STEM in the lab

Inspiring Idaho's future STEM workforce

CODING CUPS

GRADE LEVELS

This activity is appropriate for students in grades 7-8.

MISSION

Write and demonstrate code using plastic cups.

VOCABULARY

ALGORITHM: set of instructions given to a computer to perform a specific task. Computers need instructions given in a specific sequence.

CODING: Transforming actions into a symbolic language.

DEBUGGING: Computer programmers don't always get it right the first time. In that case, they need to use 'debugging' to find and fix any mistakes in their code.

PROCESSOR: The "brains" that respond to and process the basic instructions that drive a computer.

SUPERCOMPUTER: A computer that performs billions of tasks at the same time. Supercomputers can use multiple processors to perform tasks simultaneously. Traditionally, supercomputers have been used for scientific and engineering applications that must handle very large databases or do a great amount of computation (or both).

MATERIALS

» 10 plastic cups per group of 4 students

» Piece of paper to draw the code

» Pencil/pen to write the code

ABOUT THIS ACTIVITY

Computer pogrammers at INL use code to tell computers what to do. Programmerswrite instructions, or code, which the computer then executes. Computer programmers solve real word problems by creating code that will analyze the problem and provide solutions. In order to analyze large amounts of data, programmers require very fast computers to run their code. Sawtooth is one supercomputer at INL used to run code and can solve problems in minutes that would take a personal computer years to complete. In this activity, students will write a code and then use plastic cups to demonstrate the code.

WARM-UP
(5 MINUTES)

TALKING TO ROBOTS

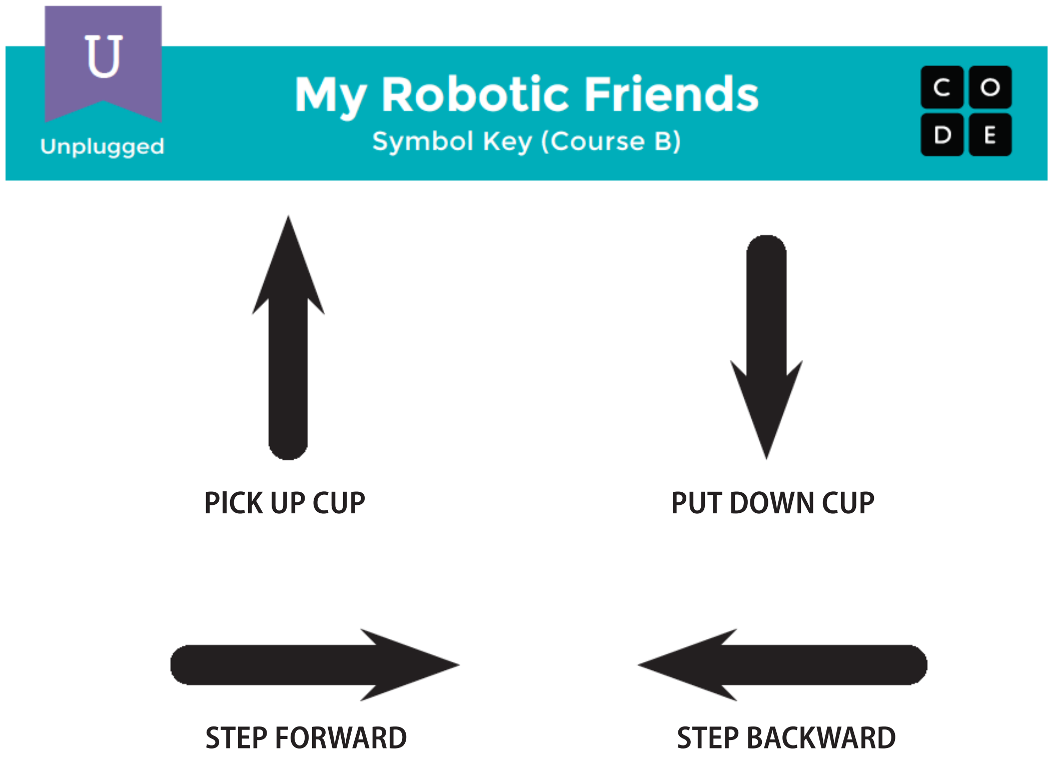
Display:
Watch one of the videos below to give students context for the types of things that robots can do:

- » Asimo by Honda (3:58)
- » Egg Drawing Robot (3:15)
- » Dancing Lego Robot (1:35)

Discuss:
Refer to the video that you chose and ask students how they think that the robot knew what to do. Does a robot really "understand" what you say? Is it worried about getting in trouble if it doesn't do what it's told?

