Learning Coding and Math with RoboBlockly

Teaching Resource for Grade 2 Math

Harry H. Cheng

UC Davis Center for Integrated Computing and STEM Education (C-STEM)

http://c-stem.ucdavis.edu
http://roboblockly.ucdavis.edu
Learning Coding and Math with RoboBlockly

Teaching Resource for Grade 2 Math

Harry H. Cheng
UC Davis Center for Integrated Computing and STEM Education (C-STEM)
University of California-Davis

Copyright © 2015, UC Davis C-STEM Center, All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of the author, including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Acknowledgements:
This work was supported in part by the National Science Foundation under grant numbers CNS-1132709, IIS-1208690, IIS-1256780, and by the CaMSP Program of the California Department of Education.

Additional Contributor(s):
Kayce Mastrup
How to Use RoboBlockly Activities in your Classroom

**What is the purpose of RoboBlockly?**

In RoboBlockly, students program a robot using a user-friendly drag-and-drop method. Students will complete basic problem solving to move a robot or multiple robots.

RoboBlockly is built to allow students to work at their own pace, but in general each problem should take approximately 5 minutes to complete each activity. We encourage you to give students additional time if needed or make it clear that they don’t need to finish the entire set of activities during one class session.

**General Computer Usage Requirements**

**Technology Requirements:** Any modern browser on computers, laptops, tables, or smartphones with any type of operating system.

It is important to know that every browser functions differently. We encourage you to test RoboBlockly on the computers you will have students using before implementation. Please test out the following: audio and video streaming quality, default browser specific mechanisms for saving blocks and saving Ch code, etc. all so you are better able to support your students. Make sure that pop ups have been enabled on all computers.

You may wish to provide headphones or ask students to bring headphones to allow students to independently watch tutorial videos.

**Prepare yourself**

Go through the activities yourself so that you are familiar with what your students will be experiencing. The Teacher Resource Packet contains all the activities and solutions for the pathway. Please note that the activities build on previous activities in each pathways such that students may need to complete some or all activities prior to the activity selected.

1) Determine the purpose for students using RoboBlockly:
   - To support student learning in Math,
   - To support student learning in Computer Programming,
   - To support student learning in Robotics.

2) Based on your purpose, determine what additional resources your students will need for instruction. We do not recommend using RoboBlockly to introduce a mathematical concept but to rather use it for skill building or as a culminating performance task.

   **Use as skill building:** We recommend that you provide your students with a worksheet that includes important related definitions, work space, leading questions, etc. and encourage your students to refer to their class notes which cover these topics.

   **Use as a culminating performance task:** Carefully select which activity directly relates to the content you have taught, making note that the previous activities may be necessary to complete to build prior knowledge.

**Prepare your students**

Help students get excited about RoboBlockly by inspiring students and discussing how computer science impacts every part of our lives. As a class, list things that use code in everyday life, or discuss different ways technology impacts our lives etc.

When using RoboBlockly in class, first demonstrate to students how to navigate and use the RoboBlockly website. There are five Video Tutorials, along with a self-guided interactive non-video tutorial which should be used to help familiarize your students with the different functionalities of RoboBlockly. Helping students understand the functionality of RoboBlockly and which elements can be manipulated in which manners is very important to ensuring your students have full access to the content.
Pre-Requisite Skills

Math
We are currently developing a comprehensive wiring guide to assist you with your planning. Please refer to the Table of Contents Standard Mapping for a complete list of Common Core Grade 2 Math Standards addressed in the RoboBlockly activities.

Computer

- **Basic computer skills:**
  - Drag and drop using a mouse
  - Key boarding
  - Navigating a web browser
  - Zoom In/Out in a browser
  - Disabling or enabling pop-up windows
  - Adjusting volume for videos

Extension

Using hardwired robots, Linkbot Controller, RoboSim or Robot Controller to execute programs built in RoboBlockly.

