LESSON 1: WHAT IS PROGRAMMING?

GETTING STARTED

CURRICULUM STANDARDS
• Know that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text e.g. programmable etc.
• Understand that computers have no intelligence and that computers can do nothing unless a program is executed
• Recognizes that all software executed on digital devices is programmed
• Shows an awareness of tasks best completed by humans or computers

OVERVIEW
In the 21st century, children grow up interacting with technology on a daily basis. By now, they probably already understand how to use a computer, a cellphone, or a television. But do they understand how these technologies work? Probably not. That is where Bomberbot comes into play.

Computers surround us everywhere we go. Most of us use them everyday and many of us are adept at using them. But how do they work? How do they think? How can we make them quicker and more intelligent?

Our computers are only as useful as we make them. By themselves, computers are unintelligent machines and can do nothing until we write instructions for computer programs and execute them. In order to do anything, computers require a program, a sequence of instructions written to perform certain tasks, to function.

All computer programs have been written by humans. As humans, you and your students can learn to program and execute these programs, too.

To order a computer to do something, one must communicate to it using the language of computers. While most programming languages, like JavaScript, Python, and Ruby, are text-based, programs do not necessarily need to be text. A program simply executes the directions it is given, whether it is a word, sound, or in Bomberbot, directions in the form of arrows and other symbols.

Regardless of the type of computer language, each one contains the same basic concepts (sequences, loops, conditionals, variables, algorithms, etc.) we will cover in this course.

The activity begins by giving a very simple introduction to programming, its uses in daily life, and examples of how it surrounds us. However, programming is more than creating the set of instructions for computers to follow. It is the art of blending human ideas and digital tools to increase creativity and
problem-solving power. By the end of the lesson, we hope to empower your students to program and inspire them with the inherent possibilities the technology offers them personally in the future, regardless of their individual life path.

PROGRAMMING IN BOMBERBOT
In Bomberbot, students must create the exact set of directions for Bomberbot to reach his goal of collecting all the stars and smashing all the rubies. In this lesson your students will practice giving instructions to Bomberbot, which will follow the directions exactly as they are given from the start of the computer program to the end. When Bomberbot runs out of instructions (sometimes in the middle of level), Bomberbot will no longer move. If students add instructions to the sequence and hit “Play,” Bomberbot only follows the instructions he is given from where he is currently standing. Remember, computers and Bomberbot do not think for themselves! We must give them the exact set of instructions to follow from beginning to end.

LESSON PLAN
(Estimated Time: 60 minutes)

LESSON OBJECTIVES
- Students recognize that almost all modern digital technology is some form of a computer program
- Students understand that a computer program is a set of instructions for a computer to follow that anyone can write
- Students can create a simple, non-text based program

KEY CONCEPTS
- To program: to create a set of instructions for a computer to follow (in order to complete a task or solve a problem)

MATERIALS & PREP
“What’s Programming?” Presentation Slides
Flappy Bird (preloaded in your browser)

[SLIDE 1] WHAT’S PROGRAMMING?
(Estimated time: 5 minutes)
Do: Ask students to raise their hands if they know what programming is. Take a few volunteers who would like to answer the question, and introduce unit on programming.

Say (suggested): What is programming? Who has heard of programming before?
TEACH NEW PROGRAMMING CONCEPT
(Estimated time: 15 minutes)

[SLIDE 2] WHAT IS YOUR FAVORITE VIDEO GAME?
Do: Ask the students if they ever played video games before, and discuss the rules of the game, emphasizing that all games have instructions. Your students might be very excited to see their favorite video games on the screen.

Say (suggested): We’re going to learn about computer programs today. Before we begin, let’s take a look at some computer programs you might already know. How about video games? What is your favorite video game? What are the rules of the game? How do you get points or move ahead? Have you ever wondered how video games work?

[SLIDE 3] FLAPPY BIRD

Do: Use an example to explain that video games work with a set of rules that defines its behavior. (Flappy Bird, for example). (If time permits, go to www.freeflappybird.org and begin playing or let one student play. Illustrate that rules determine how to win points or lose.) Look at the game together and write down on the board a couple of basic rules of Flappy Bird.

Say (suggested): What are the basic rules that go with Flappy Bird? Let’s write them down together.

These are the rules involved in playing Flappy Bird:
1. If you touch the screen, the bird jumps.
2. If the bird touches a pipe, the game is over.
3. If the bird touches the floor or the ceiling, the game is over.
4. You score a point every time the bird avoids a pipe.

[SLIDE 4] WHAT’S PROGRAMMING?

Do: Define programming.

A program is just a set of instructions or rules that a computer follows. Programming is the way the humans create a set of instructions for a computer to follow. It is how we tell computers what to do.

Say (suggested): So what is Flappy Bird anyway? It’s a game. But it’s also a program.

[SLIDE 5] IT’S JUST ABOUT EVERYWHERE
Say (suggested): Video games and apps are examples of programs in real life. Can you think of any more programs that you use?
LESSON 1: WHAT IS PROGRAMMING?

[SLIDE 6] NOT JUST IN GAMES – IT’S USED ALMOST EVERYWHERE!
Say (suggested): Not only videogames, but also other technologies are programmed. What about airplanes, computers, cars, microwaves! Can you think of any more programs that you use?

[SLIDE 7] PROGRAMMING LANGUAGES
Do: Introduce programming languages.
Say (suggested): Just like with humans, we communicate with machines using specific languages. There are different languages used to communicate with different machines, just like there are different languages (i.e.: French, Spanish, English, et cetera) to communicate with different people. Each of these group of languages are called “programming languages.” They might all look different, but they actually are telling the computer to do the same thing.

