# COMPUTATIONAL THINKING ASSESSMENT GUIDE

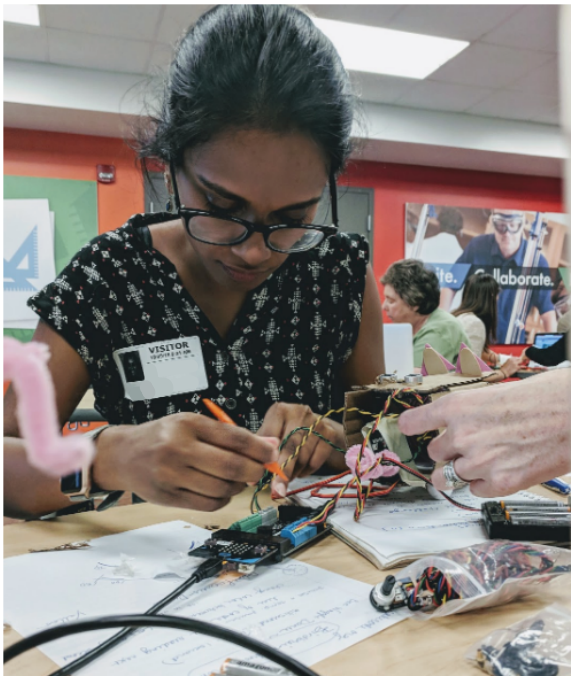
## WHAT IS COMPUTATIONAL THINKING?

**Computational thinking (CT)** is a mindset and a process for problem-solving. This mindset and process are based on methods from computer science. CT is not just math and logic; it is a way of thinking, a set of skills and attitudes.

## HOW TO USE THIS GUIDE

We hope that teachers will use this resource to:

1. **Understand** the mindset and process of CT;
2. **Identify** CT skills in students;
3. **Inform** CT curriculum and instruction.



## INCLUDED IN THIS GUIDE

### CT GLOSSARY

This glossary defines CT skills, breaks them down into observable behaviors, and gives examples of CT in action with the Hummingbird Robotics Kit. Skills include Problem-Solving, Abstraction, and Algorithmic Thinking.

### CT FORMATIVE ASSESSMENT TOOL

Record observations of your students applying CT skills. Use this evidence in portfolios, conferences, & conversations, and to inform CT curriculum and instruction.

# COMPUTATIONAL THINKING GLOSSARY

| **SKILL 1:** PROBLEM SOLVING (PS) | |
|---|---|
| PS1 | **DECOMPOSITION** |
| | **Definition:** Take a large problem and divide it into smaller problems that are each more manageable. This way when each smaller problem is solved, the complex problem becomes manageable. |
| | **Example:** A student wants a robot to perform a set of actions. Instead of coding it all in one large sequence, the student thinks of natural, smaller "abilities" for the robot. The student makes a small sequence for each of these and tests them individually. When each is working properly, the student combines them into a single sequence that uses all these smaller sequences together. |
| PS2 | **REDEFINE PROBLEMS** |
| | **Definition:** Recognize when a given problem cannot be solved with available resources. Express the problem in a different way so that available tools are more applicable. |
| | **Example:** A student is trying to get a robot dog's tail to wag back and forth continuously but cannot get the servo to move slowly enough. Instead of thinking about the problem in terms of a servo moving continuously, the student can create a program that sets the position of the tail to a series of closely spaced positions, and over ten or twenty such moves, it gives the impression of a wagging tail. |
| PS3 | **STRATEGIC DECISION-MAKING** |
| | **Definition:** Compare and weigh possible strategies and solutions, and make a justifiable decision concerning how to proceed. |
| | **Example:** A student wants its robot to blink its eyes 20 times, and considers two ways of doing that: making a sequence that lists out the blinking expression 20 times in a row, or using a numbered repeat loop. The student considers which is more work to implement and which is easier to change later. The student chooses to use the loop, because it will be easier to change the number of blinks later on. |