All can be downloaded from the UC Davis C-STEM Center’s webpage: [http://c-stem.ucdavis.edu/downloads/](http://c-stem.ucdavis.edu/downloads/)
# Learning Coding and Math with RoboBlockly

*Teaching Resource for Grade 2 Math*

## Table of Contents: Activity Description

<table>
<thead>
<tr>
<th>Activity</th>
<th>CCSSM</th>
<th>Blocks Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Applying Addition to Drive the Robot Forward</td>
<td>2.OA.2</td>
<td>driveDistance</td>
</tr>
<tr>
<td>2. Applying Subtraction to Drive the Robot Forward</td>
<td>2.OA.2</td>
<td>driveDistance</td>
</tr>
<tr>
<td>3. Applying Addition with Multiple Terms to Drive the Robot Forward</td>
<td>2.OA.2</td>
<td>driveDistance, turnRight</td>
</tr>
<tr>
<td>4. Applying Subtraction with Multiple Terms to Drive the Robot Forward</td>
<td>2.OA.2</td>
<td>driveDistance</td>
</tr>
<tr>
<td>5. Using Addition to Group Objects: How Many Squares?</td>
<td>2.OA.4</td>
<td>driveDistance, turnRight</td>
</tr>
<tr>
<td>6. Using Addition to Group Objects: How Many Triangles?</td>
<td>2.OA.4</td>
<td>driveDistance, turnLeft</td>
</tr>
<tr>
<td>7. Using Addition to Group Objects: How Many Non-Circles?</td>
<td>2.OA.4</td>
<td>driveDistance</td>
</tr>
<tr>
<td>8. Working with Money: How Many Cents do you have?</td>
<td>2.MD.8</td>
<td>driveDistance</td>
</tr>
<tr>
<td>9. Reasoning with Shapes: How Many Sides are there?</td>
<td>2.G.1</td>
<td>driveDistance, turnLeft</td>
</tr>
</tbody>
</table>
Learning Coding and Math with RoboBlockly

Teaching Resource for Grade 2 Math

Table of Contents: Standard Mapping

<table>
<thead>
<tr>
<th>Common Core State Standards for Mathematics – Grade 1</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.OA.2</strong> Fluently add and subtract within 20 using mental strategies. By end of Grade 2, know from memory all sums of two one-digit numbers.</td>
<td>1 – 4</td>
</tr>
<tr>
<td><strong>2.OA.4</strong> Use addition to find the total number of objects arranged in rectangular arrays with up to 5 rows and up to 5 columns; write an equation to express the total as a sum of equal addends.</td>
<td>5 – 7</td>
</tr>
<tr>
<td><strong>2.MD.8</strong> Solve word problems involving dollar bills, quarters, dimes, nickels, and pennies, using $ and ¢ symbols appropriately. Example: If you have 2 dimes and 3 pennies, how many cents do you have?</td>
<td>8</td>
</tr>
<tr>
<td><strong>2.G.1</strong> Recognize and draw shapes having specified attributes, such as a given number of angles or a given number of equal faces. Identify triangles, quadrilaterals, pentagons, hexagons, and cubes.</td>
<td>9</td>
</tr>
</tbody>
</table>

Textbook to Activity Alignment

<table>
<thead>
<tr>
<th>Learning Robot Programming with Linkbot for the Absolute Beginner 5th Edition</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 5.4 Move a Distance for a Two-Wheel Robot</td>
<td>1 2 3 4 5 6 7 8 9</td>
</tr>
<tr>
<td>Section 5.6 Turn Left and turn right</td>
<td>X</td>
</tr>
</tbody>
</table>
Activity #1 Applying Addition to Drive the Robot Forward

Common Core State Standards - Mathematics:
2.OA.2 Fluently add and subtract within 20 using mental strategies. By end of Grade 2, know from memory all sums of two one-digit numbers.

Objective: Students will apply their understanding of addition and subtraction to move the linkbot the specified distance.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>To drive the robot forward you use the driveDistance() block:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="driveDistance.png" alt="driveDistance block" /></td>
</tr>
</tbody>
</table>

For your activities you will need to change the numbers inside the blue boxes in order for the robot to reach the correct stopping point. You also need to see what operation you will be using.

Have Fun!

