

Robotic Buddies

Teacher Materials

Students will learn to convert real-world activities into instructions, gain practice coding instructions with symbols, gain understanding of the need for precision in coding, and gain practice debugging malfunctioning code.

Acknowledgements and License

This activity was contributed by Randy Henderson, McMillan Magnet School for UNO's TED 8006 course.

This activity was adapted with permission from Thinkersmith and Traveling Circuits, <https://csedweek.org/files/CSEdrobotics.pdf>.

This activity is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. Others may freely adapt, reuse, and distribute this for non-commercial purposes provided that they provide attribution.

Lesson Details

Grade Level: 6th-12th graders

Activity Length: 60 minutes

Format: Students working in groups of 3-4, materials for 2 groups

Connection to Standards

This activity contains content and practices addressing elements of the following CSTA K-12 Computer Science Standards (2017):

CSTA Std	Description
1A-AP-08	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.
1A-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
1A-AP-12	Develop plans that describe a program's sequence of events, goals, and expected outcomes.
1A-AP-14	Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.
1B-AP-08	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
1B-AP-11	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
1B-AP-13	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
2-AP-10	Use flowcharts and / or pseudocode to address complex problems as algorithms.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.

This activity contains content and practices addressing elements of the following Nebraska K-12 Technology Scope & Sequence Standards:

NE Tech Standard	Description
Computational Thinking 1	Create algorithms, or series of ordered steps, to solve problems.
Computational Thinking 2	Decompose a problem into smaller more manageable parts.
Computational Thinking 5	Optimize an algorithm for execution by a computer.
Computational Thinking 7	Evaluate algorithms by their efficiency, correctness, and clarity.

Bin Materials Checklist

#	Description
24	Cup Stack Cards
40	Small cups
2	Programming keys
10	Pencils
1	Pencil Sharpener

Lesson Vocabulary Words

algorithm a series of instructions on how to accomplish a task

coding transforming actions into a symbolic language

debugging finding and fixing issues in code

function a piece of code that can be called over and over

parameters extra bits of information that you can pass into a function to customize it

Lesson Setup / Introduction

This activity highlights programming techniques and illustrates the need for precise instructions when creating algorithms.

Using a predefined “Robotic Vocabulary” students will figure out how to guide one another student to accomplish specific tasks **without discussing them first**. They will learn a valuable connection between symbols and actions as it relates to writing computer programs, as well as the valuable skill of debugging.

Limit groups to four students per group, three is ideal. Students generally complete the full round of turns with plenty of time to include the supplement section on functions. The goal is to have enough programmers in each group that the group is never entirely lost.

Connection to Computing

This lesson teaches students the connection between symbols and actions, as well as the invaluable skill of debugging. Programming languages consist of symbols that are then compiled into machine language for the processor to execute (or interpreted for the processor to execute). Being able to accurately describe and order the steps to complete a goal is exactly what programming is all about. This lesson involves the translation of thinking in English to symbols for a computing device. It involves having the students define a goal (the design) and creating the actions needed to accomplish the goal.

This lesson requires collaboration between students. It helps to have good communication skills, especially in the computing industry. The lesson also reinforces decomposition - the ability to break down a problem into smaller pieces. Finally, getting the steps exactly right takes time and patience. The process of finding the errors in the steps (known as debugging in computing) requires problem solving skills. Debugging incorrect instructions is a very valuable skill to have in both programming and testing.

Further Related Resources

- Activity can be done in small groups (with one student as the robot), or as a race. In race mode, some small number of students are the robots and the other students are in teams. Each team comes up with instructions and calls over a robot to follow their directions. If correct, that team gets a point and another card. At the end of a specified amount of time the team with the most points wins. Do make sure that the cup cards are of equal complexity for each team.
- You can also introduce functions with parameters. Instead of writing just a right arrow, instructions can include right arrow (12) indicating that you repeat that move 12 times. For more complicated patterns, this is a lifesaver!
- Creating and Comparing Competing Algorithms
<https://www.flocabulary.com/lesson/Algorithms/f>

Robotic Buddies

Student Instructions

Introduction:

Have you heard of robotics? Has anyone in your group seen a robot or touched one? Does a robot really “hear” you speak? Does it really “understand” what you say?

The answer to the last question is: “Not the same way that a person does.”

All robots operate off of “instructions”, specific sets of commands or actions that they have been pre-programmed to do. In order to accomplish a task, a robot needs to have a series of instructions (sometimes called an algorithm) that it can run. Today, you are going to learn what it takes to make that happen.

