Computational Thinking And Coding For Every Student The Teacheraeurtms Getting Started Guide

Coding as a Playground is the first book to focus on how young children (ages 7 and under) can engage in computational thinking and be taught to become computer programmers, a process that can increase both their cognitive and social-emotional skills. Readers will learn how coding can engage children as producers—and not merely consumers—of technology in a playful way. You will come away from this groundbreaking work with an understanding of how coding promotes developmentally appropriate experiences such as problem solving, imagination, cognitive challenges, social interactions, motor skills development, emotional exploration, and making different choices. You will also learn how to integrate coding into different curricular areas to promote literacy, math, science, engineering, and the arts through a project-based approach.

The education system is constantly growing and developing as more ways to teach and learn are implemented into the classroom. Recently, there has been a growing interest in teaching computational thinking with schools all over the world introducing it to the curriculum due to its ability to allow students to become proficient at problem solving using logic, an essential life skill. In order to provide the best education possible, it is imperative that computational thinking strategies, along with programming skills and the use of robotics in the classroom, be implemented in order for students to achieve maximum thought processing skills and computer competencies. The Research Anthology on Computational Thinking, Programming, and Robotics in the Classroom is an all-encompassing reference book that discusses how computational thinking, programming, and robotics can be used in education as well as the benefits and difficulties of implementing these elements into the classroom. The book includes strategies for preparing educators to teach computational thinking in the classroom as well as design techniques for incorporating these practices into various levels of school curriculum and within a variety of subjects. Covering topics ranging from decomposition to robot learning, this book is ideal for educators, computer scientists, administrators, academicians, students, and anyone interested in learning more about how computational thinking, programming, and robotics can change the current education system.

Applied Computational Thinking with Python provides a hands-on approach to implementation and associated methodologies that will have you up-and-running, and productive in no time. Developers working with Python will be able to put their knowledge to work with this practical guide using the computational thinking method for problem-solving.

In this revolutionary book, a renowned computer scientist explains the importance of teaching children the basics of computing and how it can prepare them to succeed in the ever-evolving tech world. Computers have completely changed the way we teach children. We have Mindstorms to thank for that. In this book, pioneering computer scientist Seymour Papert uses the invention of LOGO, the first child-friendly programming language, to make the case for the value of teaching children with computers. Papert argues that children are more than capable of mastering computers, and that teaching computational processes like de-bugging in the classroom can change the way we learn everything else. He also shows that schools saturated with technology can actually improve socialization and interaction among students and between students and teachers. Technology changes every day, but the basic ways that computers can help us learn remain. For thousands of teachers and parents who have sought creative ways to help children learn with computers, Mindstorms is their bible.

What will you learn from this book? It’s no secret the world around you is becoming more connected, more configurable, more programmable, more computational. You can remain a passive participant, or you can learn to code. With Head First Learn to Code you’ll learn how to think computationally and how to write code to make your computer, mobile device, or anything with a CPU do things for you. Using the Python programming language, you’ll learn step by step the core concepts of programming as well as many fundamental topics from computer science, such as data structures, storage, abstraction, recursion, and modularity. Why does this book look so different? Based on the latest research in cognitive science and learning theory, Head First Learn to Code uses a visually rich format to engage your mind, rather than a text-heavy approach that puts you to sleep. Why waste your time struggling with new concepts? This multi-sensory learning experience is designed for the way your brain really works.

Mobile technologies combined with an interdisciplinary approach to knowledge and organization of learning experiences that are meaningful to children could create a creative and interactive learning environment different from that of traditional teaching. Making good use of mobile learning with appropriate devices will increase the learning motivations of the students and help them bring about positive performance. Mobile Learning Applications in Early Childhood Education is a collection of innovative research on the methods and applications of mobile learning techniques and strategies within diversified teaching settings. While highlighting topics including computational thinking, ubiquitous learning, and social development, this book is ideally designed for researchers, teachers, parents, curriculum developers, instructional designers, academicians, students, and practitioners seeking current research on the application of mobile technology within child education.

