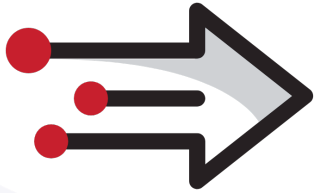
Extending the CS Pipeline



Code in Every Class

How All Educators Can Teach Programming

by Kevin Brookhouser & Ria Megnin

Book Club Facilitators' Guide

Education and Innovation Research (EIR)
United States Department of Education
Award #S411C200084

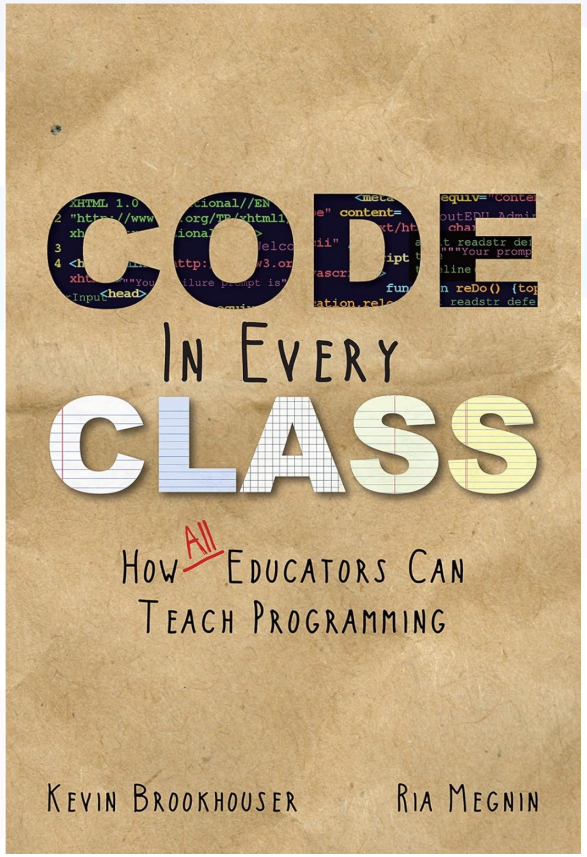
RUTGERS–NEW BRUNSWICK
Center for Effective School Practices
Graduate School of Education

Introduction

This resource is intended to be used by educators, professional learning providers, or computer science (CS) education advocates to provide educators with a meaningful and engaging medium to learn about CS education that only requires minimal preparation.

Included in this tool are the essential elements required to host this three-meeting book club. For each meeting, a conversation starter, discussion questions, and a wrap-up exercise are all provided. However, facilitators are encouraged to modify, add, remove, and further customize the club to fit the interests and goals of their groups.

The Book:
Code in Every Class



Summary

It's an accepted fact that all students need exposure to certain academic disciplines—algebra, world history, chemistry, and probably a Shakespeare play or two. Unfortunately, computer science has not yet risen to must-have status. Given society's reliance on technology in almost every aspect of modern life, author-educators Kevin Brookhouser and Ria Megnin find the lack of programming education in schools shocking. Code in Every Class explains why computer science is critical to your students' future success and shows you how to take tech education into your own hands. And here's the best news: You don't have to be a computer genius to teach the basics of coding. You'll learn…How to incorporate fundamental programming skills into your existing lessons—even if you don't have computer coding experience. How to teach coding principles without a computer. Why computer literacy levels the playing field for underprivileged students. Too many students have never been shown how to take control of computers and other digital devices. They're stuck as consumers, when they should be learning how to use tech tools to be creators and problem-solvers. Code in Every Class will allow you to open a world of opportunities to your students, one lesson at a time.

Purchase Link (Amazon): Code in Every Class

Also included with Kindle Unlimited subscription for no additional cost.

Book Club Schedule

Meeting	Requisite Reading	Discussion Topics
#1	Chapter 1: The Case for Coding Chapter 2: The Demand for Coding Chapter 3: Coding as Social Justice	What is computer science, who should learn it and why, relationship between CS and technology, CS as a change driver, CS to empower students
#2 (~14 days later)	Chapter 4: All Educators Can Teach Coding Chapter 5: Sparking Student Interest Chapter 6: Earning Administrators' Buy-in Chapter 7: Building Your Expertise	What it means to teach programming, importance of connecting CS to other topics, core concepts of programming, institutional support for CS education
#3 (~14 days later)	Chapter 8: How to Launch a Coding Program Chapter 9: Coding Lessons Chapter 10: A Call to Action	Core concepts of programming (continued), how to bring CS and programming to your own practice, committing to goals and change

Promotional Materials

Printable bookmarks are available to distribute to interested educators! Access, modify, and download the bookmarks using the link below. Print services are also available for a fee (purchased directly through Canva).

[Bookmark Template Link]

Additional Resources to Explore

In addition to discussing the book and related topics, the below activities may prove to be engaging and thought-provoking additions to your book club's activities.

1. Rutgers EIR Resource Library
2. CS Unplugged Lessons
3. 5 Ways to Get Your Students Coding Across the Curriculum
4. 15+ Ways of Teaching Every Student to Code (Even Without a Computer)
5. Resources for Your Coding Program (inspired by this book)

Meeting #1

Chapter 1 – Chapter 3

Conversation Starter

Begin the session by posing the question, “What is Computer Science?” Allow each participant to share their initial thoughts without interruption. This open-ended question encourages diverse perspectives and helps gauge the group's pre-existing understanding and biases.