Starter Activity:

Take a look at the Programming Key Card. These will be the only six symbols that you will be using for this activity. For this task, you will instruct their “robot buddy” to build a specific cup stack using only these arrows:



Figure 1: Programming Key

Definitions of robotic movement:

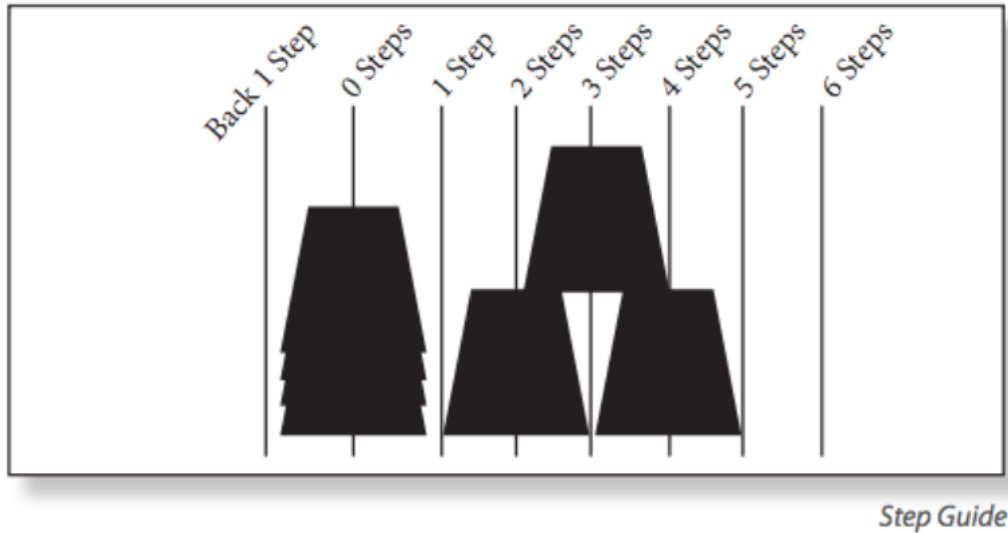


Figure 2: Arm/Hand Movements

Example Program:

Using only symbols from the programming key (Figure 1) try to perform the following programming symbols below. Refer to the movement guide if needed.

What is the first thing to do? *The correct answer is “pick up cup.”*

When your robot picks up each cup, note that the cup should automatically rise above the highest cup already in the stack.

Remember that one step forward is only half the width of a cup.

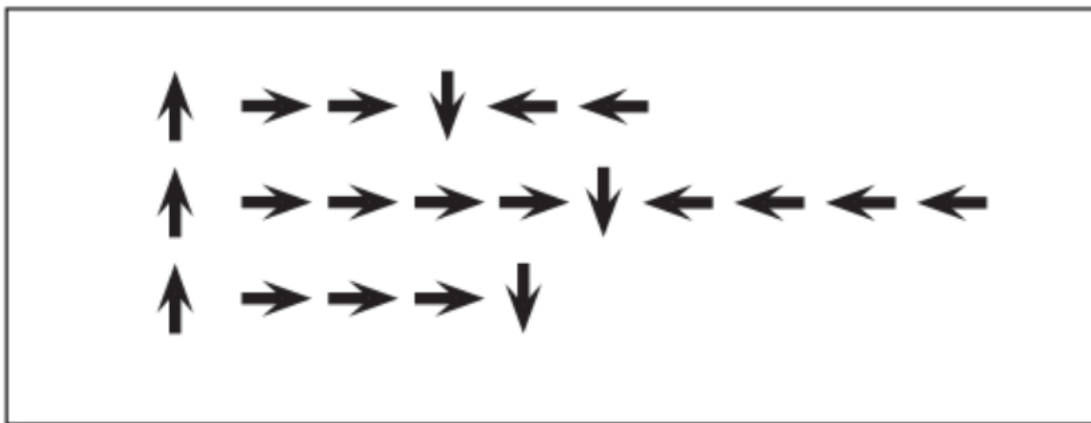


Figure 3: Example Program

After you have finished check with your teacher to see if you have the correct arrangement of cups.

Here is the pseudo code to the program you just performed/ran

```
"Pick up cup", "Step forward", "Step forward", "Put down cup"  
"Step backward", "Step backward"  
  
"Pick up cup", "Step forward", "Step forward", "Step forward",  
"Step forward", "Put down cup", "Step backward", "Step backward",  
"Step backward", "Step backward"  
  
"Pick up cup", "Step forward", "Step forward", "Step forward",  
"Put down cup"
```

Figure 4: Example Pseudo Code

Challenge 1:**Instructions:**

1. Choose one “Robot” per team.
2. Send robot to “Robot Library table” while the “Programmers” sit at the coding table.
3. Robots can use their time in the library to practice cup stacking and to ask for clarification on rules.
4. Programmers will choose one card from the Cup Stack Pack.
5. Programmers will create an algorithm for how the robot should build the selected stack.
6. Programmers will translate their algorithm to arrows, as described in Symbol Key.
7. When programmers have finished coding their stack they can get their robot out of the library.
8. Upon return, the robot reads the symbols from the cards and translates them back into movements.
9. The group should watch for incorrect movements, then work together to debug their program before asking the robot to re-run it.

