Chapter 12
"Computational Thinking" From Pre-K to Graduate Studies to Life:
A Multi-Channel Inventory of Peer-Shared Learning Resources From the Social Web

ABSTRACT

If societies are to advance, they have to be able to harness the capabilities of their technologies in broad and efficient ways. One such technology is computation, writ large, which has crept into so many aspects of modern life. "Computational thinking," as a bridge between people and computation-based and generalizable problem solving, is a multi-step approach that has seeped into pre-K through graduate school in formal (accredited) learning, nonformal (non-accredited) learning, and informal (byproduct) learning. This work explores what peer-shared open-learning resources are available for this approach based on a multi-channel search of the Social Web through social imagery, shared (digital) learning objects, shared slideshows, and social videos.

INTRODUCTION

If societies are to advance, they have to be able to harness the capabilities of their technologies in broad and efficient ways. If they can apply these changes quickly, they can reap first mover advantages. One such technology is computation, writ large, which has crept into so many aspects of modern life as to be pervasive.

A computer scientist suggested that everyone should learn to program "as part of a liberal education" (Perlis, 1962, as cited in Guzdial, Aug. 2008, p. 25). The earliest "computational thinking" reference is credited to Seymour Papert's book *Mindstorms: Children, Computers, and Powerful Ideas* (1980); in this work, computational thinking is defined as "the relationship between programming and thinking skills"

DOI: 10.4018/978-1-7998-6496-7.ch012

to benefit procedural thinking (Zhang & Nouri, 2019, p. 3). A core rationale of CT is that students at all ages will be living "a life heavily influenced by computing" (Barr & Stephenson, Mar. 2011, p. 49), and they should be empowered to engage constructively.

In 2006, Jeannette Wing defined computational thinking as "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry it out. Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution" (Wing, 2006, as cited in Grover & Pea, 2018, p. 21).

Computational thinking (CT) enables…

reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation. Computational thinking is thinking recursively, not linearly. It is parallel processing. It is interpreting code as data and data as code…It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance. (Wing, 2006, p. 33)

Hers is an all-inclusive sense of the harnessing of CT so much so that it becomes "integral to human endeavors" and "disappears as an explicit philosophy" (Wing, 2006, p. 35). Computational thinking is just "good thinking" especially when learners are using well designed technologies given how tools "shape thinking" (Guzdial, Kay, Norris, & Soloway, 2019, p. 28). Indeed, there is "a diversity in definitions, interventions, assessments, and models" regarding computational thinking (CT) (Shute, Sun, & Asbell-Clarke, 2017, p. 142). For some, CT is "a popular phrase that refers to a collection of computational ideas and habits of mind that people in computing disciplines acquire through their work in designing programs, software, simulations, and computations performed by machinery" (Tedre & Denning, 2016, p. 120); it is an idea that has not fully coalesced around a shared meaning.

Perhaps one way to approach CT is to refer to some of its historical understandings. To this end, one work identifies three "important historical currents" as precursors of CT: "evolution of computing's disciplinary ways of thinking and practicing, educational research and efforts in computing, and emergence of computational science and digitalization of society" (Tedre & Denning, 2016, p. 120). This work suggests a close tie to computer sciences but also identifies risks to instantiating CT: "lack of ambition, dogmatism, knowing versus doing, exaggerated claims, narrow views of computing, overemphasis on formulation, (and) losing sight of computational models" (Tedre & Denning, 2016, pp. 125 - 126).

Computer science at the center or the periphery? Another historical approach involves the reference to computer science as a core basis for CT. Some argue to use the clear semantics of computer science as a basis for CT terminology and then transition to new terms for "new domains of investigation" where there are no available computational models (Aho, 2012, p. 832). A precursor of CT is "algorithmic thinking" (Denning, 2009, as cited in Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014, p. 1). CT is a "set of abilities" (Orozco-Garcia, Gonzalez, Montano, Mondragon, & Tobar-Munoz, Nov. 2019, p. 185). "a new problem solving method named for its extensive use of computer science techniques" (Voskoglou & Buckley, Sept. 2012, p. 28). Here, computation serves as "a lens for looking at the world" (Denning, 2009, p. 31). This is not to suggest that computational thinking is unique to computing (it comes from outside the field, too) and CT is "not adequate to portray the whole of the field" (Denning, 2009, p. 31). Another term used for CT is "structural thinking" and "is characterized by design and construction" (He, Hang, & Ding, 2014, p. 817). Others suggest a concern that there may be a restriction in CT to