Computational Thinking: Why It Matters

Think you need to have a computer or advanced computer skills to help students learn the problem-solving processes that computer science experts use? Think again. "Computational thinking" doesn't require any specialized equipment or abilities, but it does cultivate key skills that are useful for school, work and everyday life. Students who learn computational thinking can apply these processes:

- State a problem in a way that enables you to use a computer and other tools to solve it.
- Logically organize and analyze data.
- Use models or simulations to represent data.
- Develop an automated solution for a problem by breaking it into a sequence of smaller steps.
- Identify and try various solutions to determine what works best and why.
- Use the fine-tuned processes described above to solve a variety of problems.

Computational thinking skills lay a foundation for many career paths, including ones that don't involve computers, so it is especially important to build these skills among students from groups that tend to be underrepresented in science, technology, engineering and mathematics (STEM) fields. These groups include girls and young women, socioeconomically disadvantaged students, English learners and students with disabilities.

This guide was designed to help you get started sharing activities that cultivate computational thinking. It identifies some potential challenges a 21st CCLC practitioner might encounter when teaching computational thinking, and offers suggestions for how to address them.

Problem	Example	Solution
The student has trouble understanding what the problem is, so he or she can't clearly state the problem.	The student is asked to describe a process for fixing a broken bike. However, the student has never owned or ridden a bike, and has no context for the problem.	**Sort Information (Abstraction)** Discuss the fact that you need to consider only the parts of a problem that are important to solving it. Ignore information that is irrelevant to the solution. To fix a bike, you don't need to know how to ride a bike; you do need to know what might cause a bike to break. Explain some of the common ways bikes can break — the brake cables can become loose, the chain can come off the gears, and so on.
The student has difficulty making an accurate mental model of a complicated system, and such a model might be needed to solve the problem.	The student understands the various ways a bike might break, but has trouble picturing how the pedals, gears, wheels, handlebars and brakes work together to make the bike operate properly.	**See the Process (Decomposition)** Help the student "decompose" the problem by breaking it down to its basic parts. Highlight how the pedals on a bike work to turn the gears, and the gears turn the wheels, which can be steered with the handlebars and slowed or stopped with the brakes. Decomposition can give the student a picture of how each part of a system works by itself, and how all the parts work together.

Computational Thinking: Why It Matters

Problem	Example	Solution
The student struggles to put together a complete process for solving the problem.	The student has a good picture of how all the parts of a bike work, but still isn't sure how to use that knowledge to put together the steps needed to fix a broken bike.	**Map the Steps (Algorithms)** Show the student how to structure the steps of a solution in a logical and efficient process, called an algorithm. If you need the same tools to fix the brakes that you need to fix the gears, for example, you could be efficient by putting these steps next to each other.
The student becomes frustrated when a proposed solution doesn't work.	The student has come up with a process for fixing the bike that considers all parts of the problem individually and as a part of the system, but the solution still isn't working quite right.	**Phone a Friend (Collaboration)** Have students help one another using the Collaborative Discussion Framework to ask these questions: 1. **What are you trying to do?** (Ensure that they understand the goal of the exercise.) 2. **What have you tried already?** (Have them repeat each step they have taken so far.) 3. **What else do you think you can try?** (Brainstorm for alternative solutions.) 4. **What would happen if…?** (Develop hypotheses about alternative solutions and test them to see if they work.)
The student has trouble applying the lessons learned from one problem to a different but similar problem.	Having successfully designed a process for fixing a bike, the student is now having trouble designing a process for fixing a scooter.	**Look for Patterns** Highlight the importance of recognizing patterns. A bike and a scooter aren't quite the same, but they are similar. The wheels on a scooter might break the same way the wheels on a bike might break, and they might be repaired in the same way. By recognizing this pattern and reusing a solution you've already developed, you work efficiently and make your life easier.

Computational Thinking: Why It Matters

Bibliography

International Society for Technology in Education and the Computer Science Teachers Association, *Operational Definition of Computational Thinking for K-12 Education* [Based on work supported by the National Science Foundation]. https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf

Maya Israel, Quentin M. Wherfel, Jamie Pearson, Saadeddine Shehab, and Tanya Tapia. "Empowering K-12 Students With Disabilities to Learn Computational Thinking and Computer Programming," *Teaching Exceptional Children* 48, no. 1 (2015): 45-53. doi:10.1177/0040059915594790

Satabdi Basu, Gautam Biswas, Pratim Sengupta, Amanda Dickes, John S. Kinnebrew, and Douglas Clark. "Identifying Middle School Students' Challenges in Computational Thinking-Based Science Learning," *Research and Practice in Technology Enhanced Learning* 11, no. 13 (2016). doi:10.1186/s41039-016-0036-2

Sepehr Vakil. "A Critical Pedagogy Approach for Engaging Urban Youth in Mobile App Development in an After-School Program," *Equity & Excellence in Education* 47, no. 1 (2014): 31-45. doi:10.1080/10665684.2014.866869