TechTrends (2022) 66:745–747
https://doi.org/10.1007/s11528-022-00766-1

COLUMN: HISTORY CORNER

Computational Thinking in Education: Past and Present

Linda Payne[1] · Andrew Tawfik[1] · Andrew M. Olney[1]

Accepted: 30 July 2022 / Published online: 20 August 2022

Abstract

As computers have become commonplace in everyday life, educators have begun to shift focus from *working* with computers (computer literacy) to *thinking* with computers (computational thinking). This article describes the progression of computational thinking (CT) from a historical perspective. This paper will first review the early stages of CT in the mid-1900s, along with its evolution over the succeeding decades. Finally, the article concludes with a discussion of proposed educational benefits, along with implications for future learning.

Keywords Computational thinking in education · Constructivism · Computer science

Early Years of Computational Thinking

One of the earliest researchers in using computers in support of learner-centered approaches included Seymour Papert, who also looked to extend educational theories beyond behaviorism (Ackermann, 2001; Papert, 1980; Piaget & Inhelder, 1967). Most notably, Papert extended constructivism to a paradigm that was later termed constructionism. This paradigm of constructionism states that learning happens best when learners make or constructartifacts in learning and engage in conversations related to their creations. Furthermore, Papert's theory emphasized the significance of tools to shape and reshape ideas, thought processes, and beliefs. He also stressed the importance of contextual learning in human development (Ackermann, 2001; Papert, 1980).

In the introductory chapter of *Constructionism* (Papert, 1991), Papert makes an important clarification about using computers to teach. He explains that simply transferring the in-person instruction to computer-aided instruction is not the point of constructionism; rather, computers should alter the nature of the learning process, shifting the balance between transfer of knowledge to students and the product of knowledge by students. He theorized that "computers may affect the way people think and learn" and presented ways in which "computers might enhance thinking and change patterns of access to knowledge" (Papert, 1980, p. 3). Papert (1980) reinforces this view as he reflects, "I began to see how children who had learned to program computers could use very concrete computer models to think about thinking and to learn about learning and in doing so, enhance their powers as psychologists and as epistemologists" (p. 23).

Broadening Computational Thinking to a Wider Audience

While computer science and the idea of computational thinking (CT) have been around for decades, CT it is still relatively new to educators, particularly those working in pre-college settings. In the 1990s, CT was taught primarily at the university level. K-12 computer courses during this time were focused mostly on computer literacy classes and some computer programming elective classes. At the turn of the century, educators and policymakers began to acknowledge the understanding of the mechanisms of digitalization as an important 21st-century skill (Denning & Tedre, 2019). In the mid- to late 2000s, the focus of CT began to shift from the computer science field to thinking of it as "a fundamental skill for everyone, not just for computer scientists" (Wing, 2006, p. 33). In an effort to extend beyond higher education, educational researchers

✉ Linda Payne
lapayne2@memphis.edu

Andrew Tawfik
aatawfik@memphis.edu

Andrew M. Olney
aolney@memphis.edu

[1] University of Memphis, Memphis, TN, USA

(Sykora, 2020; Wing, 2006) suggested that computational thinking should be added to every child's analytical ability in reading, writing, and arithmetic.

Defining Computational Thinking

As the utility of computers has changed in society, so has the definition of CT. In her 2006 article, Wing posited that "computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33). As interest grew over time, researchers encountered challenges regarding the ill-structured nature of the discipline (National Research Council, 2010). Barr et al. (2011) extended other approaches to include additional skill sets, including a more comprehensive view of CT and problem-solving that included the following:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them.
- Logically organizing and analyzing data.
- Representing data through abstractions, such as models and simulations.
- Automating solutions through algorithmic thinking (a series of ordered steps).
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources.
- Generalizing and transferring this problem-solving process to a wide variety of problems.

