Chapter 1
A Brief Discussion on Incentives and Barriers to Computational Thinking Education

Hüseyin Özçınar
Pamukkale University, Turkey

ABSTRACT

The idea that computational thinking or algorithmic thinking should be taught to everyone dates back to the 1960s. First in 1960s, Alan Perlis argued that computer programming should be taught to everyone because it can be used as a mental tool for understanding and solving every kind of problem. In 1980s, under the leadership of Seymour Papert, students at the level of primary education were attempted to be taught LOGO programming language with the aim of gaining procedural thinking skill. After the publication of Jeannette Wing’s “computational thinking” in Communications of the ACM in 2006, the idea that the basic concepts of computer science should be learned by all was started to be debated widely again. In the present paper, the justifications for teaching computational thinking and applicability of teaching computational thinking within the context of existing conditions will be discussed.

INTRODUCTION

It has been emphasized by some computer scientists since 1960s to teach computer science to everyone and the importance of it to be known by everybody. Prominents of these researchers are Alan Perlis, Seymour Papert, Alan Kay, Adele Goldberg, Cynthia Solomon, and Andrea diSessa’dr (Guzdial, 2015). First in 1960s, Alan Perlis argued that computer programming should be taught to everyone because it can be used as a mental tool for understanding and solving every kind of problem. According to Perlis, the value of computers stems from explaining problems and forming a specific way of thinking in the process of solution development rather than contributing to the process of problem solving as a tool. Alan Perlis suggested that this way of thinking, named as “Algorithmizing”, should be learned and used by everyone at the level of higher education (Tedre & Denning, 2016). At that time, different courses started to be included into the education programs for the first time to teach programming to everyone.

DOI: 10.4018/978-1-5225-3200-2.ch001
For instance, in New Zealand, it was started to teach computer programming as a part of Mathematics lesson (Thomson, Bell, Andreae & Robins, 2013). In Germany, computer science (CS) was started to be taught to the students at the level of secondary school as an elective course (Knobelsdorf & Vahrenhold, 2013). The main focus of these lessons was programming and teaching algorithms.

In 1980s, under the leadership of Seymour Papert, students at the level of primary education were attempted to be taught LOGO programming language with the aim of gaining procedural thinking skill. According to Feurzig, who has contributed to the development of LOGO programming language along with Papert, the aim of the development of LOGO is to revolutionize educational understanding rather than teaching Math with procedural programming (Agalianos, Noss, & Whitty, 2001). Although this movement was intriguing for a period, however; it never became very common and disappeared. This situation is justified in different ways in the literature.

Neither the development of technology nor the use of technology in schools progresses according to pre-planned patterns. The progress of technology and its use in the class is shaped by society. Thus, according to Agalianos, Noss, & Whitty (2001), the reason why the 1980s teaching computer programming movement could not be widespread is because behaviorism is the widely accepted educational approach at that time. The LOGO programming language has been designed according to the constructivism. For this reason, it has had difficulty in finding a place in the existing school system. Another justification proposed as the reason why the interest in the movement of teaching LOGO did not reach the desired level is its inability to meet the promises of this approach. No strong evidence of the effect of learning LOGO programming language on children’s thinking skills was found (Kurland, Pea, Clement, & Mawby, 1986; Pea, 1983).

Another factor leading to the end of the approach of teaching programming to everyone is some changes in the field of computer technology after 1980s. There were neither personal computers nor graphical operating systems in the pre-1980 period. Therefore, those who could use computer for any purpose were generally the professionals knowing coding and having a good command of the operating logic of computers. However, in the post-1980 period, particularly in the 90’s, personal computers, graphical operating systems and application software began to spread rapidly. This enabled people who did not have a relationship with the computer science and who did not know the programming language to use the computers for both their daily life needs and their jobs. Correspondingly, in many countries, courses trying to teach computer use as a tool took the place of courses focusing on programming or teaching algorithmic thinking. In these lessons, the aim was to make the students acquire the skills to perform the operations such as searching for information, using word processing programs, preparing presentations, image processing, in a word, computer literacy.

The rapid change in information and communication technologies revealed the limitation of this approach. As application software, hardware etc. taught within the scope of computer literacy changed rapidly, skills learned also lost their importance swiftly. Application software became more user-friendly, so there was no need for individuals to be trained to learn how to use the software. The training given for acquiring computer literacy anyway did not provide the learners with a way to transfer the skills they had acquired from their previous learning to new applications. For example, a student who learned to use a presentation program for computer literacy in a primary school can discover new and more intriguing presentation programs before completing the school, and this does not provide a “migration path” (National Research Council (NRC), 1999; p.2) to learn how to apply this new presentation application. According to NRC (1999, p.2) “[a] better solution is for the individual to plan to adapt to changes in the
Related Content

Robotics and Problem-Based Learning in STEM Formal Educational Environments
[www.igi-global.com/chapter/robotics-and-problem-based-learning-in-stem-formal-educational-environments/88167?camid=4v1a](www.igi-global.com/chapter/robotics-and-problem-based-learning-in-stem-formal-educational-environments/88167?camid=4v1a)

Foundational Processes, Home-School Partnerships, and Culturally Responsive Practices for Dual Language Learners
[www.igi-global.com/chapter/foundational-processes-home-school-partnerships-and-culturally-responsive-practices-for-dual-language-learners/219652?camid=4v1a](www.igi-global.com/chapter/foundational-processes-home-school-partnerships-and-culturally-responsive-practices-for-dual-language-learners/219652?camid=4v1a)

Telehealth Technology and Pediatric Feeding Disorders
[www.igi-global.com/chapter/telehealth-technology-and-pediatric-feeding-disorders/151241?camid=4v1a](www.igi-global.com/chapter/telehealth-technology-and-pediatric-feeding-disorders/151241?camid=4v1a)

The Next Frontier in Public Education: Cyber Charter Schools
[www.igi-global.com/chapter/the-next-frontier-in-public-education/88240?camid=4v1a](www.igi-global.com/chapter/the-next-frontier-in-public-education/88240?camid=4v1a)