CS4NorCal

Computer Science Teacher Brief

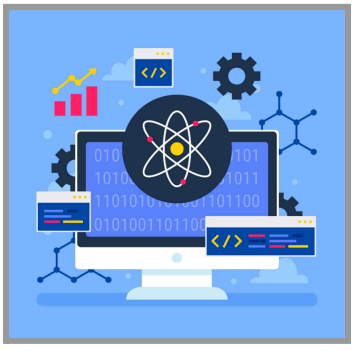
No-Stress CS: Discover, Play, and Learn

PROBLEM OF PRACTICE

Photo by: Preslee, student Scott Valley Jr. High School

In small and rural schools, internet access isn't always reliable, devices don't always play nice, updates and licensing can create unexpected headaches, and teachers often have limited access to training or tech-savvy colleagues who can lend support. *Sound familiar?*

Embedding computer science (CS) education into middle school curricula is essential for preparing students to innovate and solve problems in a technology-driven world that is increasingly shaped by computer technologies *(Huang & Looi, 2021)* but does not require educators to learn how to teach coding or navigate software-based curriculum and platforms. This short brief introduces **engaging ways that middle school educators can easily experiment with integrating foundational CS concepts into their existing lessons, across content areas,** to foster STEM-CS learning through student-centered instructional strategies.



What is Computer Science?

Computer science is a discipline: the study of computers, their applications, and their impact on society – with a focus on understanding why they work and how to create those technologies *(CCSIP, 2019)*. Computational thinking (CT) is a way to think about solving problems and draws on fundamental concepts of computer science *(Wing, 2006)* and essential 21st century skills that young people need to be successful in a digital world. For example, decomposition, pattern recognition, abstraction, algorithmic design, and evaluation are core CT competencies that are grounded in systemic approaches to problem analysis that cultivate student's problem-solving thinking skills and processes.

"One of the foundational beliefs in my classroom is that learning something new is inherently vulnerable. The brain is hardwired to assess risk – anything new feels dangerous!

By switching to a "learner mindset," students have the freedom and opportunity to experience vulnerability in the classroom – to learn through trial and error, communication, and collaboration – all within a low-stakes, safe environment."

– Hannah Shickle, Teacher
Siskiyou County Teacher of the Year 2025

Why Computer Science Education Matters

Middle school years are a challenging time of major physical and psychological changes among adolescents – where they are reaching conclusions about their aptitudes and skill development. These are also critical years for identity formation and cognitive development for the analytical reasoning required of STEM disciplines *(Reardon, 2022; Grover, Pea, & Cooper, 2015)*. A growing body of research supports the idea that CT/CS concepts and practices can be effectively integrated across content areas (such as: English language arts, social studies, mathematics, science, and the arts) to foster students' problem-solving capabilities and higher-order learning *(de Jong et al., 2023; Settle et al., 2012*), as well as critical thinking, communication, and collaboration skills. Computer science education affords educators with opportunities to develop a number of these important 21st Century skills that lay the foundation for students' further learning, adaptability and productivity in their lives and careers *(Partnership for 21st Century Learning, 2019)*.

Ms. Shickle's CS background: Hannah is a highly-motivated self-learner of educational technologies and was introduced to the Computer Science Discoveries (CSD) curriculum through CS4NorCal's professional learning summer institute in 2021.

Setting and Context

IN THE SPOTLIGHT:
Hannah Shickle, Teacher of English/Social Sciences, Grades 7 & 8, Computer Applications, CS enthusiast

SISKIYOU COUNTY

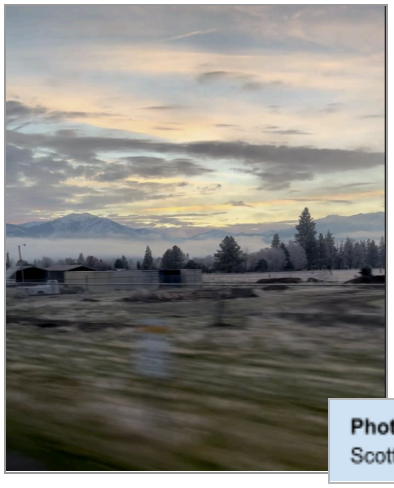


Photo by: Preslee, student
Scott Valley Jr. High School

Challenges

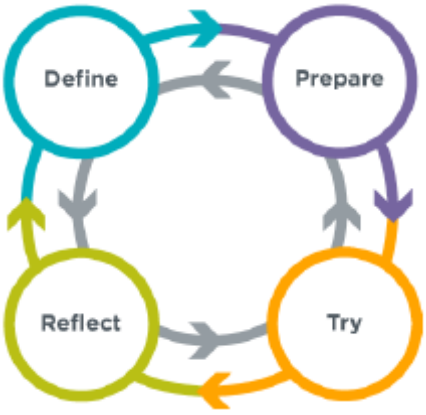
"The community I serve is located in the northernmost part of California. It is a rural agricultural community with varying degrees of access to computer science education. The primary problems with being a full-time rural classroom educator and computer science are ***time, resources and training****. As a teacher, 90% of your battle is how to keep students engaged and give them something a little bit different."*