Discussion Goal:
The goal of this quick discussion is to call out that while robots may seem to behave like people, they're actually responding only to their programming. Students will likely refer to robots from movies and TV that behave more like humans. Push them to consider robots that they've seen or heard of in real life, like Roombas, or even digital assistants like Amazon Alexa.

Say:
Robots can only do what they've been told to do, but we don't just tell them using words. In order to do something, a robot needs to have a list of steps that it can read. Today, we are going to learn what it takes to make that happen.



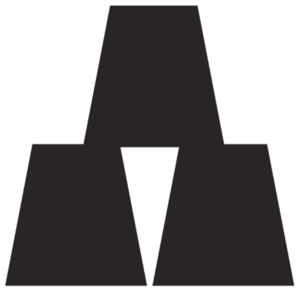
ACTIVITY
(30 MINUTES)

INTRODUCTION AND MODELING

Set Up:
Have stacks of cups or cut paper trapezoids available for groups.

Display:
*Symbol Key or write the allowed actions on the board - make sure these are in a place where they can be seen for the whole activity. Explain to the class that these will be the only four actions that they can use for this exercise. For this task, they will instruct their "robot" friend to build a specific cup stack using only the commands listed on the key.

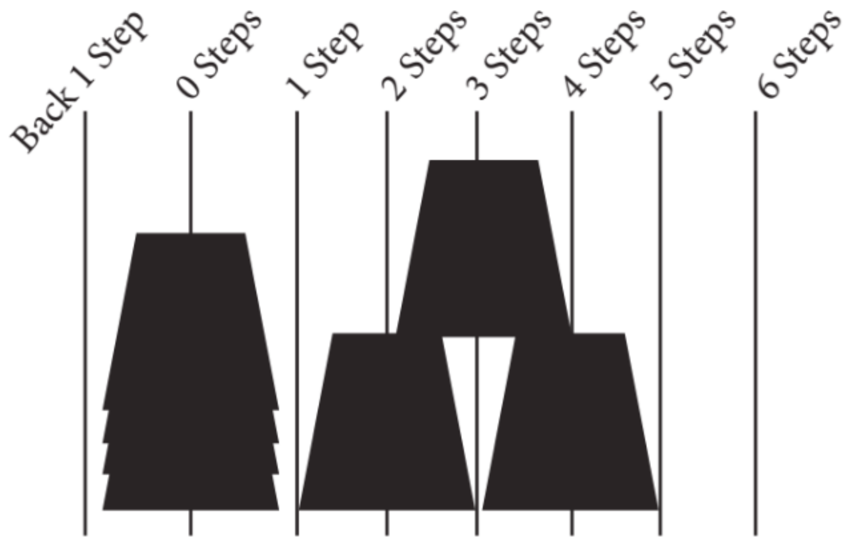
Model:
In order to explain how the instructions are intended to work, model for the class how to create and follow an algorithm for replicating a simple pattern. Place a single stack of cups in front of you to start.



Display:
Hold up the pattern you plan to model. A simple three cup pattern is a great place to start.

Prompt:
Ask the class what the first instruction should be, using only the four instructions allowed. The first move should be to "pick up cup." If students suggest something else from the list, perform that action and allow them to see their error. If they suggest something not from the list, make a clear malfunction reaction and let them know that the command is not understood.

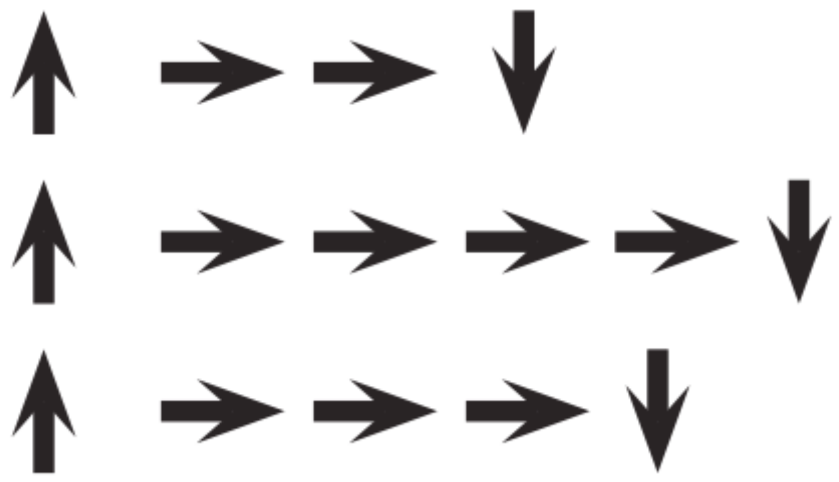
With cup in hand, ask the class to continue giving you instructions until the first cup is placed. This is a great place to clarify that "step forward" and "step backward" each imply moving half a cup width. See the image below for reference.