<table>
<thead>
<tr>
<th>Pre-Placed Blocks</th>
<th><img src="pre-placed.png" alt="pre-placed blocks" /></th>
</tr>
</thead>
</table>

| Problem Statement | Modify the preplaced blocks so the robot will drive forward 18 units. You can change one or both of the numbers, just make sure your numbers add up to 18. |

| Wrong Prompt | Sorry, you did not drive the robot the correct total distance. Please try again – use the Hint if you feel stuck. |

| Hint | You want to drive the robot a total of 18 units. The current block will drive the robot: $12 + 12 = 24$ units. Think “what should the values be to drive the robot 18 units?” OR “what two numbers add up to 18?” |

| Possible Solution in C | `#include <linkbot.h>`
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>CLinkbotI robot;</code></td>
</tr>
<tr>
<td></td>
<td><code>double radius = 1.75;</code></td>
</tr>
<tr>
<td></td>
<td><code>robot.driveDistance(12 + 6, radius);</code></td>
</tr>
</tbody>
</table>
Activity #1 Applying Addition to Drive the Robot Forward

Picture of solution in RoboBlockly

Problem Statement:
Modify the preplaced blocks so the robot will drive forward 18 units. You can change one or both of the numbers, just make sure your numbers add up to 18.

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m1.xml

Student Mathematical Calculations
Students will need to first see that the pre-placed blocks have a sum of 24: \[12 + 12 = 24\]

They then need to change the values to any two numbers that have a sum of 18. Here are all the possible combinations (for whole numbers less than 20):

\[
\begin{align*}
0 + 18 &= 18 \\
1 + 17 &= 18 \\
2 + 16 &= 18 \\
3 + 15 &= 18 \\
4 + 14 &= 18 \\
5 + 13 &= 18 \\
6 + 12 &= 18 \\
7 + 11 &= 18 \\
8 + 10 &= 18 \\
9 + 9 &= 18
\end{align*}
\]

C-STEM text alignment: Robot Programming with Linkbot for the Absolute Beginner, 5th edition
a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
Activity #2 Applying Subtraction to Drive the Robot Forward

Common Core State Standards - Mathematics:
2.OA.2 Fluently add and subtract within 20 using mental strategies. By end of Grade 2, know from memory all sums of two one-digit numbers.

Objective: Students will apply their understanding of addition and subtraction to move the linkbot the specified distance.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>Change the values in the driveDistance block so your robot drives the correct distance. This time you will be using subtraction.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Placed Blocks</td>
<td><img src="driveDistance.png" alt="Image" /></td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Modify the preplaced blocks so the robot will drive forward 13 units. You can change one or both of the numbers, just make sure your numbers add up to 13.</td>
</tr>
<tr>
<td>Wrong Prompt</td>
<td>Sorry, you did not drive the robot the correct total distance. Please try again – use the Hint if you feel stuck.</td>
</tr>
<tr>
<td>Hint</td>
<td>You want to drive the robot a total of 13 units. The current block will drive the robot to: 20 – 5 = 15 units. Think “what should the values be to drive the robot 13 units?” OR “what two numbers can I subtract to equal 13?”</td>
</tr>
<tr>
<td>Possible Solution in Code</td>
<td>#include &lt;linkbot.h&gt;</td>
</tr>
<tr>
<td></td>
<td>CLinkbotI robot;</td>
</tr>
<tr>
<td></td>
<td>double radius = 1.75;</td>
</tr>
<tr>
<td></td>
<td>robot.driveDistance(20 - 7, radius);</td>
</tr>
</tbody>
</table>
Activity #2 Applying Subtraction to Drive the Robot Forward

Problem Statement:
Modify the preplaced blocks so the robot will drive forward 13 units. You can change one or both of the numbers, just make sure your numbers add up to 13.

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m2.xml

Student Mathematical Calculations
Students will need to first see that the pre-placed blocks have a difference of 15: 
20 – 5 = 15

They then need to change the values to any two numbers that have a sum of 13. Here are all the possible combinations (for whole numbers less than 20):
20 – 7 = 13
19 – 6 = 13
18 – 5 = 13
17 – 4 = 13
16 – 3 = 13
15 – 2 = 13
14 – 1 = 13
13 – 0 = 13

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
Activity #3 Applying Addition with Multiple Terms to Drive the Robot Forward

Common Core State Standards - Mathematics:
2.OA.2 Fluently add and subtract within 20 using mental strategies. By end of Grade 2, know from memory all sums of two one-digit numbers.

