NO FEAR CODING

Computational Thinking Across the K-5 Curriculum

HEIDI WILLIAMS

Hawker Brownlow
EDUCATION

Contents

Foreword

In his 2009 TED Talk, "How Great Leaders Inspire Action," Simon Sinek introduced a powerful model for inspirational leadership that begins with asking the "Why." He explains that most organizations and companies know what they do and how they do it, but less clear is why they do what they do. Simon discovered a pattern in how great leaders and organizations think, act, and communicate. He coined it the Golden Circle.

Using the Golden Circle, successful organizations begin by exploring the "Why"—the purpose, belief, or reason for doing something. The "Why" informs the processes we take to reach our desired result. Educators can employ this same thinking by asking themselves why they embarked on a career in education. Answers to this question will vary but might include things like, "to help students succeed in their lives and careers" or even, "to make the world a better place."

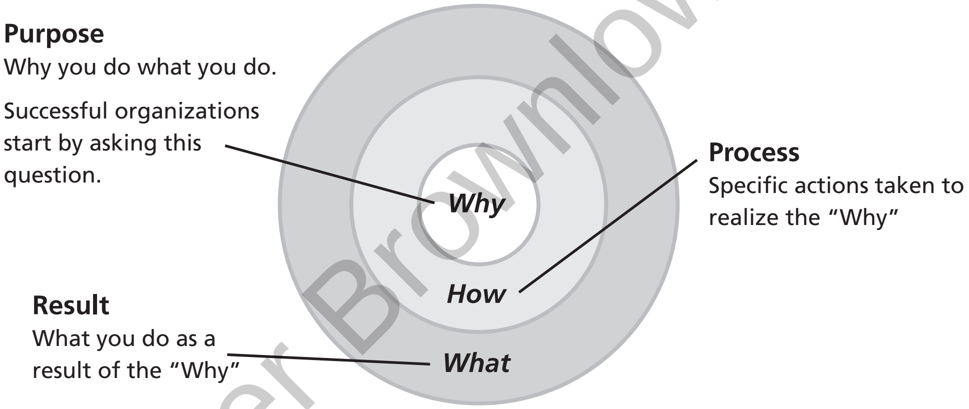

Purpose
Why you do what you do.
Successful organizations start by asking this question.
Why
Process
Specific actions taken to realize the "Why"
How
Result
What you do as a result of the "Why"
What

Figure F.1. Simon Sinek's Golden Circle.

Keeping the "Why" in mind, one can begin to explore from the inside out how to achieve teaching goals and what an effective instructional practice should look like.

No Fear Coding will start with why you should be teaching coding, how you can get it done without adding another thing to your day, and finally what you can do to bring coding and computational thinking skills into your existing curriculum and daily instruction.

For change in education to occur, we need to redefine what it means to be a teacher. The video challenges us to understand that the world has progressed and requires students who can think creatively, innovatively, critically, independently, and with the ability to connect. For their important role in shaping future leaders, the speaker argues that teachers should earn as much as doctors. Just as doctors treat each person as an individual and perform tests to determine what treatment will help improve their patient's health, teachers must help students improve their ability to process and use information. Learning to code and think computationally can help students master vital 21st century skills.

Education Is Moving from Teaching to Learning

Those who went into teaching twenty years ago studied subject matter and teaching methodologies and were often considered, and considered themselves, the holders of knowledge. With the availability of the internet, educators may no longer be the subject matter experts in the classroom. Students are constantly finding and engaging with—via Google searches, YouTube videos, and other methods—content they find interesting and have a passion for. In today's classroom, it is quite possible that a student knows more about a given topic than the teacher does.

Thanks to technology and the connectivity afforded by the internet and mobile devices, students no longer need to sit in a traditional classroom learn the same thing at the same time, all together. Students have access to tools such as Khan Academy, LearnZillion, Dreambox, IXL, and a variety of others to allow them to learn almost any subject they want to know more about. Gone are the days of having students leave school in June and return in September to "pick up right where they left off." Students now leave for the summer only to continue learning, exploring their passions, and gaining a whole new set of skills and background knowledge.

Changing our model of education involves shifting our focus from a teaching pedagogy to a learning pedagogy. In a teaching pedagogy, one asks oneself, "What should I teach?" compared to in a learning pedagogy where one asks, "What should my students be able to do with what they learn?" Both are important to the educational process and can be explained from the vantage point of the Golden Circle. Within this construct, teaching is the "How" (the process or delivery mechanism) and the "What" (the content delivered or knowledge imparted). Motivation becomes a key factor in learning, as this is the "Why" of the educational process.

