. Metacognition influences their critical thinking skills (Magno, 2006). This is the key for an ongoing and lifelong learning process, which is necessary for all. Given the multiculturalism of the classrooms and the graded cognitive level of the students, the development of metacognitive skills is a crucial factor in the acquisition of knowledge and in effective teaching.

How can students obtain this metacognition through web-based learning environments? How can educators support students to develop autonomous learning, self-discipline and self-confidence through these environments?

How can educators support students to transform a complex problem into smaller ones? Finally, which strategies have the educators to apply? The answer is suggested by the disciplines of Computational Thinking.

4.2 Teacher Training in Computational Thinking: Challenges and Opportunities

In recent years, an increasing number of countries have introduced Computational Thinking (CT) into their national curricula as a means of aligning education with the needs of our knowledge-rich societies of the Information Era and the advancements in modern technology (Saidin, Khalid, Martin, Kuppusamy, & Munusamy, 2021; Scherer, 2016). This development has been supported by results of many research studies demonstrating that Computing Science and Computer Programming can enhance student engagement, motivation, confidence, problem-solving skills, communication skills and improve their overall performance in science, technology, engineering, and maths (STEM) (Lye & Koh, 2014˙ Mason & Rich, 2019˙ Pala &Türker, 2021; Scherer, 2016). According to Wing (2006), in its essence, CT is the ability to think like a computer scientist in order to formulate a problem and express its solution in a way that could be processed by a computer.

However, CT has a broader definition that focuses on the process of abstraction, but also includes, among others, concepts such as thinking recursively, problem decomposition, data collection, representation and analysis, testing and verification, analysis and model validation, algorithmic thinking, evaluation, generalisation, simulation and automation (Barr & Stephenson, 2011˙ Wing, 2006). Integrating CT skills into any field in education, besides Computer Science (CS), has proven to be a rather challenging task and teacher training in CT seems to be a key factor for any such attempt to be fruitful (Denning, 2017˙ Saidin et al, 2021).

4.2.1 Challenges for CT in Education

Teacher's understanding of CT is considered to be critically important in order to meaningfully introduce its concepts into the field of education (Denning, 2017· Saidin et al, 2021). As a result of its loose definition, educators seem to struggle with the question of "What is Computational Thinking?" and they are often offered vague or even confusing definitions in response (Denning, 2017· Saidin et al, 2021). Selby and Woollard (2014) argue that there is a need for a robust and clear definition of CT in order to facilitate the teaching of its concepts in actual classrooms in a creative and meaningful way. A well-grounded definition of CT is essential to educators as it would not only allow them to teach and/or demonstrate in practice CT concepts in their classes, but it would also enable them to develop assessment tools to measure their students CT skills level and track their progress (Saidin et al, 2021· Selby & Woollard, 2014).

As Denning (2017) stresses, educators cannot be effective if they are not sure what they are teaching. It seems that the majority of pre-service and in-service teachers have not attended any formal training in CT and they seem to be uncertain about the concepts being involved in it as well as their applicable domains (Saidin et al, 2021). This lack of knowledge has led many educators to falsely believe that CT is a skill strictly related with the fields of CS (Computer Science) and technology (Saidin et al, 2021). Heintz, Mannila and Farnqvist (2016) argue that there is a strong need to develop teachers' digital competencies together with programming skills through specialised courses and/or training programmes and to encourage more educators to include aspects of CS in their subjects. In this context, it is considered extremely important for educators to fully understand the content they are teaching, the pedagogy related to it and the technology they are using (Mason & Rich, 2019). In regard to introducing CT in K12 classes, Shailaja and Sridaran (2015) point out that most national curricula are mostly concerned with "why" and "what" to teach and the "how" part is missing from their content.

It seems that preparing teachers to embed CT in their K12 classes and apply its concepts in various subject areas is a learning process that requires a lot of time as educators not only need to be trained in CT integration strategies for their courses, but they also need a lot of real-world, hands-on practice in order to master it (Denning, 2017˙ Mason & Rich, 2019˙ Saidin et al, 2021). Additionally, becoming proficient in CT concepts, skills and learning environments such as programming, computing or educational robotics can be really difficult and time-consuming for educators who are less competent in using digital technologies or have never been exposed to CT before or even lack confidence because of their limited pedagogical knowledge in CT teaching (Saidin et al, 2021˙ Mason & Rich, 2019).

