Algorithms and Programming
Module 5 Lessons 5.1-2

Context of the Lesson

The Big Idea: Programming Scratch

Prerequisite Knowledge and Skills: Basic knowledge of routines, patterns Basic knowledge of computers	Connections to SOLs: Computer Science 5.1 Computer Science 5.2

Objectives of the Lesson	Formative Assessment
Learning Targets (I can...): Create an algorithm Create an alternative algorithm that can accomplish the same task Program an algorithm on a computer	Scratch online assignment

Materials

Computer with internet access	

Lesson Structure and Activities

Warm Up *[5-10min]* :Opener:

Ask the students if any of them know what an algorithm is.

- Responses will vary but generally it is fine if absolutely zero students know what an algorithm is. Explain that the class will be learning about algorithms.

Launch (Engage) *[5-10min]* :Teacher Directed Instruction:

Vocabulary:
Algorithm – a step by step pattern or instructions used to solve a problem.
Sequencing – Doing tasks in a set order. A series of tasks laid out in order is a sequence.
Loop – Something that repeats itself.
Conditionals – Things that only happen “on a condition”.

Algorithms are patterns used to solve problems. Think back to when you first learned addition or even how to count. You had to add numbers over and over until you reached the number you wanted. Or when you learned multiplication and had to add the same number *multiple* times. That was a looping algorithm, an algorithm that repeats itself.

People use algorithms in their lives without realizing it, and humans are generally very good at using algorithms to solve problems. There can be more than one valid algorithm just like there are more than one correct way to solve any problem.

Imagine you were under the covers, laying in a bed, and you wanted to get out of the bed. You could get out of bed by pulling the covers off, sitting up, turning, and then moving your lower body off of the bed, then standing up. Or you could roll out of the bed. Or someone could drag you out of bed. All of them have steps that have to be taken, and all of them solve the same problem of getting out of bed.

The same thing applies to math. (Skip this portion if the class is not ready for or not used to long division. In that case, go over a simpler form of math or move on at your own discretion.) When doing long division, one has to **first** get a number out of the dividend bigger than the divisor. **Then** they have to multiply the divisor **until** its the biggest it can be without being bigger than the dividend. **Then** they subtract the numbers, and **finally** they must **repeat** this process **if** it can still be done.

Algorithms generally follow simple rules that can be broken down into simple ideas. Sequences, like doing A, then B, then C. Loops, like doing A until you're then, then doing B. And Conditionals, like doing B only if something happened to A. We use certain words in our language to express these ideas informally. *If, only, when, until, while, as needed, then, for as long as*.

Computers use these same ideas to do everything. If you can't tell how to do a conditional, you can't say *if someone clicks on the web browser, then open the web browser*. If you can't tell how to do a loop you can't say *keep loading the video until you're done*. If you can't tell how to do a sequence, you can't even begin to describe the process of turning on a computer. What happens after you press the power button? What happens after you open a refrigerator door? What happens after you do the first step of a math problem. The next step in the sequence.

Remember that there is more than one valid algorithm for every problem. There is a game called H-O-R-S-E. In this game, multiple competitors try to accomplish a task in a way that is so difficult to replicate, that their opponents can't copy them. They have to actually accomplish the task, and usually start from the same position, but the in between parts are all up to the skill and creativity of the player. There are numerous variations on this game, but the idea remains at the core. Can you create an algorithm that does the exact same thing in a different way? The winner is the person most capable of producing a result that no one else can copy enough times, while copying others' results. While H-O-R-S-E is usually used for tasks like throwing a basketball into a hoop, people all over the world use the same concept to figure out what is the best way to do the same thing a group of people are already doing. (The teacher may also tie this into innovation. Trying to improve upon previous work is in the very core of product innovation.)

Explore *[25min]* :Joint/Guided Practice | Student Practice:

To explore this concept, the students should use a computer to perform exercises in creating step by step looping instructions using a block language, or barring that, watch a demonstration of such. The teacher should be provided with the tools and ability to complete the assignment themselves, as well as copies of the correct answers and basic knowledge of the application to assist students who have a hard time completing the assignment.

Summarize *[15min]* :Debrief :

Ask students how comfortable they are with the idea of algorithms, sequences, loops, and conditionals.

Extensions:	