

Robots and Sequencing: 1st Grade

Lesson 2: Debugging



This research is supported by Award #1542465:
RET Site: Cyber Security Initiative for Nevada Teachers (CSINT)

Robot: A robot can be defined as a mechanical device that is capable of performing a variety of tasks on command or according to instructions programmed in advance.

Subject Area: Computer Science and Security/Math/English/Engineering

Grade-Level: 1st grade

Lesson Title: Sequencing, Debugging, and Cryptography

Introduction: In this lesson students will be exposed to coding through a progression of skills. At the end of the unit, students will be able to understand sequencing, trouble-shooting, and using an algorithm to solve a problem. The unit will be used to help students make connections to Math, English Language Arts, and Engineering through the real world application of computer science learning. The main focus of this lesson is to start the scaffolding process of learning so that students will become educated digital citizens. With such a strong focus on Math and English Language Arts for many teachers, computer science has not been a focus in the elementary education setting. This lesson was developed with the idea that Common Core State Standards for Math and English Language Arts can be taught through a curriculum that many teachers feel that they do not have time to teach.

Lesson Overview: The purpose of this lesson is to introduce young learners to coding and computer security through an “unplugged” environment. At the end of this unit, students will be able to understand the basics of sequencing, coding, debugging, and cryptography. The first lesson starts with sequencing and transitions into coding and debugging in the second lesson. Finally, the third lesson uses the understanding of algorithms to introduce students to encryption through cryptography.

Lesson Learning Objectives:

- **Understand that sequencing is the order of events**
- **Using ordinal numbers to list steps is important in developing sequence**

Standards:

1) Computer Science

- CPP.L1:3-04 - Construct a set of statements to be acted out to accomplish a simple task.
- CT.L1:6-01 - Understand and use the basic steps in algorithmic problem-solving.
- CT.L1:6-02 - Develop a simple understanding of an algorithm using computer-free exercises.
- CT.L2-03 - Define an algorithm as a sequence of instructions that can be processed by a computer.
- CT.L2-06 - Describe and analyze a sequence of instructions being followed.

2) ELA:

- CCSS.ELA-LITERACY.RI.1.10: With prompting and support, read informational texts appropriately complex for grade 1.
- CCSS.ELA-LITERACY.RF.1.1: Demonstrate understanding of the organization and basic features of print.
- [CCSS.ELA-LITERACY.RF.1.3](#): Know and apply grade-level phonics and word analysis skills in decoding words.
- CCSS.ELA-LITERACY.W.1.8: With guidance and support from adults, recall information from experiences or gather information from provided sources to answer a question.
- [CCSS.ELA-LITERACY.SL.1.1](#): Participate in collaborative conversations with diverse partners about *grade 1 topics and texts* with peers and adults in small and larger groups.
- [CCSS.ELA-LITERACY.SL.1.1.A](#): Follow agreed-upon rules for discussions (e.g., listening to others with care, speaking one at a time about the topics and texts under discussion).
- [CCSS.ELA-LITERACY.SL.1.1.B](#): Build on others' talk in conversations by responding to the comments of others through multiple exchanges.
- [CCSS.ELA-LITERACY.SL.1.1.C](#): Ask questions to clear up any confusion about the topics and texts under discussion.
- CCSS.ELA-LITERACY.SL.1.2: Ask and answer questions about key details in a text read aloud or information presented orally or through other media.
- CCSS.ELA-LITERACY.SL.1.3: Ask and answer questions about what a speaker says in order to gather additional information or clarify something that is not understood.
- CCSS.ELA-LITERACY.L.1.1.A: Print all upper- and lowercase letters.
- CCSS.ELA-LITERACY.L.1.2.B: Use end punctuation for sentences.
- CCSS.ELA-LITERACY.L.1.2.E: Spell untaught words phonetically, drawing on phonemic awareness and spelling conventions.
- CCSS.ELA-LITERACY.L.1.6: Use words and phrases acquired through conversations, reading and being read to, and responding to texts, including using frequently occurring conjunctions to signal simple relationships (e.g., *because*).

NGSS:

-Science and Engineering Practices

- **K-2-ETS1-1:** Asking questions and defining problems in K-2 builds on prior experiences and progresses to simple descriptive questions.
 - Ask questions based on observations to find more information about the natural and/or designed world
 - Define a simple problem that can be solved through the development of a new or improved object or tool

-Disciplinary Core Ideas

- **ET S1.A:** Defining and Delimiting Engineering Problems
 - A situation that people want to change or create can be approached as a problem to be solved through engineering.
 - Asking questions, making observations, and gathering information are helpful in thinking about problems.
 - Before beginning to design a solution, it is important to clearly understand the problem.