Apart from teachers' digital competency and CT teaching skills, the lack of relevant and age-appropriate teaching materials and assessment strategies for students' CT skills seem to be major barriers to effectively introducing CT concepts in K12 classes in an interdisciplinary way (Barr & Stephenson, 2011˙ Saidin et al, 2021). Educators trained in CT tend to envision it as a problem-solving methodology that can be applied across various subjects and recognize the benefits of introducing CT in their classroom, but implementing and integrating CT teaching and learning activities into their school's existing curricula is a major challenge for them mostly due to the entailed time-constraints and the extra time they need to devote (Barr & Stephenson, 2011˙ Denning, 2017˙ Garneli, Giannakos, & Chorianopoulos, 2015˙ Saidin et al, 2021). In this context, Kakavas and Ugolini (2019) suggest that classifying the CT concepts that students are able to learn at each grade, setting clear cognitive goals by providing specific examples along with teaching/learning activities to educators and developing valid and reliable CT assessment tools, could effectively promote CT integration into K12 curricula.

Teachers' attitudes toward STEM and CT are also considered to be important factors in effectively promoting and integrating CT in K12 curricula (Saidin et al, 2021˙ Yadav, Hong, & Stephenson, 2016). An experimental study by Yadav, Mayfield, Zhou, Hambrusch and Korb (2014) examined the impact of a one-week CT class on teachers' attitudes toward introducing CT into their classroom and researchers found that teachers who attended not only had a better understanding of CT concepts/skills, but they were also more positive and open in embedding CT in their future classes as compared to teachers of the control group.'

4.2.2 Benefits of teacher training in CT

As difficult as it may be, successfully integrating CT concepts/skills in K12 curricula is definitely worth making an effort, as CT can provide many benefits to both students and educators. Positive effects of CT introduction in K12 education include enhancing critical and analytical thinking among learners, promoting STEM education, changing teachers' and learners' attitudes towards it and further improving pedagogy and curricula (Saidin et al, 2021˙ Mason & Rich, 2019). It seems that once students start to think computationally they are able to perform better in problem-solving activities in class and in everyday life situations, as they are able to better understand and express a problem and its appropriate solution by effectively applying concepts such as abstraction, decomposition, algorithmic design, generalisation and evaluation (Selby & Woollard, 2014˙ Mason & Rich, 2019).

Further, having in mind Wing's (2006) broad definition of CT, similar claims could be made for school teachers with CT training. A study by Pala and Türker (2021) shows that participation in an Arduino programming class had positively impacted pre-service teachers' creativity, critical thinking, algorithmic thinking and CT skills. As social needs change rapidly, driven by advancements in technology, CT is becoming an essential "skill-set" that everyone, not just computer scientist, should have (Pala &Türker, 2021˙ Wing, 2006). Regarding teaching professions, educators with CT skills are able to be more creative in teaching the younger generations and as they have higher confidence in their CT skills, they can be expected to have greater commitment and job satisfaction (Saidin et al, 2021). In addition, encouraging educators to develop and/or improve their digital competencies and CT skills can be a small step towards the creation of a more relevant educational system, suitable to the needs of modern societies (Saidin et al, 2021˙ Heintz, Mannila, & Farnqvist, 2016).

4.3 Technology Mediated Learning (TML) and Computational Thinking (CT) for the Professional Development of the Educator

As already mentioned, CT uses advanced computational possibilities to comprehend and solve complex problems, exhibiting a specific way of thinking and thus is bonded with TML through the needed skill of technological literacy.

Technological Literacy can be defined by two basic skills:

a. knowledge and use of basic functions concerning computers

b. use of technology in order to increase productivity and achieve professional development of teachers, in order to support their teaching by utilising technology and by designing and implementing activities and applications. (Borthwick, Hansen 2017)

According to the Council for the Accreditation of Educator Preparation (CAEP) Standards, Technological Competence defines an educator's capability to "model and apply technology standards as they design, implement, and assess learning experiences to engage students and improve learning, and enrich professional Practice" (CAEP, 2015, Faloon, 2000).