PROMISING APPROACHES

Hannah intentionally **integrates computer science concepts into her regular curriculum**, layered within core content to serve a dual purpose. This approach not only deepens students' application of CS skills but also makes core lessons more engaging and dynamic. For instance, problem-solving, technology integration, data, input/output, and research skills can all be embedded into content learning when taught through this lens. As Hannah explains, "*I think CS aligns great with science and history because there's more opportunity for hands-on learning — especially in history, where you can pull students out of the text, get them into something active, and make a lesson come alive*

The Problem-Solving Process as an Entry Point

One of the foundational concepts in computer science is problem-solving — a process for tackling unfamiliar challenges. *"I don't think all students have a conscious understanding of problem-solving as a way to build resilience,"* Hannah notes. *"I love using the problem-solving model from* Code.org's CSD curriculum *to guide how we define the problem, make a plan, test it, evaluate, and then try again."*



The Problem-Solving Model

She explains that students don't always know what to ask – or even how to ask it. The problem-solving process helps them generate better questions. Through trial and error, they arrive at different conclusions and refine their thinking. *"That concept of asking questions – getting them curious – is critical to student success long-term,"* Hannah emphasizes.

Starting with the problem-solving process in a low-stakes environment that encourages productive struggle without the pressure of academic evaluation. Hannah has found many applications for this simple, yet foundational concept, such as the low-stakes Tangram activity which can be delivered with increasing complexity. She also offers a high-stakes approach using a U.S. History project; which incorporates CS and Universal Design for Learning (UDL) principles where students can be evaluated on content knowledge, standards and outcomes. You can find a short recap of both these activities below:

- TANGRAM PROBLEM-SOLVING ACTIVITY (LOW-STAKES)
- COLUMBIAN EXCHANGE: DINNER FOR TWO! (HIGH-STAKES)

Tangram Problem-Solving Activity (Low-Stakes)

Tangrams offer a simple way to engage students in algorithmic problem-solving, teamwork, communication, and development of perseverance. Solving tangram puzzles involves manipulating shapes to create images. Engaging students in the process of creating and reconstructing forms is an essential skill in geometric and spatial thinking.

"I have used this activity at the beginning of every year with my 7th-grade students to set expectations for a collaborative class culture and reinforce these baseline learning behaviors: learning is interactive, productive struggle is healthy, trust matters, and collaboration is powerful."

Duration: 20-45 minutes
Supplies: Class set of tangrams, tangram images (color & silhouette)

In this low-stakes activity, each student receives a bag of **pre-sorted** tangrams. To start, I introduce the concept by asking questions like, *"Have you ever heard of tangrams? Played with them? What's your experience?"*

Students share a range of responses, which we reinforce through discussion. I make sure to confirm their ideas by repeating back what I understood, and students build on each other's answers in conversation.

Next, I show examples of images created with tangrams and model a **color-blocking strategy** so students can see how the pieces fit together. For instance, I might display a picture of a cat built from tangrams. The room quickly fills with energy as students try to recreate it, testing and comparing their shapes.

Once I know they understand how to manipulate the tiles and break down an image into parts, I present this challenge: I display a **blacked-out silhouette** made from tangrams. This shift introduces **productive struggle**.

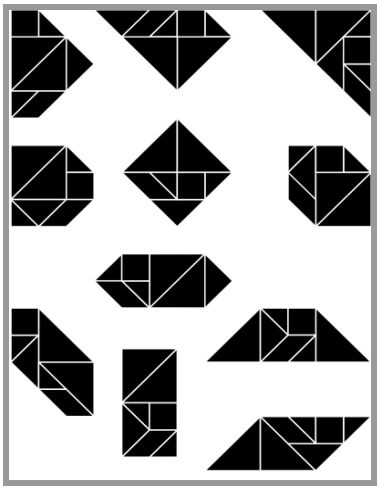
Since most students are familiar with tangrams, I add two rules:

1. **No talking to each other** – this encourages them to think critically and try independently
2. **Five-minute time limit** – this gives them confidence that the struggle won't last forever

I explain: *"This is productive struggle. You'll have five minutes to try, and then you'll be able to ask any questions you have. I'll answer your questions, but I won't show you the answer yet."*