Objective: Students will apply their understanding of addition and subtraction to move the linkbot the specified distance.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>This time you will have multiple operations happening in your driveDistance block. Operations will occur from left to right – remember the order in which you solve the problem matters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Placed Blocks</td>
<td><img src="driveDistance.png" alt="Image" /></td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Modify the preplaced blocks so the robot will drive forward 14 units. You can change all of the numbers or one, just make sure that all three numbers simplify to 14.</td>
</tr>
<tr>
<td>Wrong Prompt</td>
<td>Sorry, you did not drive the robot the correct total distance. Please try again – use the Hint if you feel stuck.</td>
</tr>
<tr>
<td>Hint</td>
<td>You want to drive the robot a total of 14 units. The current block will drive the robot: 15 – 5 + 12 = 22. Think “what should the values be to drive the robot 14 units?” You have multiple operations so figure out your first difference to determine what number you should add on to reach 14.</td>
</tr>
<tr>
<td>Possible Solution in C</td>
<td><code>#include &lt;linkbot.h&gt;</code>&lt;br&gt;<code>CLinkbotI robot;</code>&lt;br&gt;<code>double radius = 1.75;</code>&lt;br&gt;<code>robot.driveDistance((7 - 5)+ 12, radius);</code></td>
</tr>
</tbody>
</table>
Activity #3 Applying Addition with Multiple Terms to Drive the Robot Forward

Problem Statement:
Modify the preplaced blocks so the robot will drive forward 14 units. You can change all of the numbers or one, just make sure that all three numbers simplify to 14.

Student Mathematical Calculations
Students will need to first see that the pre-placed blocks simplify to 22:

\[ 15 - 5 + 12 = 22 \]

There are LOT of possible solutions to this problem.

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*

a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
Activity #4 Applying Subtraction with Multiple Terms to Drive the Robot Forward

Common Core State Standards - Mathematics:
2.OA.2 Fluently add and subtract within 20 using mental strategies. By end of Grade 2, know from memory all sums of two one-digit numbers.

Objective: Students will apply their understanding of addition and subtraction to move the linkbot the specified distance.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>Just as before, you will have multiple operations happening in your driveDistance block. Operations will occur from left to right (important to note for subtraction).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Placed Blocks</td>
<td><img src="image.png" alt="Pre-Placed Blocks Image" /></td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Modify the preplaced blocks so the robot will drive forward 11 units. You can change all of the numbers or one, just make sure that all three numbers simplify to 11. Perform your operations from left to right.</td>
</tr>
<tr>
<td>Wrong Prompt</td>
<td>Sorry, you did not drive the robot the correct total distance. Please try again – use the Hint if you feel stuck.</td>
</tr>
<tr>
<td>Hint</td>
<td>You want to drive the robot a total of 11 units. The current block will drive the robot: 28 – 11 + 4 = 21. Think “what should the values be to drive the robot 11 units?”. Figure out your first difference then what you will need to add to that to get to 11.</td>
</tr>
</tbody>
</table>
| Possible Solution in Ch| ```
#include <linkbot.h>
CLinkbotI robot;
double radius = 1.75;