4.3.1 Technology Mediated Professional Development programs (TMPD programs)

Technology mediated teaching and learning (Oliver & Herrington, 2003) sets as a prerequisite Technological Literacy and refers to the digital environment which can be used by educators in order to succeed in the maximum interaction between them and students and achieve the learning outcomes within the applied curriculum (Bower, 2019).

Some examples of teaching methods and techniques that can be practised in digital environments incorporating TML are the following:

- peer learning
- team working
- use of web-based resources, videos, and simulations
- use of electronic platforms and databases

- interactive digital lab environments for science lessons
- games and analogies
- use of software to make quizzes and crosses polls, or multiple choice questionnaires

All these methods seem to promote communication and collaboration skills and provide teachers with the appropriate tools to create inquiry-based lessons that will help their students to develop strategies, think logically and train their critical thinking. Learning outcomes should be categorised to psychomotor development of students, gain of cognitive goals and finally…entertainment and satisfaction during the learning procedure. In that way students will have learned… how to learn!

It is clear after all that to achieve great learning outcomes teachers should have the possibility to receive access to Technology Professional Development (TPD) educational programs. And since research on the field has shown disappointing results concerning conventional educational programs, based on lectures by expert instructors, the need for programs involving social collaboration and providing teachers with access to peer- support networks has gained the game (Adsit N.J. AACTE, 2004).

However, TPD programs for teachers should include training on the incorporation of ICT for the following:

- participation in digital networks through learning communities that use the "situated learning" model of Vigotsky.
- creation of educational videos including simulations where needed
- setting up classroom environments and databases to be used by students
- creation or use of available electronic material based on inquiry learning.

4.3.2 TML and CT

CT consists of three main elements:

a. Numerical and non-numerical algorithms as well as software for modelling and simulation for the solution of problems.
b. Computing infrastructure for the support of science problems and
c. The development of hardware and software systems, internet systems, data management systems, necessary for the solution of demanding problems in computers (Grover & Pea, 2013).

The above states the strong need of embedment of online technology mediated e- learning methods to CT teaching.

One of the crucial goals concerning learners' education is the acquisition of skills enabling them to solve various problems. The educator, developing professional ethics, can be innovative by using teaching techniques/practices with a view to upgrade his/her educational work, confident of successfully coping with ongoing changes. Obstacles could appear by certain educators' beliefs, that is, their ideas concerning the building of knowledge, which determine their teaching choices, direct their teaching behaviour and configure their actual teaching.

A key point to start with is to change the way of thinking about teaching and learning. It is suggested that continuous reflection and self-assessment of the didactic act should be exercised along with the support of training for professional development. The goal is to produce and utilise new pedagogical knowledge with the help of educators themselves and the application of their experience. CT is something new which we want to implement in the educational process to improve the quality of teaching and students' learning. This indicates that professional development of all educators, of all fields, is necessary if we are to apply CT in our subject areas. Lifelong training of educators is nowadays recognized as an inherent part of their professional career and development.

To summarise, future scientific progress in all societies is dependent on the development of CT, which is why modern citizens should prepare for the future by being equipped with CT skills and by familiarising themselves with the development of CT and computational ideas early on, from basic education. In previous years, learners' skills involved writing, reading and arithmetic. Nowadays we should add CT to those skills. That is to show the importance of educators' preparation if they are to cope with their complex work, which is the education of the citizens of the future. CT includes a range of skills which all educators are willing to learn to use. The first steps are being made with CT incorporated into renewed curricula in the compulsory education of many countries.

Actually, many countries are investing simultaneously in long term delivery of computer science (Wing, 2016). Nevertheless, it is necessary to expunge the misunderstanding that CT is solely linked with the use of computers. The development and promotion of CT is an advantage for scientific and technological progress which can improve people's lives and thus, it becomes necessary for it to be incorporated into general education. The educators, who will be called upon to incorporate CT into the latter, must receive support with proper preparation, to succeed in facing the pedagogical dynamics of CT.