While the growth of computational thinking has brought new awareness to the importance of computing education, it has also created new challenges. Many educational initiatives focus solely on the programming aspects, such as variables, loops, conditionals, parallelism, operators, and data handling, divorcing computing from real-world contexts and applications. This decontextualization threatens to make learners believe that they do not need to learn computing, as they cannot envision a future in which they will need to use it, just as many see math and physics education as unnecessary. The Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education is a cutting-edge research publication that examines the implementation of computational thinking into school curriculum in order to develop creative problem-solving skills and to build a computational identity which will allow for future STEM growth. Moreover, the book advocates for a new approach to computing education that argues that while learning about computing, young people should also have opportunities to create with computing, which will have a direct impact on their lives and their communities. Featuring a wide range of topics such as assessment, digital teaching, and educational robotics, this book is ideal for academicians, instructional designers, teachers, education professionals, administrators, researchers, and students.

This new edition of the popular book No Fear Coding offers new research, updated tools and more cross-curricular connections for K-5 teachers to integrate into their classes. Coding has become an essential skill for finding solutions to everyday problems, while computational thinking (CT) teaches reasoning and creativity, and offers an innovative approach to demonstrating content knowledge and seeing mathematical processes in action. No-Fear Coding introduced many K-5 educators to ways to bring coding into their curriculum by embedding computational thinking skills into activities for different content areas. The new, expanded edition of this popular book features updated tools and resources, with more discussion about the features of each resource and the concepts each one can teach. It incorporates the latest research on computational thinking and deepens coverage of the ISTE Standards for Students. Also new in this edition: Suggestions for extending CT to more subjects, such as music, art and physical education, and an explanation of how CT can be used in special education. Expanded coverage of teaching CT offline to help students apply it without digital technology. Ideas on how to alleviate fear about the subject matter, as well as how a busy educator might incorporate more content into their already intense curriculum. Insights into helping children become active creators rather than passive users of technology, especially important today as we spend hours on devices and many children face challenges with anxiety and ADHD. Discussion of how coding and CT help children develop the executive functioning skills that are critical in early childhood. Tips on demystifying basic coding concepts so that teachers are comfortable teaching these concepts to their students. No-Fear Coding, Second Edition will help build students' coding and CT knowledge to prepare them for the middle grades and beyond.

How-to books related to computer science (CS) and teaching CS in K-12 environments are often either step-by-step guides or reference books, with little or no connection to pedagogy. By contrast, Coding + Math offers the analytical foundation teachers need to inform their practice, specifically in mathematics. This book will serve as a deep dive into CS integration for elementary teachers, providing guidelines for designing integrated CS/math curricula through case studies and practical examples. Grounded in research, the book's mini-lessons contrast visual-based coding with text-based programming and provide guidance in the selection and creation of lessons, instructional materials and CS platforms to help educators prepare students for the careers of the future.

An introduction to computational thinking that traces a genealogy beginning centuries before the digital computer. A few decades into the digital era, scientists discovered that thinking in terms of computation made possible an entirely new way of organizing scientific investigation; eventually, every field had a computational branch: computational physics, computational biology, computational sociology. More recently, “computational thinking” has become part of the K–12 curriculum. But what is computational thinking? This volume in the MIT Press Essential Knowledge series offers an accessible overview, tracing a genealogy that begins centuries before digital computers and portraying computational thinking as pioneers of computing have described it. The authors explain that computational thinking (CT) is not a set of concepts for programming; it is a way of thinking that is honed through practice: the mental skills for designing computations to do jobs for us, and for explaining and interpreting the world as a complex of information processes. Mathematically trained experts (known as “computers”) who performed complex calculations as teams engaged in CT long before electronic computers. The authors identify six dimensions of today's highly developed CT—methods, machines, computing education, software engineering, computational science, and design—and cover each in a chapter. Along the way, they debunk inflated claims for CT and computation while making clear the power of CT in all its complexity and multiplicity.

Essential Computational Thinking: Computer Science from Scratch helps students build a theoretical and practical foundation for learning computer science. Rooted in fundamental science, this text defines elementary ideas including data and information, quantifies these ideas mathematically, and, through key concepts in physics and computation, demonstrates the relationship between computer science and the universe itself. In Part I, students explore the theoretical underpinnings of computer science in a wide-ranging manner. Readers receive a robust overview of essential computational theories and programming ideas, as well as topics that examine the mathematical and physical foundations of computer science. Part 2 presents the basics of computation and underscores programming as an invaluable tool in the discipline. Students can apply their newfound knowledge and begin writing substantial programs immediately. Finally, Part 3 explores more sophisticated computational ideas, including object-oriented programing, databases, data science, and some of the underlying principles of machine learning. Essential Computational Thinking is an ideal text for a firmly technical CS0 course in computer science. It is also a valuable resource for highly-motivated non-computer science majors at the undergraduate or graduate level who are interested in learning more about the discipline for either professional or personal development.