Here is where coding becomes so important. Coding provides a vehicle for students to create content, rather than just consume it. For example, if a student has a passion for electricity and comes into your electricity unit with prior knowledge that is well beyond her peers, coding becomes a great avenue to allow the student to create with the content, rather than sit through lessons and consume content she already understands.

With the demand on teacher time to differentiate instruction for all learners, coding is a 21st century skill that naturally differentiates for its users. For example, there are at least five different ways to make a character move using coding. While none of these ways are wrong, students will naturally find ways that are more efficient the more they code.

Using This Book

This book will explore why you should be teaching coding and computational thinking, show how you can teach these skills using a variety of tools, and finally demonstrate what you can do now to add these skills into your existing curriculum and teaching.

The book is organized into sections featuring tools that can be used to integrate coding into the curriculum. Each section is laid out using three key concepts:

- The reason WHY using the resource within the curriculum will benefit students
- Description of HOW to get started with the resource
- Case studies and real world application of the resource showing WHAT you can do with it

What makes *No Fear Coding* different than other coding books or products is that it will not ask you to teach coding as a separate subject, but it will intentionally show teachers how to incorporate the skills of computer science within the instruction they already are responsible for covering. This book will challenge you to think computationally as you help students break down problems and create the thought processes needed to solve them.

No Fear!

The most important thing you can do is jump in and begin using the activities in this book. Activities and examples are paired with resources and Try It! challenges to deepen teacher understanding of the concepts explored in each section.

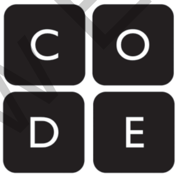
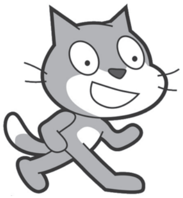
Teachers can use the information in this book to:

- Model risk taking.
- Use Bee-Bots to practice letters and numbers.
- Use Code.org to help young coders explore cardinal directions (up, down, right, left).
- Use Scratch to create a video game demonstrating knowledge on electricity.
- Use Scratch to "play" with the concept of estimation.
- Use ARIS to create a walking tour of your community.
- Incorporate coding activities into teaching the Standards for Mathematical Practice.

Table 0.1 outlines four tools featured in this book that you can use to begin your journey into coding.

The website for this book (**Nofearcoding.org**) contains worksheets and rubrics for lessons within the book, as well as links to resources and projects.

TABLE 0.1. Coding Resources and Where to Find Them

RESOURCE	WHAT IS IT?	WHAT DOES IT LOOK LIKE?	WHERE DO I FIND IT?
Bee-Bots	A Bee-Bot is a programmable floor robot intended for use by young children. Its design and interface appeals to young students and offers a perfect starting point for teaching control, directional language, and programming.		**bee-bot.us**
Code.org	Code.org is a non-profit dedicated to expanding access to computer science and increasing participation by women and underrepresented minorities. Code.org makes learning how to code accessible through lessons, videos, and other resources on its website.		**code.org**
Scratch	Scratch is a visual programming language learners of all ages can use to create interactive stories, games, and animations. Scratch helps young people learn to think creatively, reason systematically, and work collaboratively — essential skills for life in the digital age.		**scratch.mit.edu**
ARIS	ARIS is a user-friendly, open-source platform for creating and engaging with mobile games, tours, and interactive stories. Using GPS and QR codes, ARIS players experience a hybrid world of virtual interactive characters, items, and media placed in physical space.		**arisgames.org**

PART

1

Coding and Computational Thinking

CHAPTER 1

Why Should K–5 Educators Teach Coding?

It's time we realize and put to action Steve Job's words: "Everyone in this country should learn how to program a computer…because it teaches you how to think." This quote appears at the beginning of a video entitled "What Most School Don't Teach" (**youtu.be/nKIu9yen5nc**) that was published in 2013 by Code.org. Starring many famous people, including Bill Gates of Microsoft, Mark Zuckerberg of Facebook, and will.i.am of the Black-Eyed Peas, the video's message is a simple one: anyone and everyone should learn to code.

Within our schools, we must build upon the belief that coding is for everyone. There are many reasons for teaching coding, among them:

- It's about teaching perseverance.
- It's about teaching students how to think and reason (computational thinking).
- It's about creativity and expression.
- It's another way to demonstrate content knowledge (just like creating a PowerPoint or display board).
- It's a way to see math in action.