robot.driveDistance((28 - 21) + 4, radius);
``` |
Activity #4 Applying Subtraction with Multiple Terms to Drive the Robot Forward

Picture of solution in RoboBlockly

Problem Statement:
Modify the preplaced blocks so the robot will drive forward 11 units. You can change all of the numbers or one, just make sure that all three numbers simplify to 11. Perform your operations from left to right.

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m4.xml

Student Mathematical Calculations
Students will need to first see that the pre-placed blocks simplify to 21: $28 - 11 + 4 = 21$
There are a LOT of possible solutions.

C-STEM text alignment: *Robot Programming with Linkbot for the Absolute Beginner, 5th edition*
  a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
**Activity #5 Using Addition to Group Objects: How Many Squares?**

**Common Core State Standards - Mathematics:**
2.OA.4 Use addition to find the total number of objects arranged in rectangular arrays with up to 5 rows and up to 5 columns; write an equation to express the total as a sum of equal addends.

**Objective:** Students will apply their understanding of addition and multiplication to find the area of the rectangle drawn by the robot.

**RoboBlockly Student Activity:**

<table>
<thead>
<tr>
<th><strong>Initial Student Prompt</strong></th>
<th>There are many cool things you can do in RoboBlockly. For this problem you will see three new blocks: Allows you to turn the robot left or right:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Allows you to ask for input from the user:</td>
</tr>
<tr>
<td></td>
<td>For this problem you will only run the program – then you will need to enter in your answer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Pre-Placed Blocks</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Pre-Placed Blocks" /></td>
</tr>
</tbody>
</table>

| **Problem Statement** | Run the program using the pre-placed blocks, do not modify any of the blocks. Use the drawing to the left to answer the following question: “How many small squares are inside the rectangle?” You will type your answer in the dialogue box that opens. |

| **Wrong Prompt** | Sorry you did not count the number of squares correctly, please try again. |

| **Hint** | After the program completes, count how many small squares are inside the rectangle. There are three rows and eight columns |

<table>
<thead>
<tr>
<th><strong>Possible Solution in C</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>#include &lt;linkbot.h&gt;</td>
</tr>
<tr>
<td>double variable;</td>
</tr>
<tr>
<td>CLinkbot robot;</td>
</tr>
<tr>
<td>double radius = 1.75;</td>
</tr>
</tbody>
</table>
double trackwidth = 3.69;

robot.driveDistance(8, radius);
robot.turnLeft(90, radius, trackwidth);
robot.driveDistance(3, radius);
robot.turnLeft(90, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(90, radius, trackwidth);
robot.driveDistance(3, radius);

printf("How many small squares are inside the rec rectangle?\n");
scanf("%lf", &variable);
Activity #5  Using Addition to Group Objects: How Many Squares?

Problem Statement:
Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many small squares are inside the rectangle?”

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m5.xml

Student Mathematical Calculations
Students will run the preplaced blocks – there is no need to alter or change the blocks. The robot’s path will create an 8 by 3 rectangle. The students are to calculate how many small squares are inside the rectangle.

There are many approaches to solve this problem. One approach is to realize there are 3 rows and 8 columns – so 3 groups of 8 which equals 24 OR 8 groups of 3 which also equals 24.

C-STEM text alignment: Robot Programming with Linkbot for the Absolute Beginner, 5th edition
a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
b) Section 5.6 Turn Left and turn right (turnLeft block)
Activity #6 Using Addition to Group Objects: How Many Triangles?

Common Core State Standards - Mathematics:
2.OA.4 Use addition to find the total number of objects arranged in rectangular arrays with up to 5 rows and up to 5 columns; write an equation to express the total as a sum of equal addends.

Objective: Students will apply their understanding of shapes to identify all the triangles out of a group of different shapes.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>You can also draw in RoboBlockly (you’ll learn that in a different lesson)! Run the preplaced blocks and use the drawing to answer the question.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Placed Blocks</td>
<td>![Prompt for variable with message “How many pink triangles are there?”]</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many triangles are displayed on the grid?”</td>
</tr>
<tr>
<td>Wrong Prompt</td>
<td>Sorry, you did not count up the triangles correctly, please try again.</td>
</tr>
<tr>
<td>Hint</td>
<td>Remember that a triangle is a three sided shape. It has three vertices and 3 edges.</td>
</tr>
</tbody>
</table>
| Possible Solution in Ch| ```c
#include <chplot.h>

double variable;

printf("How many pink triangles are there?\n");
scanf("%lf", &variable);
```
**Activity #6 Using Addition to Group Objects: How Many Triangles?**

**Problem Statement:**
Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many triangles are displayed on the grid?”

**Student Mathematical Calculations**
Students will need to recognize which shapes are the triangles and enter in the correct value to the pop-up box.

There are:
- 3 rectangles
- 5 triangles
- 4 circles
Activity #7 Using Addition to Group Objects: How Many Non-Circles?

Common Core State Standards - Mathematics:
2.OA.4 Use addition to find the total number of objects arranged in rectangular arrays with up to 5 rows and up to 5 columns; write an equation to express the total as a sum of equal addends.

Objective: Students will apply their understanding of shapes to identify all the non-circular shapes out of a group of shapes.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>Let’s try another shape problem! Run the preplaced blocks and use the drawing to answer the question.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Placed Blocks</td>
<td><img src="image" alt="Prompt for variable with message" /> How many non-circular shapes are there?</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many non-circular shapes are on the grid?”</td>
</tr>
<tr>
<td>Wrong Prompt</td>
<td>Sorry, you did not count up the non-circular items correctly, please try again.</td>
</tr>
<tr>
<td>Hint</td>
<td>There are circles, triangles and rectangles on the grid. Triangles and rectangles are non-circular.</td>
</tr>
</tbody>
</table>
| Possible Solution in C | ```c
#include <chplot.h>
double variable;