All European countries are currently taking initiatives to promote CT. For example, in Greece, the first steps are made with the Bebras competition, (www.bebras.org) whose goal is to sensitise the educational community, with a view to filling the gaps existing in basic education, as far as CT is concerned. The goal of the aforementioned competition is for learners to be inspired by various matters dealing with computer science and CT, which have been adapted according to their age by computer scientists from all over the world. Learners' participation in the competition is a short, entertaining and enriching learning experience. The large part of the learners' participants as well as that of the educators who have been involved in the realisation of the competition, shows the recognition of the pedagogical value of CT, along with their intention to incorporate it into their daily practice.

4.4 "Train the trainer"

4.4.1 Why Computational Thinking?

Learning CT is not the same as computer science and it does not involve only this field . CT can be applied to every scientific area as well as in daily life. It focuses mainly on the development and resolution of problems of the real world. It uses a number of skills and transforms complex problems into simpler ones.

Here it seems appropriate to refer to the famous 4 rules of René Descartes, who is both a mathematician and a philosopher, described in his work *Rules for the direction of the mind*:

(1) accept nothing as true that is not self-evident,

(2) divide problems into their simplest parts,

(3) solve problems by proceeding from simple to complex, and

(4) recheck the reasoning.

Using the skill of removing details, it makes problems understandable and that leads to immediate solutions. The map of the underground railway of London is an example, which includes only the information necessary for one to determine a route from one station to another (Cansu, S. K., & Cansu, F. K., 2019). With the skill of decomposition or analysis, the problem is broken into smaller ones with understandable elements. With generalisation we are able to recognize some of the issues of the problem as known ones (Κούσης, 2017). With algorithmic thought we determine the steps to be taken to find solutions to the individual, simpler problems, and with automation, we create algorithms to be used in other problems as well (Cansu, S. K., & Cansu, F. K., 2019).

In the field of education, playmaking is suggested, that is, the application of mechanisms that target learners' mobilisation because of the motives offered. In this way, participation in learning activities increases, while at the same time they are pleased and satisfied by their participation. The educator should be able to recognize and use the cognitive-emotional ability of CT, which should be developed, according to the learning aim set. Thus, educators should be given motivation to prepare attractive learning scenarios, or even be given attractive scenarios to use, in order to help their learners solve problems with CT (Kuo, M.J., 2007).

The technological environment created will aim at activating and urging educators towards creativity. All educators should have access to the educational materials, the tools and the directions created for CT to be used correctly.

Learners who are trained in the use of CT acquire important skills, such as self-confidence in coping with complex problems, tolerance for ambiguity which can exist in a problem, skills of communicating with their peers while working together to reach a solution and problem management skills, while at the same time they become more persistent when trying to solve a problem (Kouzoukas, 2019).

4.4.2 Educational Strategies for trainers

Considering all above, a critical question arises: In what way and with which educational tools can a teacher be familiarised with CT? First of all, teachers have to be persuaded to use CT as a very useful and effective method for their lessons. For this purpose, teachers need to understand the basic principles of CT, and learn how to adopt them. Then they should be provided with the reasons to use it and to convince themselves of its effectiveness. Finally, they should be provided by the proper resources and some tools that would facilitate the application of the teaching method in class (Pinder, N., 2022).

For CT to integrate into different areas and subjects of compulsory education, it is important to provide all current and future educators with sufficient knowledge of its aspects, skills and characteristics, as well as how it can be integrated into lessons.

Jeannette Wing, of Columbia University, brought the idea of computational thinking in a paper she wrote in 2006. She says, "*Informally, computational* *thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines".* (Wing, J., (2006). This is a great statement on the fact that CT can be applied for online lessons but not only. And speaking about "mental activity" one should consider the reproduction and/or application of a computer algorithm in order to analyze and finally come to the solution.

There are three main reasons educators of all grades should use CT. First, it helps students to cope with almost any problem providing them with a step-to-step procedure in order to reach to an end solution, through a variety of disciplines, leverages. This is a powerful skill and valuable equipment for their lives. Second, it exhibits the strength of computer technology and finally complements and enhances existing school curriculum (Waterman, K., P. Goldsmith, L. & Pasquale, M. 2019).

In order to embed CT in their lessons teachers need to know how to:

- Formulate a given problem in a way that enables us to use a computer and other tools to help solve them.lessons,
- Logically organise, analyse and represent data through abstractions such as models and simulation.
- Succeed in automating solutions through algorithmic thinking (a series of ordered steps).
- Identify, analyse, and implement possible solutions with the goal of achieving the most efficient and effective combination of steps and resources.
- Generalise and transfer this problem-solving process to a wide variety of problems.

