Book Review

Teaching Digital Technologies & STEM: Computational Thinking, coding and robotics in the classroom

Cummins, K. (2016). *Teaching Digital Technologies & STEM: Computational Thinking, coding and robotics in the classroom.* Retrieved from Amazon.com
133 pp. $4.99 eBook

Michael Kolodziej
Pepperdine University

Abstract

Computational Thinking (CT) has gained increased visibility in curriculum initiatives and discussions in Educational Technology over the past decade. Despite the increased momentum and inclusion of CT in several significant areas of curriculum internationally, little has been published for educators who wish to incorporate these increasingly important computer science related skills, knowledge and attitudes in their classroom practices. *Teaching Digital Technologies & STEM: Computational Thinking, coding and robotics in the classroom* provides practical strategies and a variety of teaching tools for developing Computational Thinking skills in K-12 students.

Keywords: Computational Thinking; robotics; coding; technology

This interactive eBook provides a concise, but media-rich introduction to Computational Thinking and related concepts through the reassuring and nurturing narrative of author Kevin Cummins. The text leverages hyperlinks to videos, websites, and online communities (QR coding is available for hard copy printouts) to provide a strong foundation for readers looking to extend their learning with content and material outside the text as well.

In the introduction, Cummins provides valuable context and a series of value propositions in support of teaching Computational Thinking to students of all types. A world of opportunity exists, in which people of all types can learn to code and solve problems, which benefit the world. The introduction references some of the most notable companies in the world, e.g. Apple, Google and Microsoft, that were built by "'coders' and 'computational thinkers' who possess the skillset to teach a computer to solve a problem it couldn't solve before'" (Introduction, para. 8).

The remaining five chapters of the text mirror the "Five phases of teaching Computational Thinking, Coding and Robotics" which Cummins refers to as his "algorithm for success" (Introduction, para. 4). The steps include:

I. Teaching students to understand data
II. Teaching students to logically approach tasks through "Computational Thinking"
III. Teaching students about digital systems and how they work.
IV. Teaching students to control computers and machines through Coding and robotics.
V. Planning the next steps.
(ibid)

Cummins' algorithm for success and the order of the eBook's chapters model his conceptual understanding of Computational Thinking. According to Cummins, CT is composed of four elements: Decomposition, Pattern recognition, Abstraction, and Algorithm design. Additionally, Computational Thinking manifests in three general areas: Data Visualization and Representation, Digital Information Systems, and Coding and Robotics all of which are implicitly covered within the book.

Though Cummins explicitly states the order of the text should not be taken as an absolute prescription, both the success algorithm and table of contents start with data as a content area. The chapter on data begins with a contextual introduction to data in the modern world, followed by a brief overview of basic data concepts including data types. The chapter concludes by recounting the success of the Oakland Athletics baseball franchise in using Computational Thinking to make personnel decisions. The A's innovative method turned the team from a losing franchise into a division winner. The Boston Red Sox adopted a similar CT-based method to create their World Series winning team in 2004.

The discussion of Computational Thinking that follows reviews the work of Seymour Papert with LOGO and the highlights the role of seminal author Jeannette Wing, providing an introduction to two of the most influential thinkers in the relatively short history of the field. Next, the four elements of Computational thinking are described in sufficient detail for educators to understand the general language and meaning of the computer science related terms underlying CT.

One of the strengths of the book is Cummins' ability to take complex topics and break them down into terms and examples that most people can understand. Computer scientists and others well read in the literature on Computational Thinking may not take away significant amounts of new information. But all readers will likely appreciate how complex terms are discussed and made accessible to the layperson.

The third chapter covers digital information systems and is based on descriptions of the fundamental elements of modern computer systems, processing and hardware. Cummins describes concepts such as input, output and processing, as well as storage (bits, bytes, compression, etc.) and presents all of these topics in the same, easy to understand fashion.

Chapter four is focused on coding and provides an overview in easy to understand language supplemented with ample resources. Platforms such as Scratch, Blockly, Tynker and others are presented as options to engage students in programming for multimedia and game development. Hardware systems and microcontrollers such as Raspberry Pi, and Arduino are presented as options for engaging students in low-cost projects that provide tangible objects for coding and CT exercises. This flows neatly into the second half of the chapter, which is focused on robotics as an extension of coding into the physical world. Although beginning robotics is more expensive and there are fewer viable options for students, basic options such as Lego Robotics, Bee-bots and Lego Mindstorms are all briefly presented.

The final chapter on planning next steps provides several pages of resources linking to various organizations and communities for educators looking to connect with others interested in CT. This chapter is relatively light on prescriptions for Professional development, but provides a good overview of the various options in this area.

Overall, Cummins' book is easy to recommend for its rich coverage of tools and resources suited to those not immersed in Computational Thinking literature. The content within the text is bite-sized and comprehensible for an average reader with little to no formal training in Computer Science, and presented with an upbeat and encouraging voice.

If the text is lacking in anything, it is in specific lesson plans for apply the tools and ideas discussed within. While there is a strong overview of the general language of Computational Thinking and plenty of resources to use, more could be done to situate those tools, and practices to existing classroom lessons and culture by focusing on the pedagogical aspect of constructionism so tightly aligned to the literature of CT pedagogy.