Continue asking for instructions from the classroom until you have completed the entire design.

Once your stack is complete, point out that they just gave you a list of steps for completing a task. That's an algorithm. Algorithms are great for sharing ideas, but spelling them out word by word can take a long time. That's what the symbols are for! When you change an algorithm into symbols that a robot (or computer) understands, that's called programming.

Ask the class to help you write the "program" for that first move by changing the text into an arrow. Then work with them to write down the rest of the moves necessary to complete the pattern. Depending on the confidence of your students, you might switch back and forth frequently between acting as the "robot" and writing down the code, or you might push them to write the whole program before you will implement it. One possible solution looks like this:



Volunteer:
Once the class has completed the model program, ask one of the students to come up and act as the "robot" to ensure that the program really works. Encourage them to say the instructions out loud as they "run" the code.

PROGRAMMING YOUR ROBOTS

Group:
Place students into groups of 4. Each group should then further break down into two pairs - each pair will develop their own program to be "run" by the other pair.

Distribute:
Give each group one stack of cups or paper cutouts.

Display:
*Cup Stacking Ideas to the class or hand out individual copies for groups to use. Have each pair (not group) choose which design they would like their robots to do. Try to push for an easier design for the first time, then have them choose a more complex design later on. Encourage pairs to keep their choice secret from the other half of their group.

Discuss:
Give each pair time to discuss how the stack should be built, using only the provided symbols. Make sure each group writes down the "program" somewhere for the "robot" to read later.

Do:
Once both of the group's pairs have completed their programs, they can take turns being "robots" for each other by following the instructions the other pair wrote. Encourage students to watch their "robot" closely to ensure that they are following instructions. If a student sees a bug and raises their hand, have the robot finish the instructions to the best of their ability. Afterward, have the students discuss the potential bug and come up with solutions. Continue repeating until the stack is built properly.

Circulate:
Look for groups who are trying to take shortcuts by adding extra things (like numbers) to their code. Praise them for their ingenuity, but remind them that for this exercise, the robots do not understand anything but the provided symbols. If you like, you can hint that they should save their brilliant solution for the next time they play this game, since they might get the chance to use their invention soon!

Iterate:
Depending on the time available, mix up the pairs and give them a chance to do a different pattern. Each time groups repeat the process, encourage them to choose a more challenging pattern.

Discuss:
After everyone has had a chance to be the robot, bring the class back together to discuss their experience. The goal of this discussion is to give students space to make sense of their experience both as robot and programmer. The questions are intentionally broad, but designed to get students thinking about the challenges of writing a clear program and the constraints of a robot or computer in interpreting your instructions. In particular, discuss as a class:

» What was the most difficult part of coming up with the instructions?
» Did anyone find a bug in your instructions once your robot started following them?
» What was the bug?
» Why do you think you didn't notice it when writing the program?
» When you were the robot, what was the hardest part of following the instructions you were given?

WRAP-UP
(10 MINUTES)

REFLECTION

Prompts:
» Draw your own stack of cups that you would like to see a robot build.
» Can you create a program for that cup stack?

THE SCIENCE BEHIND IT

Coding is a precise set of instructions a computer can understand. The instructions or steps that a computer uses to complete a task are called an algorithm. Computer processors use a small number of instructions (around 300) to perform very complicated tasks.

FURTHER EXPLORATIONS

» Create different symbols for your "code."

» Add more cups

» Use colored cups to create different patterns

RESOURCES

» **My Robotic Friends**
https://curriculum.code.org/csf-18/coursee/1/

LEARN MORE

Students + Parents + Educators

For information on grants, training and student opportunities, curriculum ideas, and other resources, please visit ***stem.inl.gov.***