Engage: "Debugging"

- *In this activity students will be shown a program that a robot will run but the program will not allow the robot to reach its goal. This activity should be done using the same grid that was used in the evaluation section of lesson 1. The teacher must write a simple program that the robot will follow to reach a goal. One of the steps in the program must make the robot go in an undesired direction which will hinder it from reaching its goal. The first time through, only change the final step in the program. This will allow students some success in solving the problem. Ask the students to discuss with their shoulder partner how to fix the problem. Once they have explained their thinking, create another program that has an incorrect step somewhere in the middle of the program. This will cause the robot to miss its mark by a much greater distance.*

Explore: "Correct and Create"

- *In this section of the lesson, students will be fixing the incorrect program that was used in the engagement section. They will be broken into groups no larger than three to fix the program. Once students have fixed the program, make sure that they test the solution they created to ensure its accuracy. If they have fixed the program effectively, have the students fix or rewrite the program in two more ways. The students can use more lines of code, or less, to reach the goal.*

Student Instructions:

- *You have been given a program that will be used to help Mr. Robot reach its goal. The problem is that the program does not work. Mr. Robot has missed his goal. You will need to work in teams to fix the problem(s) with the code. Just like in life, there are many ways to fix something. You will need to come up with at least one solution to the problem. If you are successful in fixing the program, you and your team will need to fix the problem in a different way. The goal is to come up with three different solutions to the problem. Good luck!!*

Explain:

Vocabulary:

- 1) Code:** program instructions
- 2) Debug:** identify and remove errors from a code/program
- 3) Identify:** indicate who or what
- 4) Test:** check the quality of something

Debugging: once the students have completed fixing the program, have them come up in front of the class and explain their solutions. Using a white board or overhead camera with a grid will help in the explanation process. Once all of the students have presented their thinking, discuss why some of the groups came up with different solutions to the same problem (sequencing, efficiency). Make sure students understand that creating a viable solution can be accomplished both efficiently and inefficiently. Still, as long as the solution is derived, then they are successful.

Vocabulary Review: Now that the students have gone through the process of debugging, we will introduce new vocabulary that will help to deepen their understanding of coding. Create a vocabulary poster (or continue on the poster from lesson 1) for the lesson that the students will be able to reference throughout the entire process. When giving the definitions to the words, make sure to discuss possible synonyms and other definitions that the students will generate. Have the students write the vocabulary in their Technology Journal in order to maintain a reference and work on writing skills. (Make sure to use all new vocabulary

as much as possible during your discussions). The ability level of students may vary and so creating traceable letters may be necessary for a given class or group of students. The vocabulary words are multisyllabic but still can be broken down into more manageable and distinguishable parts.

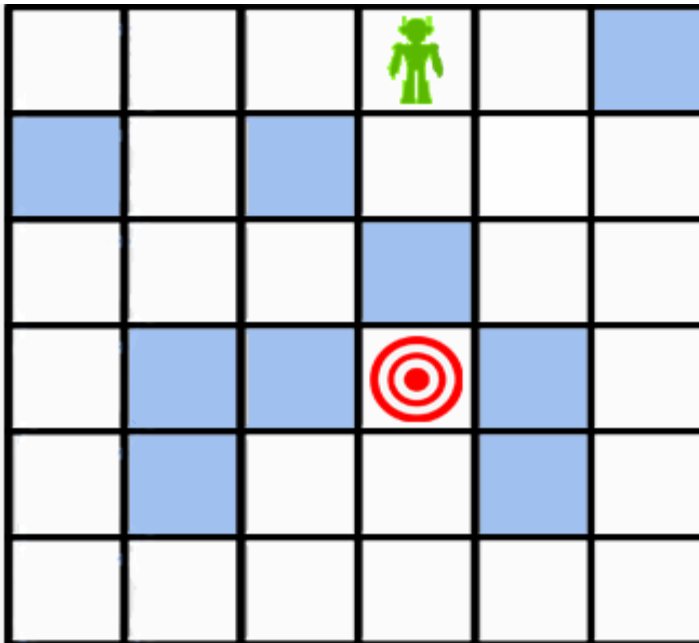
Elaborate:

- *In this activity students will, with the assistance of adults and peers, write a program for a robot to complete a task. The code that they write can be as long or as short as they want it to be. The students should be broken up into groups no larger than three. Each student should be encouraged to write their own code for the robot. If there are students that are unable to write their own code, they can work with a peer to complete the task. Once the students have completed their code and tested it, they will need to swap codes with the members of their group and have them test the code for accuracy as well. Once all codes are bug-free, the group will need to decide on the best code and then change one of the lines of code to make the code not function correctly. They will exchange this non-functional code with another group. Once the groups have exchanged their codes, they will work as a team to debug the code so that it works effectively in helping the robot achieve its goal. This activity can be reiterated in order to develop mastery of content. It can be increased in difficulty by having teams change two or three lines of code or simplified by having teams draw a line for the desired path and then have the other team debug the code so that it follows the line drawn on the grid.*

Student Directions:

- **Mr. Robot needs to get to the bullseye. We are going to write a code that will allow him to navigate to it. There are many different paths that he can take so it is up to you to choose the best path. There are only four commands that we can use. They are: move up (↑), move down (↓), move right (→), move**

left (←). Each step will be written as a line of code. We will continue writing commands until Mr. Robot reaches the bullseye. (Additional challenge: if you want to test your writing skills, you can use the words along with the symbols)



Evaluate:

- *In this activity students will be debugging code on their own. The assessment given will test to see if students are able to use the skills that have developed in a novel situation. The assessment uses all of the commands that the students are used to but adds a fifth command of coloring in. The students will check the code against the map to make sure they can correctly complete the desired task. If the assessment is too difficult for some of the students to debug, you can have the students write their code in the boxes and then correlate them to the commands that are already given. If you feel that the students would benefit from doing one together, please do!*

Student directions:

- *You are now an expert at making Mr. Robot move to where he needs to go. Now it's time to test your critical thinking and problem solving skills. You will be debugging code that was written by other people. This time you will be using the code to draw a map. The same commands of up, down, left, and right will be used, but now we will be*

adding a fifth command; fill in. You will use these commands to create the map that is presented for that question. Have fun and good luck!!