This core text for trainee primary teachers is a guide to the teaching of computing and coding, and provides an exploration of how children develop their computational thinking.

Computational technologies have been impacting human life for years. Teaching methods must adapt accordingly to provide the next generation with the necessary knowledge to further advance these human-assistive technologies. Teaching Computational Thinking in Primary Education is a crucial resource that examines the impact that instructing with a computational focus can have on future learners. Highlighting relevant topics that include multifaceted skillsets, coding, programming methods, and digital games, this scholarly publication is ideal for educators, academicians, students, and researchers who are interested in discovering how the future of education is being shaped.

Coding and computational thinking (the ability to think like a computer) are among the skills that will serve students well in the future. Coding goes beyond websites and software - it's an essential component in finding solutions to everyday problems. Computational thinking has many applications beyond the computer lab or math class -it teaches reasoning, creativity and expression, and is an innovative way to demonstrate content knowledge and see mathematical processes in action. No-Fear Coding shows K-5 educators how to bring coding into their curriculum by embedding computational thinking skills into activities for every content area. At the same time, embedding these skills helps students prepare for coding in the middle grades as they build their knowledge. To help teachers easily and effectively introduce coding, the book features: Classroom-tested lessons and activities designed for skills progression. Ready-to-implement coding exercises that can be incorporated across the curriculum. Alignment to ISTE and Computer Science Teachers Association (CSTA) standards. Case studies and explorations of technology tools and resources to teach coding.

A hands-on, problem-based introduction to building algorithms and data structures to solve problems with a computer. Algorithmic Thinking will teach you how to solve challenging programming problems and design your own algorithms. Daniel Zingaro, a master teacher, draws his examples from world-class programming competitions like USACO and IOI. You'll learn how to classify problems, choose data structures, and identify appropriate algorithms. You'll also learn how your choice of data structure, whether a hash table, heap, or tree, can affect runtime and speed up your algorithms; and how to adopt powerful strategies like recursion, dynamic programming, and binary search to solve challenging problems. Line-by-line breakdowns of the code will teach you how to use algorithms and data structures like: • The breadth-first search algorithm to find the optimal way to play a board game or find the best way to translate a book • Dijkstra's algorithm to determine how many mice can exit a maze or the number of fastest routes between two locations • The union-find data structure to answer questions about connections in a social network or determine who are friends or enemies • The heap data structure to determine the amount of money given away in a promotion • The hash-table data structure to determine whether snowflakes are unique or identify compound words in a dictionary NOTE: Each problem in this book is available on a programming-judge website. You'll find the site's URL and problem ID in the description. What's better than a free correctness check?

This Handbook describes the extent and shape of computing education research today. Over fifty leading researchers from academia and industry (including Google and Microsoft) have contributed chapters that together define and expand the evidence base. The foundational chapters set the field in context, articulate expertise from key disciplines, and form a practical guide for new researchers. They address what can be learned empirically, methodologically and theoretically from each area. The topic chapters explore issues that are of current interest, why they matter, and what is already known. They include discussion of motivational context, implications for practice, and open questions which might suggest future research. The authors provide an authoritative introduction to the field and is essential reading for policy makers, as well as both new and established researchers.

This book broadly educates preservice teachers and scholars about current research on computational thinking (CT). More specifically, attention is given to computational algorithmic thinking (CAT), particularly among underrepresented K–12 student groups in STEM education. Computational algorithmic thinking (CAT)—a precursor to CT—is explored in this text as the ability to design, implement, and evaluate the application of algorithms to solve a variety of problems. Drawing on observations from research studies that focused on innovative STEM programs, including underrepresented students in rural, suburban, and urban contexts, the authors reflect on project-based learning experiences, pedagogy, and evaluation that are conducive to developing advanced computational thinking, specifically among diverse student populations. This practical text includes vignettes and visual examples to illustrate how coding, computer modeling, robotics, and drones may be used to promote CT and CAT among students in diverse classrooms.