As the facilitator, guide the conversation to explore the multifaceted nature of computer science. Here are some key points to bring up:

1. Interdisciplinary Connections: Highlight how computer science integrates with other disciplines such as mathematics, science, art, and even humanities. Discuss examples like algorithmic art, computational biology, and digital humanities.
2. Beyond Coding: Emphasize that while programming is a significant aspect, computer science also includes theory, hardware development, system design, and problem-solving methodologies.
3. Impact on Society: Explore the role of computer science in societal change, including topics like ethical AI, data privacy, and the digital divide.

Discussion Questions

- The authors spend the first few chapters covering many of the “what” and “why” of Coding in Every Class, or across the curriculum. In your opinion, what is the most compelling reason?
- This book was published in 2016. In your experience, do you feel the content is dated? If so, how? If not, why do you think it is still relevant? (why have there not been more advances in integrating CS education? What messages make it lasting?)
- Chapter 3 discusses Coding as Social Justice, and outlines steps that can be taken. Has your school, or society in general, made any advances in these steps? What additional steps can be taken? How can diversity in CS education be improved?
- Why is diversity in CS and technology so important?
- What do you think is the current stereotype for a "techie" person? Does this need to change? How can we do that?
- Exposure to CS can be a boost for a student who otherwise struggles academically or socially (Ch 1 “A Chance to Shine”). How do we help the student who struggles with understanding technology? Is there an opposing stereotype to the “techie”?

Wrap-Up & Reflection

In this reflection activity, participants will consider computer science as an interdisciplinary and inclusive subject area. Using a tool like Picker Wheel or printed index cards, select a subject area and a CS domain at random. Then, discuss potential applications or connections of the CS domain to the subject area.

Subject Areas: English & Literacy, Visual & Performing Arts, Mathematics, Social Studies & History, Finance & Business

CS Domains: Computing Systems, Networks and the Internet, Impacts of Computing, Data & Analysis, Algorithms & Programming

Meeting #2

Chapter 4 – Chapter 7

Conversation Starter

Begin the session with the question, "What do you need to teach programming effectively?" Encourage each participant to list items or conditions they believe are essential. This could include tangible resources like software and hardware, as well as intangible needs like pedagogical strategies and institutional support.

As the discussion unfolds, guide the conversation to cover several crucial areas:

1. Resources: What kinds of tools and materials (like IDEs, computers, textbooks) do participants find indispensable for teaching programming?
2. Skills and Knowledge: Beyond knowing programming languages, what other skills do educators need? Consider aspects like understanding learning theories, ability to encourage critical thinking, and techniques for engaging students.
3. Interdisciplinary Connections: How important is it to link programming with other subjects? Discuss ways to integrate CS with math, science, art, and more to enrich students' learning experiences.
4. Institutional Support: What type of support from schools and districts enhances the teaching of programming? Explore needs like professional development, curriculum freedom, and administrative backing.

Discussion Questions

- The book asserts that “all educators can teach coding.” Do you agree with this sentiment?
- How do you believe that math and computer science became so closely associated? Do you believe that strong math skills are needed for success in CS?
- Chapter 6 talks about earning administrator’s buy-in. What initiatives have you experienced that have or have not had this institutional support? What role do you think it will play in bringing CS education to your school?
- There can be hurdles to adding CS to the curriculum. Is the cost still one of them/at your school? What other hurdles are there at your school?
- As librarians, you know the value of finding and using information. In our world that is largely done with technology. What technology skills do you see your students possess? What technology skills or knowledge are they lacking?
- What topics discussed in Chapter 7 (build your expertise) resonated most with you? Do any of them seem like they can be incorporated into your instruction?

Wrap-Up & Reflection

As a group, review the video and lesson plan for the Parity Magic demonstration, which builds on the concept of binary numbers, bits, and bytes discussed in Chapter 7 (build your expertise). What did you learn from this demonstration? Is this something you can replicate in your classes? The most important section is the debrief discussion, where the concepts from the demonstration are connected to previous concepts and further explained.

Meeting #3

Chapter 8 – Chapter 10

Conversation Starter

- Each participant should select a resource shared in Chapter 9 and spend 15 minutes exploring it. To the extent possible, resources from each area should be selected before doubling up. Afterwards, each participant should share out on their experience, what topics were covered and how, and what they believe students would think of the activity.
- Participants should be encouraged to debrief on the experience and think about other resources they could use, how the resource they explored would be incorporated into a larger, more connected lesson or unit, and how feasible implementing it would be.

Discussion Questions

- What programming concepts did you struggle with understanding? What can we review together as a group so we can all have clarity?
- How comfortable are you with explaining these core concepts? Can you identify an activity or lesson that supports them? Can you identify an appropriate area in your curriculum to incorporate any?
- What is a project or goal you would like to implement next year?
- How are you going to build your skill set to support teachers and students? What resources or support do you need, and where can you get them?
- Do you feel inspired to incorporate any programming or CS concepts in your class? Why?
- This book self describes as a book about "keeping technology, and specifically code writing (aka programming), as simple as possible." Do you feel the book met this goal? Why or why not?
- Did this book change your perception of computer science education? If so, how?
- Now that we have completed the book, what were our overall thoughts?

Wrap-Up & Reflection

Develop and commit to a SMART goal around implementing a programming or CS topic in your instruction. Identify what resources you will need to make your goal happen (other people, physical and digital resources, content or pedagogical knowledge) and think through a timeline for the planning and implementation of your goal. Participants should review their goal in depth with at least one other person to get meaningful feedback.