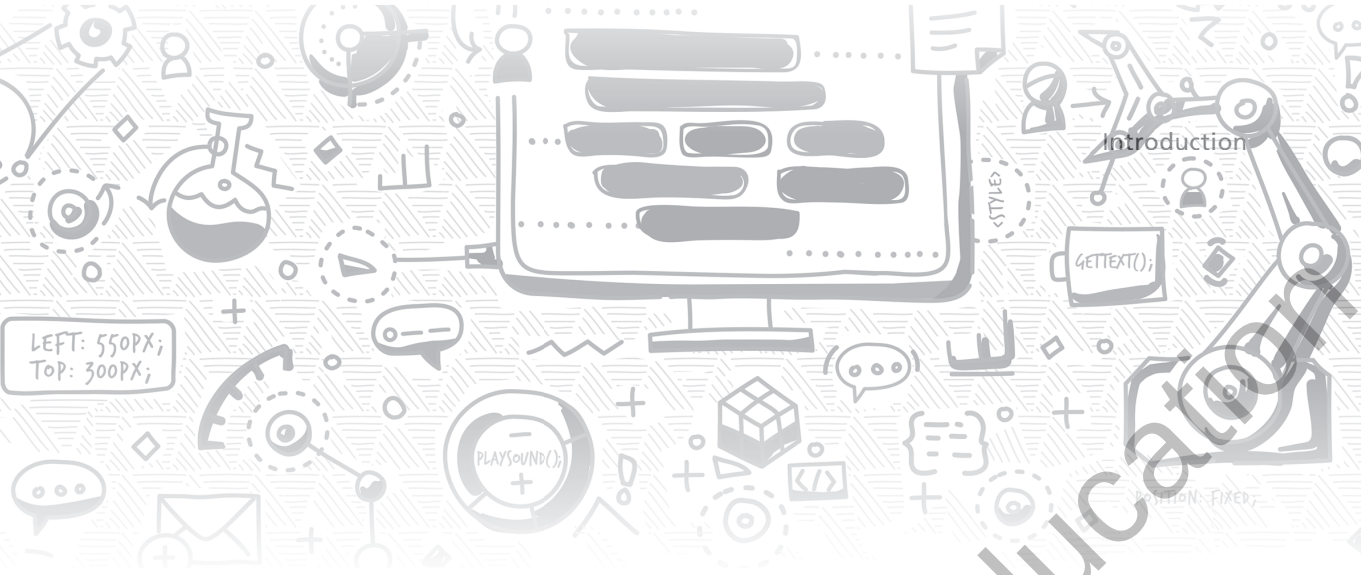
Contents

IST8061 • 9781760568061 • © 2019 Hawker Brownlow Education

IST8061 • 9781760568061 • © 2019 Hawker Brownlow Education



Introduction

I came to teaching as a second career. When I first met with my graduate-school adviser, I knew little more than that I enjoyed working with kids, and I enjoyed middle-school students in particular. What I didn't yet know was ***what*** I wanted to teach, or what I even ***could*** teach. I had been a theater major in college, but theater positions aren't available frequently.

I'd been working in IT and had done a little coding for both work and pleasure. I thought computer science might be a subject for which I'd have a basic skill set and enjoy teaching, but I was told there was no such thing as a computer science endorsement. Computer science teachers were some sort of mystical entity—the professors in my program knew they existed, but nobody could tell me how to become one. After a fairly long discussion of my skills and interests, and an in-depth review of my college transcripts (I needed enough credits to be considered "highly qualified" in my chosen endorsement), we landed on an English Language Arts (ELA) endorsement.

I loved teaching English, and I like to believe I was pretty good at it. But despite my love for the subject, I had a gnawing voice in the back of my mind. I knew that I

had benefited from my experience with computer science. It had changed the way I thought, empowered me to create and solve interesting problems, and it was something I enjoyed, but it wasn't something to which my students had any access. I had the privilege of parents who supported my interests, but access to something as powerful and transformative as computer science shouldn't be predicated on privilege. The students who get access shouldn't be only those whose families can afford expensive coding camps. What's the point of public education if not to ensure that every child enters adulthood equipped with the tools to engage, contribute, and succeed in society? Today, that must mean that students are at least literate in computer science, and if the school district wasn't going to make that happen, then I had to.

Starting with small projects, I incorporated programming and computational thinking into my ELA classes. I started by simply swapping out the medium I used for assignments: a webpage instead of a PowerPoint, or a game instead of a poster. To be honest, in these early attempts the computer science projects were more a carrot for engagement than a core component of the learning. Over time, however, I sought the intersection between the skills students needed to be successful in ELA and the skills they were developing through coding. As I created lessons and activities that integrated computational thinking and computer science concepts with the Common Core ELA standards, I found opportunities to make my instruction relevant to more students. After all, most of us likely interact with phones more than poetry.

Over the years, I gradually transitioned to a position as a full-time computer science teacher, and eventually a computer science curriculum developer and teacher facilitator, but I've never forgotten that moment when my students first made the connection between what they believed to be two disparate content areas.

The landscape has changed since I received a teaching endorsement. As of the writing of this book, computer science classes count toward graduation in 32 states and Washington D.C., 11 states have adopted computer science standards, and some are even offering endorsements to teach computer science. The field is growing, but there's still quite a ways to go, particularly given that computing jobs represent the number one source of new wages in the U.S., and half a million of those jobs are still unfilled due to a lack of graduates (Code.org, 2017a).