Perhaps the most dramatic difference in this new definition was the idea that students could structure problems in a way that would facilitate the use of computers and other tools to help solve those problems, essentially helping them to think like a computer. As outlined above, this new definition also included more traditional components of CT methods and included the generalization and transference of the problem-solving process of CT. In doing so, these perspectives build on Papert's constructionism (1991) and Wing's (2006) suggestion that CT should be considered a fundamental skill for all students.

While the definition is subject to debate, many experts agree that CT involves a set of thinking skills, and based on an analysis of former studies, Korkmaz et al. (2017) suggested that this set of skills should include creativity and cooperation, in addition to algorithmic thinking, critical thinking, and problem solving.

Computational Thinking, Education, and Future Directions

In the first decade of the twenty-first century, as the definition of CT became more concrete, scientists and educators were finding new ways to integrate CT across classrooms and age groups, which included exploring new tools to make coding more accessible and easier for children to learn. In 2007, the MIT Media Lab released a block-based, visual programming language called Scratch (https://scratch.mit.edu/) to the public, which was aimed at helping children learn to code. Block-based and visual approaches allow students to drag and drop pieces of code to form programming commands, making coding much easier for young children to learn. In the following decade, the success of blocks programming in schools led to commercial products using this technology, such as Osmo (www.playosmo.com) and Kano (https://kano.me/us) also emerged, which provided alternative visual approaches to CT education directed toward younger learners.

In terms of future directions, more recent publications indicate a trend toward research in connecting and integrating CT within disciplinary education, especially with regard to STEM education (for examples, see Li et al., 2020a, b). There is also a combination of CT with other prominent instructional strategies, such as intelligent pedagogical assistants, problem-based learning techniques, educational gaming, and flipped classroom strategies (Gong et al., 2020).

Conclusion

Over the years, agreeing on a definition for computational thinking has proven to be a difficult task among researchers and industry leaders (Gong et al., 2020). Based on an analysis of former studies, Korkmaz et al. (2017) concluded that those computational thinking skills include creativity, algorithmic thinking, cooperation, critical thinking, and problem solving.

While the definition of CT continues to evolve, the field will likely continue to foster new ideas, developments, and growth. As history unfolds, external forces (e.g., a move toward distance learning) may require broader perspective to ensure that CT is accessible to a broader array of learners.

Finally, educational technologists and instructional designers can play an important role in weaving in CT as the field becomes an important skillset in other domains.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant 1918751. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference? *Scientific Research*. http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning and Leading with Technology*. http://www.learningandleading-digital.com/learning_leading/20110304?pm=2&pg=22#pg22

Denning, P., & Tedre, M. (2019). Thinking like a computer. *American Scientist, 107*(3). https://www.americanscientist.org/article/thinking-like-a-computer

Gong, D., Yang, H. H., & Cai, J. (2020). Exploring the key influencing factors on college students' computational thinking skills through flipped-classroom instruction. *International Journal of Educational Technology in Higher Education, 17*(19). https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-020-00196-0

Korkmaz, O., Cakir, R., & Ozden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior, 72*, 558–568. https://doi.org/10.1016/j.chb.2017.01.005

Li, Y., Schoenfeld, A., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020a). On computational thinking and STEM education. *Journal for STEM Education Research, 3*, 147–166. https://link.springer.com/article/10.1007/s41979-020-00044-w

Li, Y., Schoenfeld, A. H., diSessa, A. A., Grasser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020b). Computational thinking is more about thinking than computing. *Journal for STEM Education Research, 3*(1), 1–18. https://doi.org/10.1007/s41979-020-00030-2

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press. https://doi.org/10.17226/12840

Papert, S. (1980). *Mindstorms*. Basic Books.

Papert, S. (1991). Situating constructionism. In S. Papert & I. Harel (Eds.), *constructionism* (pp. 1–8). Ablex Publishing Corp.

Piaget, J., & Inhelder, B. (1967). *The child's conception of space*. W. W. Norton & Co.

Sykora, C. (2020). Computational thinking for all. *ISTE*. https://www.iste.org/explore/computational-thinking-all

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://dl.acm.org/doi/fullHtml/10.1145/1118178.1118215