

# Coding Without Computers



45  
minutes



Grades  
6–8, 9–12



Program a human robot to build a pyramid of cups in as few steps as possible.

## Instructions

Using a predefined “robot programming vocabulary” (see below), students will write a program a student “robot” will follow to build cup pyramids as efficiently as possible without verbal conversation. Students learn how computer engineers use the connection between symbols and actions, as well as the valuable skill of debugging code. See examples of this activity in action by searching for “cup stacking coding” on YouTube.

*Note:* Coding and programming essentially have the same meaning. Both refer to the activity of providing any digital platform (like a PC, robot, or browser) a set of instructions on the tasks it needs to perform.

- 1 For preparation, write the programming symbol key and draw the 6-cup pyramid on the board (see diagrams below).
- 2 Ask students how robots know what actions to take. Explain that they are programmed to follow specific step-by-step instructions and are programmed just like computers.
- 3 Tell students they will write code, using only the 6 arrow symbols on the board, that a human “robot” will use to build a cup pyramid. Their goal is to design a program that gets the 6-cup pyramid built in as few steps as possible.



— Pick Up Cup



— Put Down Cup



— Move 1/2 Cup Width Right



— Move 1/2 Cup Width Left



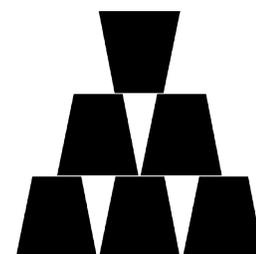
— Rotate Cup 90° Clockwise



— Rotate Cup 90° Counterclockwise

## Materials

- Disposable cups (10 per team)
- Blank paper
- Pencils or pens



4 Begin with a simple example: Stack 2 cups and hold your hand in the air above the stack. Explain that you need them to write a program to tell a robot to get the cups next to each other on the table. Ask the class for the first instruction. The correct answer is “pick up a cup.” (Note: For this activity, “pick up a cup” means to lift the cup above the highest cup already in the stack. Also, a robot may only work with each cup once in a program—there’s no way to instruct the robot to return to a cup that’s already been placed.) With your hand still in the air, ask for the next move. You may have to remind the class that one step forward is only half the width of a cup. Once you’ve placed the cup, ask the class to help you write the symbols on the board to create a program for a robot to follow properly.

5 Split the class into teams of 3 or 4. Each team should choose one robot. Send the robots to the “robot waiting zone” in another part of the classroom while the “programmers” start coding. Robots can use their time to make sure they understand the rules. The programmers should work together to figure out the moves the robot needs to follow to create a 6-cup pyramid. The idea is to make the instructions as efficient as possible by using the fewest steps. Once they have the steps figured out, the team should translate the steps into a written program using the arrow symbols. They should test their program to make sure it works as they expect, and then fix the bugs as needed.

6 Next the group will get the robot back to run the program. The programmers are not allowed to tell the robot what to do at all—the robot must exactly follow the instructions in the program. If the group notices a mistake, they are allowed to halt the program, send the robot back to the waiting zone, and fix the error before bringing the robot back to complete the challenge.

7 Once all the groups are done, have them compare their programs to see how many steps each one has.

8 Now have each group pick a new robot to go to the robot waiting area. Have the programmers draw a complex pyramid with 10 cups. Then have them write a program that will instruct the robot to build this pyramid. The object is to see how accurately the robot builds the pyramid just from the program without seeing the drawing.

9 As time allows, have other students take the role of robot with new pyramids and programs.

## Guiding Questions ?

How can you eliminate unneeded steps from your program?

How will you debug your program before providing it to your robot?

Why is it critical for a computer program to have absolutely no mistakes?



This activity was adapted from the Travelling Circuits lesson “My Robotic Friends,” developed by Thinkersmith and published under the Creative Commons CC-BY-NC-SA license.