Chapter 15
Computer Programming in Elementary and Middle School:
Connections across Content

Danielle Boyd Harlow
University of California – Santa Barbara, USA

Charlotte Hill
University of California – Santa Barbara, USA

Hilary Dwyer
University of California – Santa Barbara, USA

Ashley Iveland
University of California – Santa Barbara, USA

Alexandria K. Hansen
University of California – Santa Barbara, USA

Anne E. Leak
University of California – Santa Barbara, USA

Diana M. Franklin
University of Chicago, USA

ABSTRACT

Computing has impacted almost all aspects of life, making it increasingly important for the next generation to understand how to develop and use software. Yet, a lack of research on how children learn computer science and an already impacted elementary school schedule has meant that very few children have the opportunity to learn computer science prior to high school. This chapter introduces literature on teaching computer programming to elementary and middle school, highlights three studies that span elementary and middle school, and discusses how programming can be integrated into other content areas and address national standards.

INTRODUCTION

To become the next generation of innovators, today's children must learn to create with technology. To create new technology or innovate on existing technology increasingly requires learning computer programming skills. Computer science in K-12 classrooms is rapidly expanding as technology, programming environments designed for children, and computer science curricula become available. Both the

DOI: 10.4018/978-1-4666-9616-7.ch015

International Society for Technology in Education (ISTE, 2007) and the Computer Science Teachers Association (CSTA) have developed standards relevant to computer programming and computer use.

Teaching computer programming to elementary and middle school children has been facilitated by the development of graphical programming environments that allow children to create programs by dragging and dropping commands (represented as images of blocks) onto a screen, lowering the cognitive barrier to programming (e.g., Maloney, Peppler, Kafai, Resnick, & Rusk, 2008) and increasing novices' interest and excitement in programming (e.g., Malan & Leitner, 2007). These programming environments allow younger students to access computer science ideas without first learning complicated syntax and formatting – attributes of traditional programming. Despite these developments in programming environments and curricula, as well as nationwide efforts to include computer programming at all levels, the vast majority of children do not learn programming in schools prior to high school.

Computer science, when it is taught to younger children, is often limited to outreach events, summer camps, and projects with parents. These informal types of learning experiences have shown success, but they impact only self-selected students, often those already interested in computers and those from middle and upper class families (e.g., Margolis & Fisher, 2003). Integrating computer science into elementary and middle school classrooms, rather than only out-of-school environments is important for addressing equity issues. The current demographics in computer science and computer engineering are indicative of a larger access gap in technology fields. Currently, female students receive a slight majority of all undergraduate degrees nationally, yet they represent only 14.5% of Computer Science degrees awarded. Latino/a students earn only 6.5% of undergraduate computer science degrees while representing 14% of the national population (Computing Research Association, 2013). The disparity in interests and experience with computer programming must be addressed early in students' education. In fact, research indicates that students' reported interest in pursuing a career in science and engineering areas as 8th graders is a strong predictor of whether or not they will pursue a science career (Tai, Liu, Maltese, & Fan, 2006). This means that students' experiences prior to 8th grade are important to recruiting students to careers into science careers, including computer science.

To reach all students, computer science needs to be integrated into the elementary school classroom at the elementary and middle school levels. Yet, integrating computer science into the school day is challenging, in part, because the school day is already full, making adding an additional topic difficult. To be widely integrated into the elementary and middle school curriculum, computer programming activities must address the content standards in mathematics, literacy, and science that teachers are held accountable for in their K-12 classes.

The United States recently made significant changes to educational standards to ensure the nation's graduating students are college and career ready and equipped with the skills and knowledge necessary to be competitive on a global scale. These new standards include the Common Core State Standards (CCSS) for English Language Arts and Mathematics (National Governors Association Center for Best Practices & Council of Chief State & School Officers, 2010) and the Next Generation Science Standards (NGSS) (NGSS Lead States, 2013). All focus on practices of the discipline along with important disciplinary ideas. One important aspect of the NGSS is that, for the first time, engineering design is included in standards. Engineering design is the process by which engineers develop innovative solutions to problems. The process involves understanding the problem, generating ideas, selecting an idea based on multiple constraints, and improving the idea, a process consistent with computer programming.

This shift in educational standards provides an opportunity to consider how computer programming can support students' learning as they work to meet standards in other content areas. Rather than teach-