In the formation of the complex and multiple identity of the students of the 21st century, the teachers have a decisive role. It takes dedication, training and effort to create a community of educators ready to convey to students the essence and true nature of CT. Especially for teachers who have not been taught computer science well enough to be able to incorporate key concepts into their teaching, thus, their choices and degree of involvement depend on their willingness to train and their enthusiasm towards the subject.

The dual challenge of understanding the content and choosing the proper pedagogical strategy to transfer knowledge and skills in students creates a situation quite complex for teachers. As with TML educational programs it seems that educators can benefit significantly through:

- participation to educational programs
- appropriate training workshops
- hands on trainings
- seminars
- self-study
- participation to communities of learning and practice, interested on the subject (Patton, K. Parker, M. 2017)

In addition, the cooperation of teachers with computer science experts is considered necessary. Teachers' perceptions about computer science and its application on various disciplines can change after training (Yadav, A., Gretter, S., Good, J., & McLean, T. 2017).

Educational programs, training and seminars will definitely help the educators to develop a shared understanding of the teaching of CT in schools, provide the pedagogic approach for teaching and offer guides for assessment. They will help pre-service teachers to acquire new ways of thinking and incorporate IT to their lessons. (Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T., 2014). Training of educators in CT and online teaching methods could be organised within the framework of interschool or intraschool educational programs of many countries.

Participation to learning and application communities under the umbrella of online TML and CT is unambiguously the most important self-training medium.

4.4.3 Tools and resources for self-study

In order to be encouraged to use computational thinking in class, all teachers should have the possibility to have at their disposal a number of tools and resources, in order to understand in depth all aspects of application of CT when they teach their subjects. Of course, as explained before, digital literacy is a basic prerequisite to the latter. Since computational thinking consists of the breakdown of a complex problem to simpler ones, pattern recognition and creation of the algorithm (set of instructions), it is clear that a teacher could potentially use the method in class using only basic digital skills or even no digital skills. So, teachers need to be educated on the ways to integrate computational thinking to their lessons using the disciplines on which a computer is based to solve a problem or create an activity. However even if breaking the problem and providing an algorithm could be based on a methodology that can be created without use of ICT, the pattern recognition and representation strongly needs the help of digital technology to encourage students to use their imagination and recognize similar situations developing their critical way of thinking. CT can be applied to most subjects at school, starting from elementary or even pre-elementary education. However, the first consideration of a newcomer teacher to the world of computational thinking should be an extensive literature search and self-study on the definition and disciplines of CT.

- Online courses are organised by Universities, focusing to educate teachers – train the trainer- on various subjects and grades, to the approach and disciplines of CT. Some of them provide free material for teachers to use in their classes.

i) UCX University of Canterbury provides MOOCs, free online courses on several subjects. The computational	https://www.edx.org/course/

thinking course provides basic knowledge of CT, as well as tools for teachers to teach computational thinking, from unplugged to plugged in with computer programming. The course is based on the website csunplugged.org, a collection of free teaching material for educators	teaching-computational-thinking
ii) Coursera is a free signing in project network, where references are given for a series of online training on computational thinking. The MOOC "Problem Solving Using Computational Thinking" is organised by Michigan University. The course is focusing on the disciplines of CT, "abstraction, problem identification, decomposition, pattern recognition, algorithms, and evaluating solutions. Students will puzzle through some real-world cases that illustrate how computational thinking can be used to solve complex problems. They will also complete a project that allows them to apply computational thinking to a real-world situation".	https://www.coursera.org/courseraplus https://www.coursera.org/learn/compthinking?action https://www.coursera.org/browse
iii) International Society for Technology in Education (ISTE), among other activities, organises online courses. One of them is titled **"Introduction to Computational Thinking for Every Educator"** and focuses on teaching how to integrate CT to different subjects and grade levels.	https://www.iste.org/areas-of-focus/computational-thinking-in-the-classroom
iv. Courses are also offered by **KA1 Erasmus projects** of personal teachers' mobility in the area of Information Technology and related subjects. Teachers at schools should be encouraged to apply to such European projects under the umbrella of STEM education.	https://www.englishmatters.org/index.php/pages/view/erasmus_plus/erasmus?gclid

v. **The Computer Science Education Research Group at the University of Adelaide** in Australia has been partnering with Google to create introductory courses for implementing Australia's Digital Technologies Curriculum and teaching computer science and computational thinking at primary and secondary levels, explicitly tied to the Australian curriculum	(https://csdigitaltech.appspot.com