Teacher Tip! Some common programming languages include Java, Python, and C. Many popular web applications or “apps” for short are built on Ruby on Rails. iPhone apps are built on C (or Objective C), and Facebook is built on PHP! JavaScript (not to be confused with Java) is a programming language used to make websites interactive.

[SLIDE 8] BOMBERBOT HELPS US UNDERSTAND PROGRAMMING
Say (suggested): We’re going to learn to program by playing a game with a special robot called a Bomberbot. We need to give Bomberbot directions to move around and complete levels. We need to try to do this in as few directions as possible. The first thing we’re going to do is to learn to move our Bomberbot up, down, left, and right to solve a task. Later on, we’ll give Bomberbot more complicated directions to follow as we get better at programming. But first, we’re going to do an activity to see how this works in the computer.

UNPLUGGED ACTIVITY: HUMAN BOMBERBOT #1
(Estimated Time: 10 minutes)

Do: Choose a student to pretend to be a “Human Bomberbot.” Give the Human Bomberbot a goal from his or her current position, such as retrieving a book, turning on the light, or opening the door. (Alternatively, you can also pretend to be a “Human Bomberbot” and follow instructions from your students).

Do: Define what is right, left, forward and backward by drawing the arrows on the board. Make sure that the human Bomberbot begins in the back of the class (that way he or she can see the arrows). Ask students to give directional instructions, one at a time, and ask the human Bomberbot to follow them exactly as directed. For example:

- Walk forward 5 steps.
- Walk to the left 10 steps
LESSON 1: WHAT IS PROGRAMMING?

- Walk backwards 3 steps

When a wrong direction is given and the robot does an incorrect action, stop the exercise (and ask the robot to freeze)! Ask students why this has happened. Explain that the robot only understands instructions it is given. It does not think on its own. Remember:

The instructions are the program. The robot is the computer. And the person calling out the instructions is the programmer.

Do (optional): Repeat this activity with another “robot.” Ask the robot how they knew what to do.

[SLIDE 9] PRACTICE LEVEL
Do: Show the first level on the board and show how Bomberbot works.

Say: The goal of the game is to collect all the stars in the least amount of steps. We do this by giving our robot a set of instructions to follow.

Say and demonstrate: On the left is our program panel, where we can select instructions to create our program. You can drag and drop instructions into the panel with your mouse, or you can click on the instructions. If you make a mistake, you can click the red trash can icon to delete the instructions one at a time. You can also insert instructions between other instructions by dragging the instruction and dropping it in between other instructions.

Bomberbot does not know to follow these instructions until you run your program. When you click Play, Bomberbot follow these instructions always from beginning to end, starting in the top left hand corner. If you see Bomberbot make a mistake, you can click “Pause” to stop the program.

Click “Return” to return Bomberbot back to his original position. This leaves your original program in place, so you can change it and try again. Click “Restart” if you want to restart the level from the beginning. This will return Bomberbot back to his original position and erase your entire program, if you want to start over from scratch.

Ask: What happens if I click Pause to stop the program, add or change instructions and click Play again? (Suggested answer: The robot begins to run the program from where he is currently positioned. This means that he will start with the first instruction in the program again and this will most likely won’t lead to Bomberbot solving the level).

Say: To get 3 stars in the level, you have to create a program in the least number of instructions possible. If you get stuck, you can click on the Hint and Help buttons on the right, where the robot trainer will tell you how to solve the level with 3 stars.
LESSON 1: WHAT IS PROGRAMMING?

[SLIDE 10] PRACTICE: PLAY BOMBERBOT MISSION #1
(Estimated Time: 15 minutes)

Do: Students can log onto Bomberbot with their username and password. They can click “Missions” and start Mission 1. There is an optional video you can also watch in class.

Teachers Note! Students should finish levels 1 – 13 to fully understand the programming concept of this lesson. Levels 14 and on are additional exercises for students who finish early in class or want to play more levels at home.

[SLIDE 11] COMPUTERS ONLY FOLLOW THE INSTRUCTIONS YOU GIVE THEM!
Do: Show the slide on the board and tell students to refer to it when they get stuck on how to play the game.

Do: Once you see that some students have the hammer power, come together as a class again and explain how the hammer works.

Say: If a yellow block is in the way to get to the star, we need to smash it with the hammer. We do this with the green hammer instruction. But if you smash the yellow block, that does not mean that the robot also moves. The robot is still in the same place (you can demonstrate this by pretending you smash a book for instance, students can see that you are in the same place).

Say: We also have a new goal in the levels now. In some levels you will see rubies. You have to smash all the rubies and collect the stars to complete the level.

[SLIDE 12] FEEDBACK
(Estimated Time: 5 a 10 minutes)

Discuss the instructions that Bomberbot understands and the problems students face when programming Bomberbot.

Say (suggested): Remember that Bomberbot does not read words. It reads symbols that stand for visual instructions. What are the instructions that Bomberbot understands?

Say (suggested): What problems did you face when you programmed Bomberbot? How did you solve this problem?

Do: Explain to your students that they programmed their robots in order to solve the levels, similarly to how people created video games by programming. (It’s a bit more complicated than this, because as a
programmer you need to check the rules, repeat orders, list events, etc. but these are skills your students will learn with additional Bomberbot activities.) When Bomberbot moved the wrong direction, it was not because the robot did anything wrong. It was because we gave him the wrong instructions. Computers do not think. They only do what we tell them to do!

**ADDITIONAL EXERCISES**

Students can complete levels they did not finish in class at home.

Ask your students to write a list of instructions to play “Human Bomberbot” with another student (or sibling) to accomplish a goal (such as, opening the door or turning on the light from across the room). Each instruction should contain one movement.