Computational thinking (CT) is a powerful ingredient for solving ambiguous, complex and open-ended problems by drawing on principles and practices central to computer science (CS). CT is at the core of CS and is a gateway to sparking student interest and confidence in learning CS. The ISTE Standards for Educators: Computational Thinking Competencies were created to inspire every educator to add more computational thinking into their core problem-solving strategies. These competencies augment and home in on the competencies embedded in the ISTE Standards for Students and Educators.The information in this guide will provide a framework and structure to build creative learning opportunities in CT and increase students' ability to adapt to unfamiliar challenges, allowing for more success with innovative lesson plans.

Learn approaches of computational thinking and the art of designing algorithms. Most of the algorithms you will see in this book are used in almost all software that runs on your computer. Learning how to program can be very rewarding. It is a special feeling to seeing a computer translate your thoughts into actions and see it solve your problems for you. To get to that point, however, you must learn to think about computations in a new way—you must learn computational thinking. This book begins by discussing models of the world and how to formalize problems. This leads onto a definition of computational thinking and putting computational thinking in a broader context. The practical coding in the book is carried out in Python; you'll get an introduction to Python programming, including how to set up your development environment. What You Will Learn Think in a computational way Acquire general techniques for problem solving See general and concrete algorithmic techniques Program solutions that are both computationally efficient and maintainable Who This Book Is For Those new to programming and computer science who are interested in learning how to program algorithms and working with other computational aspects of programming.

"This book starts with an introduction to the topic of computational thinking and young children and then presents chapters of different aspects to consider in teaching computational thinking to young children,"--

"Code is the 21st century literacy and the need for people to speak the ABCs of Programming is imminent." --Linda Liukas Meet Ruby--a small girl with a huge imagination. In Ruby's world anything is possible if you put your mind to it. When her dad asks her to find five hidden gems Ruby is determined to solve the puzzle with the help of her new friends, including the Wise Snow Leopard, the Friendly Foxes, and the Messy Robots. As Ruby stomps around her world kids will be introduced to the basic concepts behind coding and programming through storytelling. Learn how to break big problems into small problems, repeat tasks, look for patterns, create step-by-step plans, and think outside the box. With hands-on activities included in every chapter, future coders will be thrilled to put their own imaginations to work.

"Unlike other robotics books and curriculum, Rev Up Robotics takes a cross-curricular approach, showing educators how to begin incorporating robotics in tandem with computational thinking into content area lessons or adapting for electives. The book meets readers where they are and is arranged in three major parts. Part 1 covers the basics, defining robotics and sharing real-world applications along with how to teach foundational skills for computational thinking and computer science. Part 2 shows robotics in practice within the context of content areas and features lesson plans mapped to academic and technology standards, including the ISTE Standards and the Computer Science Teachers Association Standards. Part 3 offers advice on pedagogy and teaching strategies backed by research from the learning sciences, and shares approaches to teaching robotics using project-based learning or as part of after-school clubs or robotics competitions. Included in the book are programming considerations, including a pathway from working with visual blocks to programming in C++ and K-8 applicable resources from leading organizations, including Carnegie Mellon, LEGO Education, littleBits, Ozobot, VEX Robotics, Code.org and NASA. The book also features actionable steps, pro tips and resources for getting started, improving practice and preparing students for computational thinking, programming, core coding concepts and computer science fundamentals. The goal of Rev Up Robotics is to provide an evergreen professional development resource that both teachers and schools can use to discover how to incorporate computational thinking, robotics and computer science into lessons that engage students and activate learning"--

Empower tomorrow's tech innovators Our students are avid users and consumers of technology. Isn't it time that they see themselves as the next technological innovators, too? Computational Thinking and Coding for Every Student is the beginner's guide for K-12 educators who want to learn to integrate the basics of computer science into their curriculum. Readers will find Strategies and activities for teaching computational thinking and coding inside and outside of school, at any grade level, across disciplines Instruction-ready lessons for every grade A discussion guide and companion website with videos, activities, and other resources

Computational Thinking in Compulsory Education explores the relevance of computational thinking in primary and secondary education. As today's school-aged students prepare to live and work in a thoroughly digitized world, computer science is providing a wealth of new learning concepts and opportunities across domains. This book offers a comprehensive overview of computational thinking, its history, implications for equity and inclusion, analyses of competencies in practice, and integration into learning, instruction, and assessment through scaffolded teacher education. Computer science education faculty and pre- and in-service educators will find a fresh pedagogical approach to computational thinking in primary and secondary classrooms.