- Digital platforms, networks and websites are also a powerful tool for self-training and study of concepts of CT, through information given, learning and acting communities, resources, and teaching material. They are usually provided by supporting organisations such as CSTA, ISTE, and the National Science Teachers Association (NSTA), which are also developing and sharing online tools and resources for current and future teachers.

i) **CAS** (Computing at Schools) is an international network community of teachers, academics and professionals who are involved in computer science and provides teaching resources and material. CAS offers an interesting free guide for teachers involved in Computational Thinking.	https://www.computingatschool.org.uk/teaching-resources/2014/june/cas-computational-thinking-a-guide-for-teachers https://www.computingatschool.org.uk/media/ajylyssj/150818computationalthinking.pdf
ii) **BSB** Education (Build Something Different) is a community of experts who support teachers, parents and students and provide them with digital skills. It is a pedagogical foundation that among others, provides online lessons.	https://bsd.education/l/computational-thinking/

BSB provides upon request a workbook on CT with activities to apply in class	
iii**) Computational Thinking in K–12 Education teacher resources ISTE** (International Society for Technology Education) organisation in collaboration with the Computer Science Teachers Association (**CSTA**) created a short manual on K-12 Education. The workbook describes the basic disciplines of CT, skills needed to teach using CT disciplines and provides some learning experiences on several subjects.	https://cdn.iste.org/www-root/2020-10/ISTE_CT_Teacher_Resources_2ed.pdf
iv) **Neoblog** is a resources platform that supports teachers providing tools and tested online teaching methods. It provides background knowledge on CT concepts and some digital tools that could support teachers to apply CT lessons in class: Scratch, Kodable, Minekraft and others.	https://blog.neolms.com/6-digital-tools-that-encourage-computational-thinking/ https://scratch.mit.edu/ https://www.kodable.com/ https://education.minecraft.net/
v) **CS unplugged** is a collection of free teaching materials on computer science and provides training for educators on several Information Technology subjects.	https://www.csunplugged.org/en/
vi) **Google's Exploring Computational Thinking** website provides more than 130 lesson plans and sample programs aligned with international education standards; a collection of videos demonstrating how computational thinking concepts are used in real-world problem solving as well as and a "Computational	http://g.co/exploringCT

Thinking for Educators" online course (http://g.co/computationalthinking).	
vii) TPACK: Technological Pedagogical Content Knowledge Framework. TRACK framework combines three types of knowledge, technological knowledge (TK), pedagogical knowledge (PK), and content knowledge (CK). TRACK theory should be adapted from educators in order to enhance their professional development.	https://educationaltechnology.net/technological-pedagogical-content-knowledge-tpack-framework/
viii) **Computational Thinking Initiatives**. This web page offers programs and resources aiming to develop and cultivate computational thinking for educators and learners.	https://www.computationinitiative.org/

References

Adsit N.J. (2004): Technology Mediated Professional Development Programs. American Association of Colleges for Teachers Education.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48-54

Borthwick A.C. & Hansen R. (2017) Digital Literacy in Teacher Education: Are Teacher Educators Competent?, Journal of Digital Learning in Teacher Education, 33:2, 46-48, DOI: 10.1080/21532974.2017.1291249

Cansu, S.K. & Cansu F.K., (2019): An Overview of Computational Thinking. International Journal of Computer Science Education in Schools, April 2019, Vol. 3, No. 1, ISSN 2513-8359

Denning, P. J. (2017). Remaining trouble spots with computational thinking Communications of the ACM, 60(6), 33-39

Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. In 2015 IEEE Global Engineering Education Conference (EDUCON) (pp. 543-551). IEEE

Grover, S. & Pea, R., Computational Thinking in K-12. Educational Researcher, Vol. 42 No. 1, pp. 38–43. doi: 10.3102/0013189X12463051