Rules:

1. Coders should translate all moves using only the six arrows suggested.
2. Cups should remain with the robot, not provided to programmers during coding.
3. **Once robots are back with their groups, there should be no talking out loud.**
4. Start with the cups in one large stack.

Hints:

Once a card is selected you can begin to figure out the algorithm for their stack.

Some questions you might want to ask are:

- How many cups will they need?
- How many steps for the first cup?
- The second?
- Are any cups upside down?
- How do you get the robot to flip a cup?

Once these questions are answered, you can use the symbols to write your code on the blank paper. Programmers should review your code to see if it makes sense for the stack before asking for the robot to come out of the robot library.

Remember:

When the robot is back with the group, everyone should be quiet.

When Completed:

Each time you solve a challenge, you should choose a new robot to head to the library, and the new programmers should be given a new Cup Stack Card. Repeat until time is done or all group members have been robots.

Challenge 2:

Create your own cup stacking drawings for other groups to do.



Hand-Outs



- Robotic Buddies Student Algorithm Worksheet

Write your cup stacking algorithms below...

Cup Stack Pack

Cup Stack Pack

Programming Key



Pick Up Cup



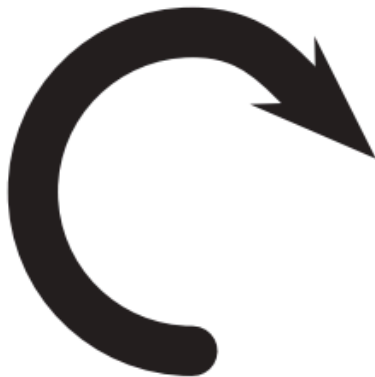
Put Down Cup



Step Forward



Step Backward



Turn Cup Right 90°



Turn Cup Left 90°

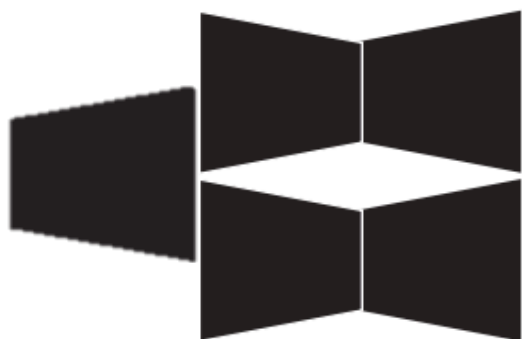
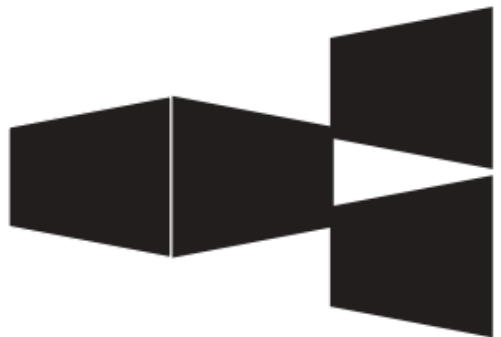
Cup Stack Pack

