Computer programming is becoming an essential skill in the 21st century, and in order to best prepare future generations, the promotion of computational thinking and literacy must begin in early childhood education. Computational thinking can be defined in many ways. The broad definition offered in this chapter is that computational thinking practices refer to techniques applied by humans to express themselves by designing and constructing computation. This chapter claims that one of the fundamental ways in which computational thinking can be supported and augmented is by providing children with opportunities to code and to create their own interactive computational media. Thus, computational literacy will allow children to become producers and not only consumers of digital artifacts and systems.

As coding and computer science become established domains in K-2 education, researchers and educators understand that children are learning more than skills when they learn to code – they are learning a new way of thinking and organizing thought. While these new skills are beneficial to future programming tasks, they also support the development of other crucial skills in early childhood education. This chapter explores the ways that coding supports computational thinking in
young children and connects the core concepts of computational thinking to the broader K-2 context.

Chapter 3
Unplugged Learning: Recognizing Computational Thinking in Everyday Life...41
Emily Relkin, Tufts University, USA
Amanda Strawhacker, Tufts University, USA

This chapter explores perspectives on unplugged coding and computational thinking (CT) in early childhood. Concepts, definitions, and research on unplugged learning and its relationship to computer science are considered. Several examples illustrate how young children can encounter powerful ideas of CT in both formal educational settings and in the process of everyday life. Resources are provided that aid in the identification and integration of unplugged activities into early childhood settings. Finally, the authors advocate for further research on teaching CT concepts to children that includes both coding and unplugged approaches.

Section 2
Connections

Chapter 4
The Role of Executive Function and Self-Regulation in the Development of Computational Thinking .......................................................................................64
Elizabeth Kazakoff Myers, WGBH Educational Foundation, USA

This chapter summarizes theoretical connections between computational thinking through learning to code, self-regulation, and executive function and discusses why it is important to continue exploring the intersection of executive function, self-regulation, and computational thinking, including the need to revisit the socio-cultural underpinnings of foundational self-regulation, executive function, and school readiness research. As an example, findings from a 2014 study that explored the relationship between self-regulation and computational thinking when learning to code are shared. Research supports the idea of teaching computational thinking skills within an integrated early childhood curriculum to support the development of well-prepared citizens for the 21st century by drawing on the connections between executive function, self-regulation, and computational thinking.

Chapter 5
Rhyme and Reason: The Connections Among Coding, Computational Thinking, and Literacy...........................................................................................................84
Madhu Govind, Tufts University, USA
Ziva Reimer Hassenfeld, Brandeis University, USA
Laura de Ruiter, Tufts University, USA
The chapter begins with an exploration of computational thinking (CT) and its relationship to computational literacy, followed by a summary of theoretical and empirical work that aims to elucidate the connections among coding, CT, and literacy. The authors argue that these connections thus far have been predominantly one of support (i.e., unidirectional) and motivated by technological and policy advances, as opposed to considering the connections as mutually reinforcing and developmentally coaligned. The authors discuss the coding as another language (CAL) pedagogical approach, a pedagogy that presents learning to program as akin to learning how to use a new language for communicative and expressive functions, emphasizing the bidirectional connections between the two domains. Finally, the authors detail various curricula that use the CAL approach and discuss the implications of CAL for teaching and learning in early childhood.

Chapter 6
Computational Thinking and Life Science: Thinking About the Code of Life .107
Amanda L. Strawhacker, Tufts University, USA

Life science and computer science share the educational goals of fostering students to engage in inquiry-based learning and solve problems through similar practices of discovery, design, and experimentation. This chapter outlines the pedagogical links among traditional life science and emerging computer science domains in early childhood education, and describes an educational intervention using the CRISPEE technological prototype. CRISPEE, designed by a research team of developmentalists, biologists, educators, and computer scientists, invites young children to use computational logic to model design processes with biological materials. Findings are discussed as they relate to new understandings about how young children leverage computational thinking when engaged in design-based life science, or bioscience.

Chapter 7
Computational Expression: How Performance Arts Support Computational Thinking in Young Children ..............................................................134
Amanda L. Strawhacker, Tufts University, USA
Amanda A. Sullivan, Tufts University, USA

In the past two decades, STEM education has been slowly replaced by “STEAM,” which refers to learning that integrates science, technology, engineering, arts, and mathematics. The added “Arts” portion of this pedagogical approach, although an important step towards integrated 21st century learning, has long confused policymakers, with definitions ranging from visual arts to humanities to art education and more. The authors take the position that Arts can be broadly interpreted to mean any approach that brings interpretive and expressive perspectives to STEM activities. In this chapter, they present illustrative cases inspired by work in real learning
settings that showcase how STEAM concepts and computational thinking skills can support children’s engagement in cultural, performing, and fine arts, including painting, sculpture, architecture, poetry, music, dance, and drama.

Section 3

Contexts

Chapter 8
Fostering Computational Thinking in Homes and Other Informal Learning Spaces ................................................................. 158

Madhu Govind, Tufts University, USA

This chapter provides theoretical and practical insights for fostering children’s computational thinking (CT) in homes and other family-friendly spaces such as libraries, museums, and after-school programs. The family context—the kinds of roles, interactions, and opportunities afforded by parents, caregivers, and siblings—is essential for understanding how young children learn and engage in CT. This work is informed by research on how everyday activities and educational technologies (and the contexts in which they are used) can be designed to promote opportunities for CT and family engagement. This chapter discusses ways to support children’s CT by co-engaging family members in collaborative coding activities in homes and other informal learning spaces.