Heintz, F., Mannila, L., & Farnqvist, T. (2016). A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K–12 Education. Frontiers in Education Conference (FIE) (pp 1–9). IEEE. doi:10.1109/fie.2016.7757410

Falloon, G. (2020). From digital literacy to digital competence: the teacher digital competency (TDC) framework Education Tech Research Dev. https://doi.org/10.1007/s11423-020-09767-4

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41, 51–61. doi:10.1016/j.chb.2014.09.012

Kakavas, P., & Ugolini, F. C. (2019). Computational thinking in primary education: A systematic literature review. Research on Education and Media, 11(2), 64-94.

P. Knight (2002), A systemic approach to professional development: learning as practice. Teaching and Teacher Education 18 (2002) 229–241.

Knight, S.L., Wiseman D. (2005): Professional Development for Teacher s of Diverse Students. Journal of Education for Students placed at Risk: A Summary of the Research, 10(4), 387–405

Kouzoukas, G., (2019): Thesis, Design and evaluation of an introductory teacher training workshop to understand the concept of Computational Thinking

Kuo, M.J. (2007): How does an online game based learning environment promote students' intrinsic motivation for learning natural science and how does it affect their learning outcomes? The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning, Jhongli City, Taiwan.

Magno, C. The role of metacognitive skills in developing critical thinking. Metacognition Learning 5, 137–156 (2010). https://doi.org/10.1007/s11409-010-9054-4

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. Contemporary Issues in Technology and Teacher Education, 19(4), 790-824

Oliver, R. & Herrington, J. (2003): Exploring Technology-Mediated Learning from a Pedagogical Perspective, Interactive Learning Environments, 11:2, 111-126

Pala, F. K., & Mıhcı Türker, P. (2021). The effects of different programming trainings on the computational thinking skills. Interactive Learning Environments, 29(7), 1090-1100.

Patton, K. & Parker, M. : Teacher education communities of practice: More than a culture of collaboration. Teaching and Teacher Education Volume 67, October 2017, Pages 351-360, https://doi.org/10.1016/j.tate.2017.06.013

Pinder, N., (2022): ISTE :https://www.iste.org/explore/computational-thinking/why-you-should-integrate-computational-thinking-your-curriculum.

Saidin, N. D., F. Khalid, R. Martin, Y. Kuppusamy, and N. A. Munusamy. Benefits and Challenges of Applying Computational Thinking in Education. International Journal of Information and Education Technology 11, no. 5 (2021): 248-254

Scherer, R. (2016). Learning from the past–the need for empirical evidence on the transfer effects of computer programming skills. Frontiers in Psychology, 7, 1390

Shailaja, J., & Sridaran, R. (2015). Computational thinking the intellectual thinking for the 21st century. International Journal of Advanced Networking & Applications, 7, 39-46

Selby, C., & Woollard, J. (2014). "Refining an understanding of computational thinking

Suskind, D. (2016): Teacher as Researcher: The Ultimate Professional Development. https://www.edutopia.org/

Synchronous tools and how to use them. Center for Academic Innovation, (2020) https://onlineteaching.umich.edu/synchronous-tools-and-how-to-use-them/

UNICEF Greece (2022): A day at school during COVID-19 https://www.unicef.org/greece/en/stories/day-school-during-covid-19

Waterman, K.,P. Goldsmith, L. & Pasquale, M. (2019): Integrating Computational Thinking into Elementary Science Curriculum: an Examination of Activities that Support Students' Computational Thinking in the Service of Disciplinary Learning. Journal of Science Education and Technology https://doi.org/10.1007/s10956-019-09801-y

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education (TOCE), 14(1), 1-16

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. TechTrends, 60(6), 565-568

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational Thinking in Teacher Education. Emerging Research, Practice, and Policy on Computational Thinking, 205–220. doi:10.1007/978-3-319-52691-1_13

Young, L. & Paterson, B. (2007): Developing a Student-Centered Learning Environment. Lippincott Williams & Wilkins, p.5

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35

Wing, J., (2016).Computational Thinking and thinking about Computers. Phil. Trans. R. Soc. A (2008) 366, 3717–3725. doi:10.1098/rsta.2008.0118.