Cup Stack Pack

Cup Stack Pack



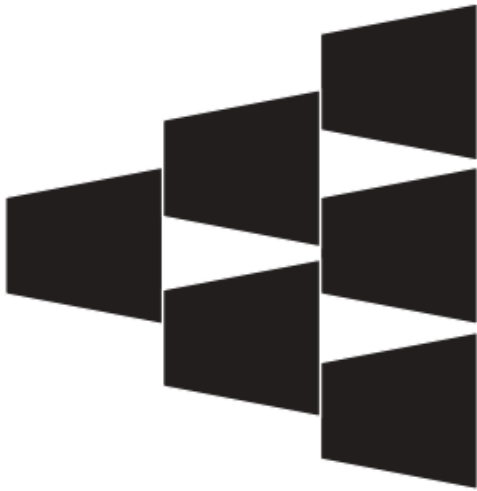
Cup Stack Pack



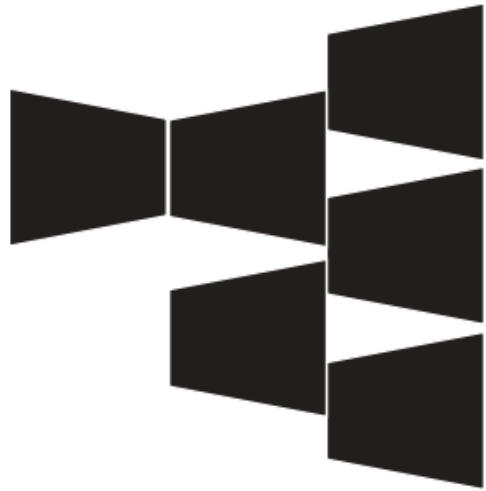
Cup Stack Pack

Cup Stack Pack

Cup Stack Pack



Cup Stack Pack



Cup Stack Pack



Cup Stack Pack



Cup Stack Pack



Cup Stack Pack



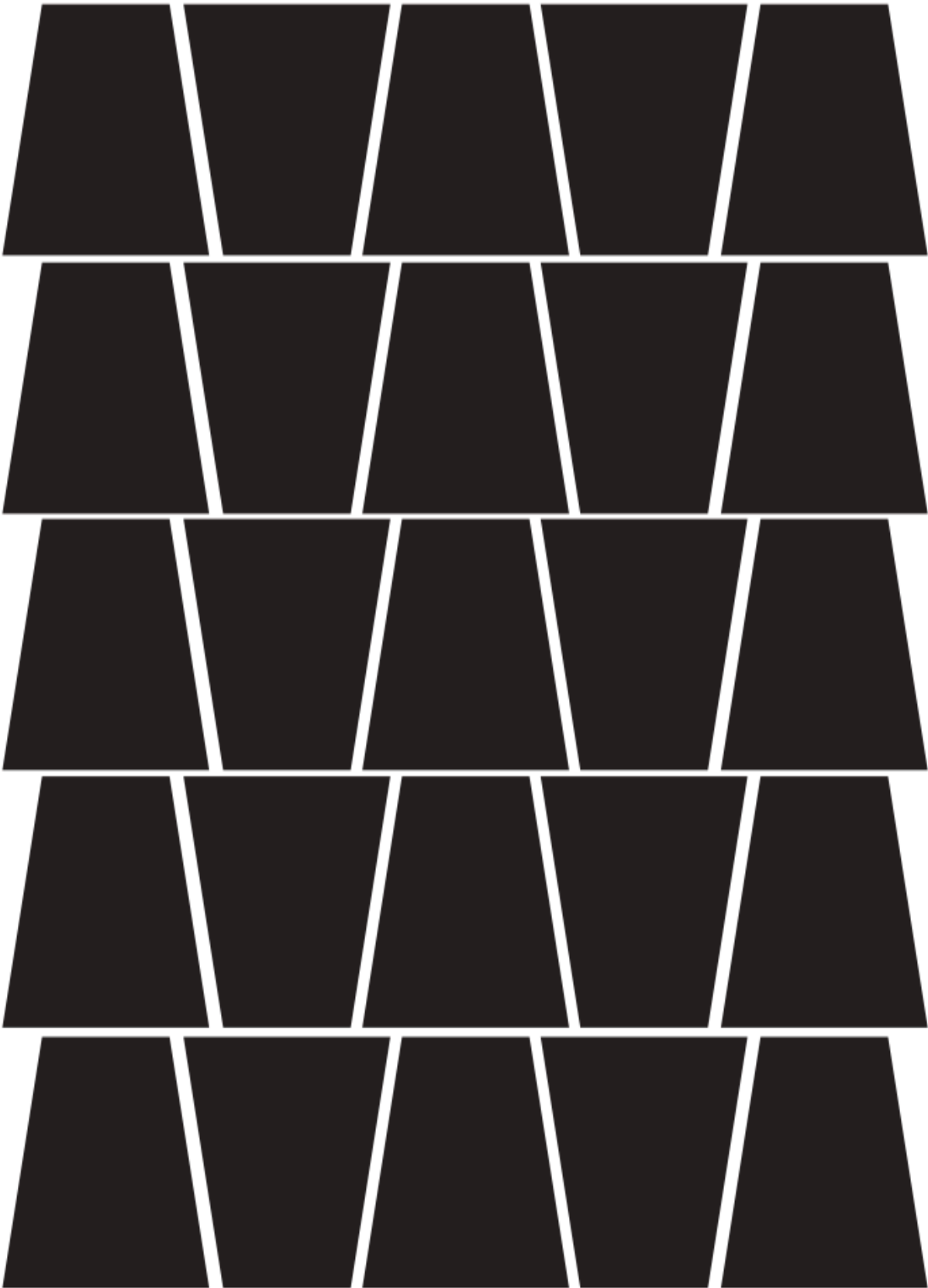
Cup Stack Pack



Cup Stack Pack

Cup Stack Pack

Cup Stack Pack



Cup Stack Pack

Cup Stack Pack



Resupply

- Sharpen and/or Restock Pencils