Chapter 9

Makerspaces as Learning Environments to Support Computational Thinking .176

Amanda L. Strawhacker, Tufts University, USA
Miki Z. Vizner, Independent Researcher, USA

Makerspaces are technology-rich learning environments that can uniquely support children’s development. In education communities, makerspaces have become sites to take up explorations of personally-motivated problem solving, and have been tied to 21st century learning outcomes of perseverance, creativity, persistence, and computational thinking. Elsewhere in this book, Bers described computational thinking as the set of skills and cognitive processes required to give instructions for a specific task in such a way that a computer could carry it out. But Bers also argued that the purpose of computational thinking is to cultivate a fluency with technological tools as a medium of expression, not an end in itself. Computational making is part of this expression. This chapter explores the ways in which tools, facilitation, and the physical environment can support children’s engagement with powerful ideas of computational thinking through making.
Chapter 10
Coding, Computational Thinking, and Cultural Contexts .............................................. 201
 Libby Hunt, Tufts University, USA
 Marina Umaschi Bers, Tufts University, USA

This chapter examines the relationship between coding, computational thinking, and the contexts in which those concepts are learned. It recounts a pilot study where a 12-week robotics curriculum was taught in kindergarten classrooms at eight interfaith and secular schools in Boston, United States of America and Buenos Aires, Argentina. In this chapter, the authors explore how teachers and students drew from their socio-cultural environments to inform the language of computational thinking and support the internalization of computational concepts and, in turn, how computational thinking was used as a tool for deeper exploration of cultural traditions and beliefs, meaning-making, and creative expression.

Chapter 11
Supporting Girls’ Computational Thinking Skillsets: Why Early Exposure Is Critical to Success ................................................................................................................................................ 216
 Amanda Sullivan, Tufts University, USA

The representation of women in technical fields such as computer science and engineering continues to be an issue in the United States, despite decades of research and interventions. According to the most recent Bureau of Labor Statistics reports, only 21.1% of computer programmers are women, and only 16.5% of engineering and architecture positions are filled by women. This chapter discusses the long-term importance of exposing girls to computational thinking during their formative early childhood years (Kindergarten through second grade) in order to set them up for equal opportunities in technical fields throughout their later educational and career years. This chapter presents a case example of a K-2nd grade robotics and coding curriculum in order to highlight examples of developmentally appropriate technologies, activities, and strategies that educators can implement to foster young girls’ computational thinking skills. Best practices and instructional strategies to support girls—as well as young children of any gender identity—are discussed.

Chapter 12
Including Students With Disabilities in the Coding Classroom .............................................. 236
 Tess Levinson, Tufts University, USA
 Libby Hunt, Tufts University, USA
 Ziva Hassenfeld, Brandeis University, USA

This chapter discusses understandings of coding and computational thinking education for students with disabilities. The chapter describes the special education system in the United States, including limitations in how computer science education is made
available to students receiving special education services. The chapter then provides a summary of research in computer science education for students with disabilities, including both high-incidence and low-incidence disabilities. A case study of a young student with a mild disability learning in a general education computational thinking program is then presented, and the implications of the case study for future research directions are discussed.

Section 4
Evaluation

Chapter 13
TechCheck: Creation of an Unplugged Computational Thinking Assessment for Young Children ...............................................................250
Emily Relkin, Tufts University, USA

This chapter describes the development and validation of TechCheck, a novel instrument for rapidly assessing computational thinking (CT) skills in 5-9 years old children. TechCheck assessments can be administered in classroom or online settings regardless of whether students have prior knowledge of coding. This assessment probes six domains of CT described by Bers as developmentally appropriate for young children including algorithms, modularity, control structures, representation, hardware/software, and debugging. TechCheck demonstrates good psychometric properties and can readily distinguish among young children with different CT abilities.

Chapter 14
Examining Young Children’s Computational Artifacts .........................265
Apittha Unahalekhaka, Tufts University, USA
Madhu Govind, Tufts University, USA

Computational thinking (CT), in line with the constructionist perspective, is often best displayed when children have the opportunity to demonstrate their skills by producing creative coding artifacts. Performance-based or project portfolio assessments of young children’s coding artifacts are a rich and useful approach to explore how children develop and apply CT abilities. In this chapter, the authors examine various rubrics and assessment tools used to measure the levels of programming competency, creativity, and purposefulness displayed in students’ coding artifacts. The authors then discuss the development of ScratchJr and KIBO project rubrics for researchers and educators, including examples to illustrate how these highly diverse projects provide insight into children’s CT abilities. Finally, the authors conclude with implications and practical strategies for using rubrics in both educational and research settings.
Chapter 15
Insights Into Young Children’s Coding With Data Analytics

Apittha Unahalekhaka, Tufts University, USA
Jessica Blake-West, Tufts University, USA
XuanKhanh Nguyen, Tufts University, USA

Over the past decade, there has been a growing interest in learning analytics for research in education and psychology. It has been shown to support education by predicting learning performances such as school completion and test scores of students in late elementary and above. In this chapter, the authors discuss the potential of learning analytics as a computational thinking assessment in early childhood education. They first introduce learning analytics by discussing its various applications and the benefits and limitations that it offers to the educational field. They then provide examples of how learning analytics can deepen our understanding of computational thinking through observing young children’s engagement with ScratchJr: a tablet coding app designed for K-2 students. Finally, they close this chapter with future directions for using learning analytics to support computer science education.

Compilation of References

About the Contributors

Index