printf("How many non-circular shapes are on the grid?\n");
scanf("%lf", &variable);
``` |
Activity #7 Using Addition to Group Objects: How Many Non-Circles?

Problem Statement:
Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many non-circular shapes are on the grid?”

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m7.xml

Student Mathematical Calculations
Students will need to recognize which shapes are not circles and count them.

There are:
3 rectangles
5 triangles
4 circles

Total of 8 non-circular shapes.
### Activity #8 Working with Money: How Many Cents do you have?

**Common Core State Standards - Mathematics:**

2.MD.8 Solve word problems involving dollar bills, quarters, dimes, nickels, and pennies, using $ and ¢ symbols appropriately. *Example: If you have 2 dimes and 3 pennies, how many cents do you have?*

**Objective:** Students will apply their understanding of addition to find the total amount of change displayed on the grid.

**RoboBlockly Student Activity:**

<table>
<thead>
<tr>
<th><strong>Initial Student Prompt</strong></th>
<th>Let’s try another program that has drawings. This time we will use change! You will need to find the sum of the change drawn after running the program.</th>
</tr>
</thead>
</table>
| **Pre-Placed Blocks**     | ![Coins](image)
| **Problem Statement**     | Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “What is the total amount of cents displayed on the grid?” |
| **Wrong Prompt**          | You did not find the correct total of the change. Please try again. |
| **Hint**                  | You have 2 pennies, 1 nickel, 1 dime and 1 quarter. Add up there amounts $1 + 1 + 5 + 10 + 25$
| **Possible Solution in C**| ```c
#include <chplot.h>

double variable;

printf("What is the total amount of change?\n");
scanf("%lf", &variable);
```
Activity #8 Working with Money: How Many Cents do you have?

Picture of solution in RoboBlockly

Problem Statement:
Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “What is the total amount of cents displayed on the grid?”

The page at roboblockly.ucdavis.edu says:

What is the total amount of change?

[Input field]

[Options]

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m8.xml

Student Mathematical Calculations
Students will need to apply their understanding of counting change. The program will display:

- 2 pennies = 0.02
- 1 nickel = 0.05
- 1 dime = 0.10
- 1 quarter = 0.25
- Total = 0.42
Activity #9  Reasoning with Shapes: How Many Sides are there?

Common Core State Standards - Mathematics:
2.G.1 Recognize and draw shapes having specified attributes, such as a given number of angles or a given number of equal faces. Identify triangles, quadrilaterals, pentagons, hexagons, and cubes.

Objective: Students will apply their understanding identifying attributes of a shape.

RoboBlockly Student Activity:

<table>
<thead>
<tr>
<th>Initial Student Prompt</th>
<th>For this activity, the robot will draw a shape and you will need to identify how many sides (edges) the shape has. Below is a picture to remind you where the edges (edges) are on a shape.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Placed Blocks</td>
<td><img src="image" alt="Pre-Placed Blocks Image" /></td>
</tr>
<tr>
<td>Problem Statement</td>
<td>Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many sides does the shape on the grid have?”</td>
</tr>
<tr>
<td>Wrong Prompt</td>
<td>Sorry, you did not count the number of sides up correctly. Please try again.</td>
</tr>
</tbody>
</table>
**Hint**
Remember a side connects the corners of the shape. A square has four sides, as shown in this picture:

```
+-----+-----+-----+
|     |     |     |
+-----+-----+-----+
|     |     |     |
+-----+-----+-----+
```

**Possible Solution in C**
```
#include <linkbot.h>
double variable;
CLinkbotI robot;
double radius = 1.75;
double trackwidth = 3.69;

robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);
robot.turnLeft(45, radius, trackwidth);
robot.driveDistance(8, radius);

printf("How many sides does the shape have?\n");
scanf("%lf", &variable);
```
Activity #9 Reasoning with Shapes: How Many Sides are there?

Picture of solution in RoboBlockly

Problem Statement:
Run the program using the pre-placed blocks. Use the drawing to the left to answer the following question: “How many sides does the shape on the grid have?”

Location of solution for “Load Blocks” tab in RoboBlockly
C-STEM Studio -> Teaching Resources -> TeachMath2 -> RoboBlocklySolution -> m9.xml

Student Mathematical Calculations
Students will need to run the preplaced code and then count the number of sides/edges of the shape drawn. They will enter 8 into the pop up box.

C-STEM text alignment: Robot Programming with Linkbot for the Absolute Beginner, 5th edition
a) Section 5.4 Move a Distance for a Two-Wheel Robot. (driveDistance block)
b) Section 5.6 Turn Left and turn right (turnLeft block)