TIP: *To increase complexity, pair students by roles as "eyes" or "hands"*
Only the "eyes" can see the image and give instructions to the "hands"
on how to create the image

** **WARNING** – this can be tense. I advise setting a time limit (3-5 minutes) AND actively walking around the room, engaging with groups!*



During this phase, I take two approaches:

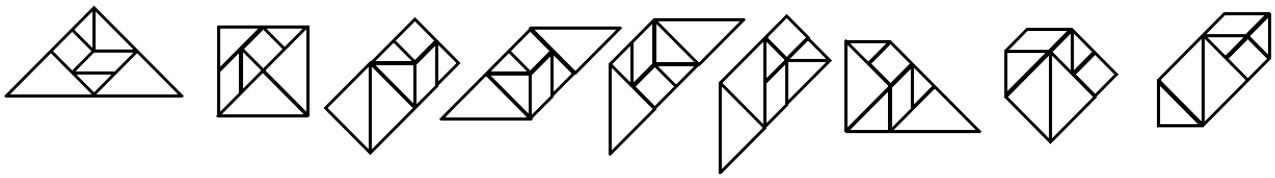
1) Sometimes I sit at my desk with my own tangrams, displaying my attempts on the document camera while quietly working alongside them. I let students see me demonstrate real frustration or mistakes to model that even teachers wrestle with the challenges.

2) Other times, I walk around the room giving encouragement – thumbs up, nudges of support – while students share progress with their peers.

After five minutes, we move into a **question inquiry phase**. Students share what they tried, where they got stuck, and what they noticed. Common challenges – like confusing the small and large triangles – become opportunities to normalize struggle and refine questions.

I remind them, *"Sometimes it's not the answer that's off, it's the way we asked the question. Learning to ask better questions is just as important as solving the puzzle."*

This activity builds problem-solving, resilience, and also a classroom culture of inquiry. Students begin to see struggle as part of the learning process, which is a disposition of computer science education and prepares them for deeper critical thinking.



Columbian Exchange – Dinner for Two! (High–Stakes)

This high-stakes classroom assignment requires rigorous preparation to demonstrate a comprehensive understanding of a world event – both in its historical and contemporary contexts. It incorporates CS concepts such as input and output models (via computer program integration) and emphasizes problem-solving through peer-to-peer and self-to-world communication.

Grade Level: 8th Grade U.S. History
Duration: 7-10 (45-minute class periods)
Supplies: Computers or Chromebooks

For example, after understanding the task, considering what information is needed, they must identify the inputs and outputs (e.g., information, prototypes, feedback from peers) needed to meet the task requirement.

Frontloading Content

Before this project, students will have studied the Columbian Exchange through multiple class lectures, videos, discussions, mini-interactive activities, and class or independent readings. These lessons will give them sufficient background knowledge on the Columbian Exchange, its purpose, and its historical significance (e.g., the continents of the Old World and the New World were in geographic isolation from one another).

The UDL aspect of this project allows students to exercise choice and creativity in the design of their project to showcase their comprehensive understanding about these lessons.

Activity 1: Food Research

Students will research foods that existed only in the New World or only in the Old World before the Columbian Exchange. Using provided resources (or independent research), they will compile a T-chart list to serve as the design foundation for their project. Students should create a **1–1.5 page, T-Chart list** of potential foods available before the Columbian Exchange.

Activity 2: Studying Menus

Students will review what makes an effective menu. The teacher will model and discuss examples, highlighting criteria such as:

> Food item choices and descriptions: *Do the dishes sound appetizing?*

> Word choice and captions: *Are they engaging and clear?*

> Layout and design: *Is the menu easy to read?*
Are fonts legible, appropriately sized?

Activity 3: Creating the Menus

Students will design **two menus, using a medium of their choice** – one for the New World and one for the Old World. Each menu must only include items authentic to that side of the world *before* the Columbian Exchange. See suggested guidelines for evaluation criteria.

Actionable Advice

Student-centered instructional strategies are key to Hannah's teaching and lesson planning to stimulate student engagement and promote discovery… where students can take an active role in their learning. The problem-solving process, critical thinking, team-building, and communication learning modalities are key take-aways from CS professional learning that Hannah considers as baseline criteria for integration. Middle school students can vary widely in prior CS exposure, so a UDL approach is a key strategy for student success:

- **Classroom Foundation**: Create a classroom culture where student-led instruction is not only encouraged… but valued. Position students as lead learners and mentors – this allows curiosity to thrive and leads to deeper learning.

