Chapter 10
Programming Language Learning in K–12 Education

Eric Poitras
Dalhousie University, Canada

Brent Glen Charles Crane
Dalhousie University, Canada

David Dempsey
https://orcid.org/0000-0002-3152-7185
Dalhousie University, Canada

Angela A. Siegel
https://orcid.org/0000-0001-9211-1223
Dalhousie University, Canada

Christine Farion
https://orcid.org/0000-0002-9751-888X
Dalhousie University, Canada

ABSTRACT

This chapter provides a comprehensive review and analysis of recent advancements in both theory and practice concerning programming language learning, specifically within the K-12 educational environment. We synthesize the relevant research literature across the domains of reading, writing, and emerging work integrating both perspectives. Evidence-based pedagogical approaches are reviewed alongside the instructional capabilities offered by educational technology. The chapter concludes by reflecting on the challenges and opportunities of generative artificial

DOI: 10.4018/979-8-3693-9806-7.ch010

intelligence tools. These insights are intended to inform educators and researchers with an interest in skill development within the domain of computer science education, offering practical recommendations for the design of effective instruction to support novice programmers.

INTRODUCTION

Computer science is an academic discipline that encompasses conceptual notions such as algorithms, data structures, programming, and computer systems, among others. Computational thinking, as described by Wing (2006), is an important competency in achieving success in STEM fields, holding significant potential as a means for creative problem-solving and innovation across disciplines. We define computational thinking as broadly applicable problem-solving skills, as conceptualized by Grover and Pea (2018) in terms of abstraction, decomposition, evaluation, pattern recognition, logic, and algorithm design. Thinking computationally involves processes in solving ill-defined problems, understanding, and interpreting data, and communicating information to others using computers.

In this decade, significant efforts have been made to expand computer science education and computational thinking skills within K12 classrooms. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) outlined a framework to define computational thinking skills to guide instruction (CSTA & ISTE, 2011). Much like similar initiatives in STEM, the K-12 Computer Science Framework was developed as a high-level set of guidelines to inform the development of standards, curriculum, course sequence, and professional development opportunities for instructors (K-12 Computer Science Framework, 2016). In 2020, Canada recently formulated a computing curriculum for all elementary and secondary students (Pan-Canadian K-12 Computer Science Framework, 2020).

Learning of these concepts has often been associated with programming or coding, often used interchangeably, and taught as a practice within computer science and commonly associated with computational thinking. From this perspective, programming language learning entails the process of acquiring the knowledge and skills necessary to understand and write code in a specific programming language. It is a fundamental skill in computer science and a tool for supporting and demonstrating the cognitive processes involved in computational thinking (Grover & Pea, 2013). Programming is also quite challenging for learners to become proficient in the skills required to be successful in solving problems (Robins, Rountree, & Rountree, 2003). It is critical that learners can transfer their knowledge and skills to be successful