This book helps classroom teachers, in several core content areas, develop activities and projects to encourage computational thinking and coding skills, and to build bridges between those skills and practice.

Recent innovations and new technologies in education have altered the way teachers approach instruction and learning and can provide countless advantages. The pedagogical value of specific technology tools and the cumulative effects of technology exposure on student learning over time are two areas that need to be explored to better determine the improvements needed in the modern classroom. Advanced Methodologies and Technologies in Modern Education Delivery provides emerging research on educational models in the continually improving classroom. While highlighting the challenges facing modern in-service and pre-service teachers when educating students, readers will learn information on new methods in curriculum development, instructional design, and learning assessments to implement within their classrooms. This book is a vital resource for pre-service and in-service teachers, teacher education professionals, higher education administrative professionals, and researchers interested in new curriculum development.

This book covers studies of computational thinking related to linking, infusing, and embedding computational thinking elements to school curricula, teacher education and STEM related subjects. Presenting the distinguished and exemplary works by educators and researchers in the field highlighting the contemporary trends and issues, creative and unique approaches, innovative methods, frameworks, pedagogies and theoretical and practical aspects in computational thinking. A decade ago the notion of computational thinking was introduced by Jeannette Wing and envisioned that computational thinking will be a fundamental skill that complements to reading, writing and arithmetic for everyone and represents a universally applicable attitude. The computational thinking is considered a thought processes involved in a way of solving problems, designing systems, and understanding human behaviour. Assimilating computational thinking at young age will assist them to enhance problem solving skills, improve logical reasoning, and advance analytical ability - key attributes to succeed in the 21st century. Educators around the world are investing their relentless effort in equipping the young generation with real-world skills ready for the demand and challenges of the future. It is commonly believed that computational thinking will play a pivotal and dominant role in this endeavour. Wide-ranging research on and application of computational thinking in education have been emerged in the last ten years. This book will document attempts to conduct systematic, prodigious and multidisciplinary research in computational thinking and present their findings and accomplishments.

Coding as a Playground, Second Edition focuses on how young children (aged 7 and under) can engage in computational thinking and be taught to become computer programmers, a process that can increase both their cognitive and social-emotional skills. Learn how coding can engage children as producers—and not merely consumers—of technology in a playful way. You will come away from this groundbreaking work with an understanding of how coding promotes developmentally appropriate experiences such as problem-solving, imagination, cognitive challenges, social interactions, motor skills development, emotional exploration, and making different choices. Featuring all-new case studies, vignettes, and projects, as well as an expanded focus on teaching coding as a new literacy, this second edition helps you learn how to integrate coding into different curricular areas to promote literacy, math, science, engineering, and the arts through a project-based approach and a positive attitude to learning.

Computational thinking (CT) is a timeless, transferable skill that enables you to think more clearly and logically, as well as a way to solve specific problems. With this book you'll learn to apply computational thinking in the context of software development to give you a head start on the road to becoming an experienced and effective programmer.

Report of a Workshop on the Scope and Nature of Computational Thinking presents a number of perspectives on the definition and applicability of computational thinking. For example, one idea expressed during the workshop is that computational thinking is a fundamental analytical skill that everyone can use to help solve problems, design systems, and understand human behavior, making it useful in a number of fields. Supporters of this viewpoint believe that computational thinking is comparable to the linguistic, mathematical and logical reasoning taught to all children. Various efforts have been made to introduce K-12 students to the most basic and essential computational concepts and college curricula have tried to provide a basis for life-long learning of increasingly new and advanced computational concepts and technologies. At both ends of this spectrum, however, most efforts have not focused on fundamental concepts. The book discusses what some of those fundamental concepts might be. Report of a Workshop on the Scope and Nature of Computational Thinking explores the idea that as the use of computational devices is becoming increasingly widespread, computational thinking skills should be promulgated more broadly. The book is an excellent resource for professionals in a wide range of fields including educators and scientists.

This This book is open access under a CC BY 4.0 license. This book offers a comprehensive guide, covering every important aspect of computational thinking education. It provides an in-depth discussion of computational thinking, including the notion of perceiving computational thinking practices as ways of mapping models from the abstraction of data and process structures to natural phenomena. Further, it explores how computational thinking education is implemented in different regions, and how computational thinking is being integrated into subject learning in K-12 education. In closing, it discusses computational thinking from the perspective of STEM education, the use of video games to teach computational thinking, and how computational thinking is helping to transform the quality of the workforce in the textile and apparel industry. This work was published by Saint Philip Street Press pursuant to a Creative Commons license permitting commercial use. All rights not granted by the work's license are retained by the author or authors.

