Intro to Programming

Objectives: 1) Verbally instruct a human 'robot' in how to make a peanut butter sandwich, 2) Take on the role of a 'robot' and sort a packet of Starbursts by following a script generated by other students.

Concepts: Algorithmic (step-by-step) thinking, clearly defining operational parameters, for loops, if-then statements.

Setup:

Materials: Whiteboard, Jar of peanut butter, butter knife, loaf of bread, pack of Starburst, computer, paper tow

Teacher Preparation: The head mentor will print out the lesson plan for every group leader, group leaders need to bring energy to the discussion

Classroom Preparation: Distribution of peanut butter, bread slices, and starburst packages, lay out whiteboards, markers, and erasers. Every computer must be set up to immediately run the 'starburst' code.

In the Classroom:

Warm-up Activity (10 minutes): Whole group discussion on the differences and similarities between computers and brains

Lesson Introduction/Description (10 minutes): Break into groups. Ask the students if they know why we use computers in science. Play with them for a bit with the question (powerpoints, making posters, papers, goofing off, etc.), then lead them to the notion of doing math on the computer if they don't immediately realize it. Write down the following equation (or a similar one to your liking):

$$Y = 3X + 10$$

On the whiteboard, draw the following grid:

X	1	2	3	4	5	6	7	8	9
Y									

Have them run through 3 or 4 of them. When they start getting it, ask them if they're bored yet, and play up the fact that math like this is really, really boring. Scientists don't like doing it either. Emphasize that this is the reason why we use computers as scientists - so that they can have something else do the hard work, leaving us to do the interesting stuff. Explain to them we are going to teach them the basics of programming because it is a powerful tool to make doing science easier.

Activity (50 minutes):

Part 1 - Peanut Butter Jelly Time (25 minutes)

Using the materials already set up at the group table, the mentor will assume the role of the 'robot'. With the bread and peanut butter set-up in front of them, the head mentor will listen to commands from the group and 'execute' them. (Note: depending on the personalities in the group, i.e. if one person hogs all the attention, you may have to institute a policy where students take turns giving commands). This requires a bit of improv work, but the goal is to be as literal minded as possible. As they get better at defining things, and also time starts running out, start getting better at understanding them. There are three main ideas to get across in this lesson.

1) Algorithmic thinking: You can only take and execute one command at a time
2) Defining parameters: Everything must be defined properly before it can be used.
3) For loops: Whenever you spread the peanut butter, do not just spread it. Make them move your hand every time. Then introduce For loops and explain that we use them because we are lazy.

If there are enough mentors, one mentor should be explaining, while the other is a robot. It adds to the comedy and memorability of the activity. Common mistakes include:

1) 'Pick up the knife' - Response: what is knife?
2) 'Move left' - simply keep rotating left until they figure out they need to define when to stop

3) ‘Put the peanut butter on the bread’ - put the jar of peanut butter on the bread
4) ‘Rotate the knife in the peanut butter’ - don’t hold on to the peanut butter jar while you rotate, making it an ineffective process

Part 2 - Starburst (25 Minutes)

One of the students will now become the Robot. They will have the job of sorting a packet of Starburst into 4 piles based on color. The difference this time is that the commands will not be spoken, but rather written on the board. Students will write the script, and we will have the robot perform what it says. This task is also improvisational - the mentor should offer help with writing the ‘script’, but the should be allowed to fail. If the ‘script’ would fail, and the robot doesn’t fail accordingly, use your best judgement on whether to correct them. This lesson should reinforce the previous section, with the additional introduction of if-then statements.

- **Intermediate Level:** If they finish both part 1 and 2, then open the laptop and show them the code and how it works. Match the sections to the concepts explained in the lesson, and explain (very roughy) how they work.
- **Advanced Level:** Start playing with the parameters (e.g., number of starbursts, starburst probability)
- **Assessment:** Can they answer the following questions:
    - Can computers process more than one thing at a time?
    - What is a for-loop?
    - What is an if-then statement?
    - We have the following statements written down: X=1; X = 2; Y = 4; X = Y+X; What does X equal?
    - Why do scientists use computers?

Debrief (5 minutes): Go over the assessment questions, and make sure that the students understand the answers. Emphasize the reason that scientists use programming - we want to do math repeatedly, but we’re too lazy/it would take too much time to actually do it.

Schedule (1 Hour 15 Min):

10 minutes **Warm-up Activity**

10 minutes **Lesson Introduction**

50 minutes **Activity**

5 minutes **Debrief**

Notes/Concerns/Issues: Ideas about tricks that can aid understanding, issues students are likely to have, classroom management during the lesson, specific points to emphasize, etc.