- **Spotlight Student-led Instruction**: Hannah always communicates that students bring valuable ideas to our classroom. She likes to say: *"When I learn something from my students, I do a big class pause and shout out the learning. It reinforces the concept that we're all learners, and everyone in the room is equal in that way."*

- **Get Curious, Ask Questions – The QFT**: Turn learning into a collaborative investigation. Get students asking questions before an activity begins to spark thinking, lower barriers, and increase engagement. The Question Formulation Technique (QFT) is a powerful, inclusive strategy that helps students generate their own questions about core content or activities and promotes creativity, problem-solving, and reflective thinking.

Hannah's Steps Modelling the Question Formulation Technique (QFT):

- I start with a focus topic (e.g., World War II, Cells, Science Fiction) and ask students: *"What do you know?"* I compile their responses on the board.
- Then I ask, *"What questions do you have?"* Students get 2 – 4 minutes to write down as many questions as possible – without judging or answering them.
- Next, they repeat the process in small groups using large chart paper. Each student uses a different colored pen, and together they compile their questions.
- We regroup as a class to share, then review the difference between open and closed questions.
- As a group, they label them (O for open, C for closed) and star the four most interesting "must-know" questions. Throughout the lesson, I display the posters with their questions.
- At the end, we revisit them: "*Which did we answer? Which remain unanswered?*"

THINGS TO CONSIDER

Teaching CS with Heart: Hannah's Reflection on Orientation & Motivation

Embark on the CS and curriculum integration journey with the mindset of a learner

- One thing I value tremendously as an educator is being a learner myself and showing students what it looks like to learn. Learning is often messy and difficult, and that's okay. I encourage both teachers and students to lean into that mess, because within it lie unexpected opportunities for problem-solving and growth.
- When integrating computer science content, I remain open and willing to let my students teach me. By doing this, I have gained more knowledge and insight into computer science than I could have on my own.
- Create an environment for productive struggle. Build trust with and between students – value it, celebrate it, and model it daily in your class culture. It takes time. Actively participate alongside students.
- As educators, we need to embrace the challenges and unpredictability that come with teaching. The "mess" creates moments we could never plan for, but those moments are often where the deepest learning happens.

Plan for dual standards to maximize learning

- Start with the outcomes in mind. What standards are you trying to achieve within the curriculum? and which CS standards, concepts, or practices can help you get there?
- I ask: *What CS standard do I want my students to master or demonstrate proficiency in, and what is the most engaging content I can pair with that standard to spark buy-in and enthusiasm from students?*

REFLECTIONS

When Creativity Clicks: Turning Skills Into Possibility

I was teaching spreadsheet art to my 7th-grade computer applications class using Google Sheets. It's part of Google's applied skills curriculum where students learn to transform data visually into a spreadsheet. What makes it so engaging is that students practice conditional formatting by creating art.

One student created the face of the Joker – and it was phenomenal. I was blown away by both his creativity and his technical skill. I tell my students, *"If you make rad art, I'll print it." When he saw it on paper, he asked, "You printed that?" I said, "Heck yeah, that was amazing!"* I made sure to tell him that what he produced wasn't just "spreadsheet art"; it was the foundation of graphic design. When I asked if he knew people get paid to do this kind of work, he looked surprised – it had never even crossed his mind.

That moment stuck with me. It reminded me how powerful it is to show students the real-world value of their talents, especially in rural areas where access and exposure to creative career opportunities can be limited. It also showed me how teaching new content can create unintended but life-changing consequences – ones we may never anticipate when we design a lesson.

STUDENT POTENTIAL IS ENDLESS...

AND YOU ARE THE FACILITATOR OF THEIR CREATIVITY!

REFERENCES

de Jong, T., Lazonder, A. W., Chinn, C. A., Fischer, F., Gobert, J., Hmelo-Silver, C. E., ... & Zacharia, Z. C. (2023). Let's talk evidence – The case for combining inquiry-based and direct instruction. Educational Research Review, 39, 100536.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. Computer science education, 25(2), 199-237.

Partnership for 21st Century Learning. (2019). Framework for 21st century learning. https://www.marietta.edu/sites/default/files/documents/21st_century_skills_standards_book_2.pdf

Reardon, R., (2022) Integrating Computational Thinking in Rural Middle School Art Classes in Eastern North Carolina. Theory & Practice in Rural Education, Vol. 12, No. 2, 71–87

Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., & Wildeman, B. (2012, July). Infusing computational thinking into the middle-and high-school curriculum. In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (pp. 22-27).

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33 – 35.

Authored by: *Hannah Shickle, Teacher, Siskiyou County, and*
Robin Martin, Director of Educational Research and Evaluation, University of California at Davis

Computer Science in Rural CA (CS4NorCal) is a project funded by a U.S. Department of Education grant and sponsored by the Small School Districts' Association.