# COMPUTATIONAL THINKING GLOSSARY

| **SKILL 2:** ABSTRACTION (AB) | |
|---|---|
| AB1 | **MODELING** |
| | **Definition:** Create a model or simulation to represent a complex system in order to better understand the system. Represent key elements of the system while ignoring superfluous details. |
| | **Example:** The student is trying to create a robotic arm that demonstrates the movement of bones and muscles. The student first models the elbow joint of the robot using just the servo and two rectangles of cardboard to test the servo motions, prior to creating bone shaped pieces and adding decorative muscles to the final system. |
| AB2 | **PATTERN RECOGNITION** |
| | **Definition:** Recognize the common features that tasks share. |
| | **Example:** A student is programming a complex robot behavior with numerous smaller sequences. The student recognizes that a desired action is similar to an existing sequence. The student uses the existing sequence as a template and then modifies the sequence as needed without recreating the shared actions. |
| AB3 | **MODULARITY** |
| | **Definition:** Recognize which components may be reusable. Create solutions that are generalizable for multiple tasks. |
| | **Example:** The student is making a robotic mask which expresses different emotions using eye color and mouth position. One solution would be create complete expressions for "Jealous", "Sad", "Angry," and "Tired" which each contain both eye and mouth settings. A modular solution would be to define set eye color and mouth position expressions which can be combined in different ways to create a wide variety of expressions. |

# COMPUTATIONAL THINKING GLOSSARY

| **SKILL 3:** ALGORITHMIC THINKING (AT) | |
|---|---|
| AT1 | **ALGORITHMIC DESIGN** |
| | **Definition:** Develop a step-by-step strategy for solving complex problems. Beyond just making smaller sequences to solve a larger problem, an exceptional student will combine these smaller sequences to create larger and more elaborate final sequences. |
| | **Example:** While designing a robotic lobster, the student recognizes that the "behaviors" will include distinct actions for making the mouth move, snapping the claws, moving eyestalks, and arching the lobster's back. The student is able to plan and describe the relative timing of such behaviors, the sequence of the actions, and their relationships to one another. |
| AT2 | **INCREMENTAL DESIGN AND EVALUATION** |
| | **Definition:** Solve complex challenges by breaking the problem down and implementing simple, manageable parts. Test and perfect each part one-by-one and eventually combine them into the full solution. |
| | **Example:** A student is creating a robotic theater with a four act play. The student designs, tests, and refines each act separately before combining them to create the complete play. |

# COMPUTATIONAL THINKING ASSESSMENT GUIDE

| CODE | DESCRIPTION OF SKILL | OBSERVATIONS |
|---|---|---|
| PS1 | **Decomposition:** Does the student take a large problem and divide it into smaller, more manageable problems? | |
| PS2 | **Redefine Problems:** Does the student redefine a problem such that it may be solved with available tools? | |
| PS3 | **Strategic Decision-Making:** Does the student compare and weigh possible solutions and make justifiable choices? | |
| AB1 | **Modeling:** Does the student create models or simulations which represent key elements of the systems while ignoring superfluous detail? | |
| AB2 | **Pattern Recognition:** Does the student see common features that the tasks share? | |
| AB3 | **Modularity:** Does the student recognize which components are reusable for solving multiple problems? | |
| AT1 | **Algorithmic Design:** Does the student develop a step-by-step strategy for solving complex problems? | |
| AT2 | **Incremental Development and Evaluation:** Does the student solve complex challenges by breaking the problem into smaller manageable parts, solving each problem part and combining those into the full solutions? | |

To put Computational Thinking into action, use one of our **First Hour Lesson Plans!**

Adapted from the "Arts and Bots Talent Definitions" from
Cross, J., Hamner, E., Zito, L., & Nourbakhsh, I. (2016, October). Engineering and computational thinking talent in middle school students: a framework for defining and recognizing student affinities. In Frontiers in Education Conference (FIE), 2016 IEEE (pp. 1-9). doi: 10.1109/FIE.2016.7757720