**Lesson 1: Algorithms**

This lesson is an optional lesson. It is a great introduction lesson if your students have no understanding of coding.

**Problem statement:** Your team must design a new game and present your product to the leadership team of Games R Us, your goal is to get approval from the Games R Us executives to produce your game.

This lesson will teach students the basics of how to make a ball bounce using Scratch.

**Learning objectives:**

* Students be able to define algorithm, program, and programmer.
* Students will be able to verbally instruct a partner through step-by-step instructions in marking a graph.

**Lesson standards (NGSS, CCSS, CTE):**

**Washington State Computer Science Standards**

**1B-A-3-7 --** Construct and execute an algorithm (set of step-by-step instructions) that includes sequencing, loops, and conditionals to accomplish a task, both independently and collaboratively, with or without a computing device.

**Common Core Mathematical Processes**

**6**  -- Attend to precision.

**Soft skills:**

**Partnership for 21st Century Learning**

* Communicate clearly
* Interact effectively

C**onnections to career and educational pathways:**

Software Developer/Programmer - Students learn that programmers are people that write programs to run on a computer and that programs are algorithms, or a set of instructions, that a computer understands.

**Materials:**

* Student worksheet
* Index cards or paper (for students to create a grid)
* Pencil

**Lesson preparation:**

**Time required:**

40 minutes

**Grouping of students for instruction:**

Students will be need to be in partners of two for the main activity. Students will each play the role of the “Programmer” and the “Computer.”

**What is the instruction? Consider the PBL Procedure that is being addressed here:** Students will be understanding, exploring, and resolving the challenge. Students will understand the concept of what an algorithm is and will create “unplugged” versions on paper.

**Lesson:**

1. **The Teacher will ask the students:** “Have you ever followed a recipe to make something? How do you know how much of an ingredient to add? How do you know when to bake it in the oven?”
2. **The teacher:** “A list of steps that you follow to finish a task is also known as an algorithm. Can you think of other examples?” (Ex: how to brush your teeth, hot to make a paper airplane, and addition problem in math, a science experiment, etc.)
3. **The teacher** will have the students give instructions to them on how to draw a smiley face on the board. **The students** will give instructions sharing ideas of step-by-step instructions. **The teacher** will explain that many tasks can be described using a specific list of instructions. That list is called an algorithm.
4. **The teacher** will share that a **program** is an **algorithm**, or set of instructions, that a computer understands. A **programmer** is a person that write programs to run on a computer.
5. Activity:

* **The teacher** will give each student an index card or small square piece of paper and will ask them to draw lines to divide it into a 4 x 4 grid on each side.
* On both sides of the paper, have the students draw a star in the top left square. That is the “start” square.
* On one side of their paper, have the students randomly select a square to mark with an “X.”
* Working in pairs, have students sit back to back so that they cannot see each other’s grid.
* Have one student act as the “Programmer” and one student to act as the “Computer.”
* Programmer instructions:
  + Looking at your grid, starting at the star in the upper left-hand corner, give instructions to the “Computer,” one-step-at-a-time, to help them navigate to your secret square that you marked with an “X.” Tell them to draw a “?” in that square.
* Computer instructions:
  + Looking at your blank grid, starting in the starred square, listen to the “Programmers” instructions. Mark a “?” in the square at the end of the exercise.
  + “Computers” may NOT ask questions. You can only follow instructions from the “Programmer.”
* Have the students flip over their grids and switch roles and repeat the activity.

1. Have the students debrief the activity:

* How many ‘Computers’ landed on the correct square?
* What did the ‘Programmers’ do that helped the ‘Computers’ mark the correct square?
* What could the ‘Programmers’ have done differently (if you were unsuccessful as a team)?

1. Have the students complete the algorithm worksheet.

**Accommodations:**

For students with visual impairment: grids can be enlarged or created using either wiki sticks or textured paper to create a raised surface.

For students who are ELL: algorithms can be created using visuals that they can put in sequential order; students may be grouped with a partner who can help assist them.

**Extensions:**

* **https://www.brainpop.com/technology/computerscience/computerprogramming/**

**Assessment:**

* Formative Assessment in the Lessons: You will be able to informally assess students as they work with their partners and as they work on their student worksheet.