A Practitioner's Guide to Using CS Curricular Resources

For middle school computer science teachers and the administrators who support them

DAVID J. AMIEL, CYNTHIA L. BLITZ, & FRAN P. TREES

Walk into almost any conversation about CS curriculum and you'll hear the same names: Code.org, CodeHS, Tynker, CMU CS Academy, Project STEM, CS Unplugged, Google CS First, PLTW, and that's before you get to individual lessons on Teachers Pay Teachers, YouTube, or the platform your district just licensed. The problem isn't that there's nothing to use. It's the opposite. With dozens of credible options available, deciding what to use can feel paralyzing, and the pressure of a ticking clock too often leads to arbitrary choices: the resource a colleague happened to mention, or the first hit on a search, rather than the one that is the best fit your students, your classroom, and your goals.

Here, we offer a framework for more making more intentional decisions that treats curricular decision-making as an ongoing professional practice and that teachers rarely use a resource exactly as-is.

Planning Your Search

Before you start searching, identify what you actually need. This is the single most important, and most often skipped, step. It helps to think broadly about what counts as a "curricular resource." Full-course curricula get most of the attention, but resources also include individual lesson plans, worksheets, activities, slide decks, videos, projects, guest speaker connections, analogies, and discussion prompts.

- → **Replace?** Are you swapping out a lesson, unit, or entire course with something better aligned?
- → **Add?** Are you introducing a new topic, project, or lesson not currently in your curriculum?
- → **Supplement?** Are you looking for extensions, alternative explanations, or additional context to enhance what you already have?

Think *broadly* about what a resource can be, and *narrowly* about what you need. What exactly are you looking for? What would the perfect find look like? How are you going to use what you find?

Finding Resources

With a clear sense of what you need, you can search more efficiently. CS curricular resources live in a wide range of places and knowing where to look (and how to look) saves time and surfaces better options.

> **Tip!** You can decouple a good activity or lesson from its parent curriculum. If one lesson from a larger unit works for your class but the rest does not, take that lesson. You are not obligated to adopt an entire curriculum to use one piece of it.

Start with established platforms. Organizations like Code.org, CSTA, CS Unplugged, and Google CS First maintain curated libraries of lessons, units, and full courses. These are often well-documented, standards-aligned, and free. If you are looking for a full curriculum or a structured unit, these are strong starting points.

Funded by the U.S. Department of Education, Education Innovation & Research (EIR) Award #S411C200084

Suggested Citation

Amiel, D. J., Blitz, C. L., & Trees, F. P. (2026). *A Practitioner's Guide to Using CS Curricular Resources* [Practitioner Brief]. Rutgers University Center for Effective School Practices. cesp.rutgers.edu

Search with specificity. A general search for "middle school CS lesson" returns thousands of results. A search for "CS Unplugged sorting activity middle school" or "block-based intro to loops lesson plan" returns something you can actually evaluate. The clearer your need (from the previous step), the more targeted your search can be.

Look beyond full curricula. Teachers Pay Teachers, YouTube, GitHub repositories, university outreach programs, and conference presentation archives all contain individual lessons, activities, and projects that may be exactly what you need. A single well-designed activity can be more useful than an entire curriculum that only partially fits.

Ask your network. Colleagues in your school or district, your CSTA chapter, state CS education communities, and online teacher groups are among the most reliable sources for recommendations. A colleague who teaches in a similar context can tell you not just what they used, but how it worked in practice.

Evaluate as you go. Not every result deserves a deep look. As you browse, you can quickly screen resources against your need: Does this address the right topic? Is it designed for roughly the right grade level? Is it obviously incompatible with your schedule or technology? A quick scan saves you from investing time in a detailed evaluation of something that was never going to work.

Selecting Resources: Three Lenses to Consider

Once you have identified a few promising options, three lenses can help you evaluate them more carefully. These are not a rigid scoring rubric. They are a thinking tool, and the weight of each lens shifts depending on the situation.

Logistic Factors (Feasibility)

Can you actually use this?

- Materials and equipment required
- Device compatibility (Chromebooks? iPads? Specific software?)
- Lesson and class duration (Will a 90-minute lesson work in your 42-minute period?)
- Prep time required of the teacher
- Included assessments, and whether they are useful
- Cost (free, freemium, subscription?)
- Language accessibility for multilingual learners

Content Factors (Relevance)

Does it teach what you need it to teach?

- Difficulty level appropriate for your students
- Your own familiarity with the material, or the learning curve if the content is new to you
- Alignment with state standards, CSTA standards, and local curriculum requirements
- Whether it builds lasting conceptual understanding or functions as a one-off activity

Context Factors (Fit)

Does it fit your students and classroom?

- Connections to what students have already learned and what they will learn next
- Examples and analogies: are they relevant to students' lives and culturally responsive?
- Instructional methods that align with your teaching (instruction, inquiry-based, project-based)
- Activities compatible with your class size, classroom setup, and student population

Adapting Resources: Make Them Yours

One of the most important messages in this brief: you almost never use a curricular resource exactly as-is. Adaptation is not a failure of the resource or a sign you have chosen poorly. It is normal, expected, and necessary. Three kinds of adaptation are worth naming:

Adjust the structure.
- → Combine multiple resources into one lesson or unit.
- → Reorder concepts or activities.
- → Stretch or compress a lesson to the right number of class periods.

Adjust the content.
- → Swap examples for ones more relevant to your students.
- → Add connections to previously covered material or previews of concepts coming next.
- → Modify the depth of coverage based on student readiness.

Adjust the pedagogy.
- → Add a teacher demonstration or modeling step.
- → Build in scaffolding such as partial solutions, hints, or worked examples.
- → Change how an activity runs in the classroom

To make modifications to the pedagogy of a lesson, you could make it collaborative, add pair programming, incorporate think-pair-share, infuse cooperative learning structures, or close with a debrief discussion or student reflection. And remember, it's Iterative. The first version of an adapted lesson probably will not be perfect, and that is expected. Working with curricular resources is a cycle, not a one-time decision.

> **In Practice – Selecting (some of) Code.org**
>
> One middle school teacher faced a tight constraint: CS every other day, one marking period, 37 minutes per class, roughly 900 minutes of instructional time total. Code.org's AI & ML unit looked promising but ran about 1,150 minutes as written. Rather than cut the unit entirely or rush through it, she selected just two chapters, modified most of those lessons, extended some activities, and removed others. The result was a version of Code.org's content that actually fit her classroom.

Share What You Learn

When you find, adapt, or build something that works, share it. Your CSTA chapter, state CS teacher communities, social media, district colleagues, conferences, and resource libraries are all valuable channels. Sharing is a great way to contribute to the field get feedback on your own practice!

> **In Practice – Assembling a Routing Lesson**
>
> An EIR partner educator wanted a hands-on way to teach routing and deadlock, networking concepts that can be hard to make concrete. Rather than hunting for one perfect resource, she combined three: The Orange Game from CS Unplugged, a Mail Game from Teachers Pay Teachers, and Muddy City from CS Unplugged. She then added her own connective tissue: debrief discussions, explicit connections to the underlying CS concepts, and cooperative learning structures.