

Editorial

Editorial: Educational Computing and Computer Science

Betul C. Czerkowski
Editor, ITET

Eugene W. Lyman
Managing Editor, ITET

Abstract The resurgence of interest in teaching computational thinking (CT) to K-12 students is an opportunity to provide students with a critical foundational skill. This will only be possible, however, if CT is understood to be separable from the more specific field of Computer Science.

Keywords: *editorial, computational thinking, computer science, educational computing, 21st century skills*

Jeanette Wing's influential 2006 article on computational thinking (CT) asserted that CT should be taught in K-12 schools as a foundational skill along the lines of reading, writing and arithmetic. This assertion fired a renewed interest in study, teaching and research about computational thinking in general and, more specifically, about teaching computer programming in schools. This is a kind of resurgence of the work done by Seymour Papert (1980), who first drew attention to the importance of educational computation, especially for younger students. Now that computers are a ubiquitous part of daily life and the future looks as if it will become even more digitized, an understanding of computation—what it is and how it works—begins to seem more and more like a general skill for daily life. This basic need, however, does not appear to be defining the direction of CT teaching in K-12 education.

Schools across the United States are starting to experiment with teaching CT. Organizations such as CSTA (The Computer Science Teachers Association), ISTE (International Society for Technology in Education) and SRI International are supporting and promoting more systematic and organized research and implementation efforts. We have noticed, however, a tendency to regard teaching CT as starting and ending with teaching programming in schools. This in turn appears to be linked to the expectation that an increasing number of children will develop an interest in computer science and STEM fields. While CT and computer science are linked, computational thinking is not simply computer science; teaching CT should not be reduced merely to teaching Scratch or Alice. The fact that many proponents of CT are computer scientists themselves unfortunately reinforces the assumption that CT must, inevitably, point to computer science as a sole destination.

So why are CT and Computer Science so often conflated? Computer science is the study of computers and computational systems. Computer scientists extend knowledge in their field by researching topics like artificial intelligence, network topologies, data structures, programming languages, software engineering and theories of computing. Computational thinking, on the other hand is “an approach to solving problems in a way that can be implemented with a computer” (Barr & Stephenson, 2011, p.51). In other words, CT is a problem-solving approach involving methods related to computers and computational theory; it is critical that solutions derived from it can be transferable to other subject areas. Coding may be an excellent vehicle for teaching CT, but it is important to understand that the emphasis is on the underlying thinking processes such as abstraction, automation, pattern identification, and so on, rather than simply learning the programming language itself.

Educational computing has been used for decades, but the recent spotlight on CT posits a more generalized and broad-reaching approach to the teaching of computation. Arguably, teacher education programs in the U.S. are lagging behind the most recent developments. The teaching of CT is still largely embedded in computer science courses; much effort will be required to make CT a 21st century skill taught widely across different subject fields. At ITET, we are putting together a special issue on CT to raise awareness and to discuss the positioning of CT in higher education and specifically in teacher education programs. See our call for proposals for more information.

We hope you enjoy reading our new issue!

References

- Barr, V. & Stephenson, C. (2011 March). 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.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books, Inc.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.