Why every child needs to learn to code: the shift from “computational thinking” to computational participation. Coding, once considered an arcane craft practiced by solitary techies, is now recognized by educators and theorists as a crucial skill, even a new literacy, for all children. Programming is often promoted in K-12 schools as a way to encourage “computational thinking”—which has now become the umbrella term for understanding what computer science has to contribute to reasoning and communicating in an ever-increasingly digital world. In Connected Code, Yasmin Kafai and Quinn Burke argue that although computational thinking represents an excellent starting point, the broader conception of “computational participation” better captures the twenty-first-century reality. Computational participation moves beyond the individual to focus on wider social networks and a DIY culture of digital “making.” Kafai and Burke describe contemporary examples of computational participation: students who code not for the sake of coding but to create games, stories, and animations to share; the emergence of youth programming communities; the practices and ethical challenges of remixing (rather than starting from scratch); and the move beyond stationary screens to programmable toys, tools, and textiles.

The new edition of an introductory text that teaches students the art of computational problem solving, covering topics ranging from simple algorithms to information visualization. This book introduces students with little or no prior programming experience to the art of computational problem solving using Python and various Python libraries, including PyLab. It provides students with skills that will enable them to make productive use of computational techniques, including some of the tools and techniques of data science for using computation to model and interpret data. The book is based on an MIT course (which became the most popular course offered through MIT's OpenCourseWare) and was developed for use not only in a conventional classroom but in in a massive open online course (MOOC). This new edition has been updated for Python 3, reorganized to make it easier to use for courses that cover only a subset of the material, and offers additional material including five new chapters. Students are introduced to Python and the basics of programming in the context of such computational concepts and techniques as exhaustive enumeration, bisection search, and efficient approximation algorithms. Although it covers such traditional topics as computational complexity and simple algorithms, the book focuses on a wide range of topics not found in most introductory texts, including information visualization, simulations to model randomness, computational techniques to understand data, and statistical techniques that inform (and misinform) as well as two related but relatively advanced topics: optimization problems and dynamic programming. This edition offers expanded material on statistics and machine learning and new chapters on Frequentist and Bayesian statistics.

This book offers a gentle motivation and introduction to computational thinking, in particular to algorithms and how they can be coded to solve significant, topical problems from domains such as finance, cryptography, Web search, and data compression. The book is suitable for undergraduate students in computer science, engineering, and applied mathematics, university students in other fields, high-school students with an interest in STEM subjects, and professionals who want an insight into algorithmic solutions and the related mindset. While the authors assume only basic mathematical knowledge, they uphold the scientific rigor that is indispensable for transforming general ideas into executable algorithms. A supporting website contains examples and Python code for implementing the algorithms in the book.

"This new edition of the popular book No Fear Coding offers new research, updated tools and more cross-curricular connections for K-5 teachers to integrate into their classes. Coding has become an essential skill for finding solutions to everyday problems, while computational thinking (CT) teaches reasoning and creativity, and offers an innovative approach to demonstrating content knowledge and seeing mathematical processes in action. No-Fear Coding introduced many K-5 educators to ways to bring coding into their curriculum by embedding computational thinking skills into activities for different content areas. The new, expanded edition of this popular book features updated tools and resources, with more discussion about the features of each resource and the concepts each one can teach. It incorporates the latest research on computational thinking and deepens coverage of the ISTE Standards for Students. Also new in this edition: Suggestions for extending CT to more subjects, such as music, art and physical education, and an explanation of how CT can be used in special education. Expanded coverage of teaching CT offline to help students apply it without digital technology. Ideas on how to alleviate fear about the subject matter, as well as how a busy educator might incorporate more content into their already intense curriculum. Insights into helping children become active creators rather than passive users of technology, especially important today as we spend hours on devices and many children face challenges with anxiety and ADHD. Discussion of how coding and CT help children develop the executive functioning skills that are critical in early childhood. Tips on demystifying basic coding concepts so that teachers are comfortable teaching these concepts to their students. No-Fear Coding, Second Edition will help build students' coding and CT knowledge to prepare them for the middle grades and beyond"--