Integrating computer science into core academic classes serves two primary purposes. First, it provides an opportunity for students to experience computer science in schools where they otherwise wouldn't be afforded the opportunity. Second, it exposes students to the many contexts in which computer science can be applied in the real world.

IST8061 • 9781760568061 • © 2019 Hawker Brownlow Education

Who Is This Book For?

You don't need any prior computer science or coding experience to use this book. My goal is to show you the natural overlaps between CS and core academic areas, with a specific focus on the needs of a middle-school class. In addition to the higher-level connections between computer science and each specific content area, I've selected a handful of classroom activities from the CS Unplugged website (www.csunplugged.org) that can be used to introduce computer science concepts without requiring any technology in the classroom. Finally, each content area section culminates in a coding project that has been tailored for that specific subject. These projects are designed for a variety of experience levels, from a very first experience with only a light amount of programming to a richer project that includes more computer science concepts.

Is It "Coding" or "Computer Science"?

Throughout this book I'll refer to both coding and computer science. While the two terms may seem interchangeable (and are often treated as such), the difference is important.

> **Coding** is the act of writing code for a computer to process.
>
> **Computer science** or **CS** is the study of the core principles of computing, including the processing of information, the design and interplay of hardware and software, and the applications of computing. Coding is frequently the way in which this study is conducted or expressed, but coding is merely one component of computer science.

For the Content-Area Teacher

If you teach language arts, social studies, math, or science, this book is for you. In the appropriate content area section, you'll find support for bringing in core computer science concepts and practices (such as computational thinking) in a way that supports and strengthens your content area instruction. This support includes context about how the principles of computer science apply in your content area in the real world, as well as lesson and project ideas aligned to content area standards, designed to help your students make those connections in the classroom.

If you teach art, music, a foreign language, physical education, underwater basket-weaving, or any other content area, this book is also for you! While the structure is designed around the "core four," all of the ideas presented can be easily modified to fit the context of any number of classes. At the end of each project, you'll find some specific modification ideas for other contexts and content areas.

For the Technology or Computer Science Teacher

For those already teaching computer science or coding in some form, you'll find this book to be a useful foot in the door to build cross-curricular opportunities for your students. While all of these projects could be used without modification in your technology class, I urge you to use the content area contextual supports to work with your peers teaching other courses. Arguments for teaching computer science in each content area and the standards addressed are offered to help you make your case for initiating this collaboration.

By supporting your fellow teachers in their instruction, you'll get a chance to reach students who might not otherwise take your courses. This will also give you the chance to demonstrate the value of your course(s) to the school community at large, hopefully allowing you to build and strengthen your computer science program.

This Is Not a "How-to-Code" Book

There are lots of places to learn the basics of programming in any number of languages—that ground is well trod and supported elsewhere. The focus of this book is finding authentic opportunities for students to apply computer science skills in their core academic classes. While developing your own skills as a programmer is certainly an asset in this endeavor, it's not essential that you learn to code to apply the ideas and lessons in this book. You'll find the projects are quite approachable to new programmers. As long as you can embrace the lead-learner mindset (more on that in Chapter 1), you'll be just fine!

How To Use This Book

This book is broken into three primary sections.

Part I introduces the core pedagogy, strategies, and tools that will be used throughout the rest of the book. Reading this section encourages you to consider the role computer science might play in your regular instruction, and what new instructional strategies you'll want to incorporate as you blaze a trail into becoming a teacher of computer science. The philosophy introduced in this section will help you transform your classroom and your instructional practice, while providing the context for how and why the later activities are designed in the way they are.

Part II is organized into four subsections: language arts, social studies, science, and math. In each of these sections, I make the case for teaching computer science in that content area, with particular attention to the intersection between each content area and the real-world applications of computer science, as well as

IST8061 • 9781760568061 • © 2019 Hawker Brownlow Education

any content-area specific considerations to keep in mind. From there, I explore offline activities from CS Unplugged that can be used as a computer-free introduction to the role computer science plays in each content area. Finally, each section culminates in a rich coding project designed to highlight ways in which computer science and each content area overlap in the real world, aligned to appropriate CSTA, ISTE, Common Core, and NGSS standards.

Part III deals with the more practical aspects of bringing computer science into the classroom. With the projects from Part II in mind, you can explore various approaches to assessment and student support. I offer a handful of approaches to assessing student work, including consideration of ***what*** you'll really want to assess when balancing your content-area standards and the computer science you've integrated. In addition, I'll broach the very real possibility that things won't always work for your students (astonishing!), and how to go about dealing with the inevitable bugs and issues that new programmers are sure to encounter.

Appendixes are provided to help drive your growth as a computer science teacher. Specific resources are provided to get your classroom prepared for the necessary tools, and these include places to learn more about computer science and coding. I've also provided specific examples for how each coding project can be adapted to use different tools or programming languages, as appropriate.

Finally, you'll want to visit the website that accompanies this book: www.creativecodingbook.com. That's where you'll find printable lesson resources, examples for each project, and lots of additional resources to help you on your journey of bringing computer science and coding into your classroom.

Let's dive in!