|  |  |  |
| --- | --- | --- |
| **Name:** John McClellan | **Contact Info:** jmcclellan@readingschools.org | **Date:**11/01/17 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Lesson Title :** How Computers Think | **Unit #:**1 | **Lesson #:**1 | **Activity #:**2 |
| **Activity Title:** Introduction to Computer Functions |

|  |  |
| --- | --- |
| **Estimated Lesson Duration:** | 4 Days |
| **Estimated Activity Duration:** | 3 Days |

|  |  |
| --- | --- |
| **Setting:** | Classroom |

|  |
| --- |
| **Activity Objectives: The students will be able to:** |

1. Read and construct pseudo-code
2. Map out a logical flow chart
3. Describe an interpret loops.

|  |
| --- |
| **Activity Guiding Questions:** |

* Why does a computer use Boolean logic?
* What is a loop?
* What is the purpose of constructing pseudo-code/flow charts?
* How are computer variables different from algebraic variables?

| **Next Generation Science Standards (NGSS)**  |
| --- |
| **Science and Engineering Practices (Check all that apply)**  | **Crosscutting Concepts (Check all that apply)** |
| ☒ Asking questions (for science) and defining problems (for engineering) | ☐ Patterns |
| ☒ Developing and using models | ☒ Cause and effect |
| ☒ Planning and carrying out investigations | ☐ Scale, proportion, and quantity |
| ☒ Analyzing and interpreting data | ☐ Systems and system models |
| ☒ Using mathematics and computational thinking | ☐ Energy and matter: Flows, cycles, and conservation |
| ☐ Constructing explanations (for science) and designing solutions (for engineering) | ☐ Structure and function.  |
| ☒ Engaging in argument from evidence | ☐ Stability and change.  |
| ☒ Obtaining, evaluating, and communicating information  |  |

| **Ohio’s Learning Standards for Science (OLS)** |
| --- |
| **Expectations for Learning - Cognitive Demands (Check all that apply)** |
| ☒ Designing Technological/Engineering Solutions Using Science concepts **(T)** |
| ☐ Demonstrating Science Knowledge **(D)** |
| ☒ Interpreting and Communicating Science Concepts **(C)** |
| ☐ Recalling Accurate Science **(R)** |

| **Ohio’s Learning Standards for Math (OLS) and/or** **Common Core State Standards -- Mathematics (CCSS)** |
| --- |
| **Standards for Mathematical Practice (Check all that apply)** |
| ☒ Make sense of problems and persevere in solving them | ☐ Useappropriate tools strategically |
| ☒ Reason abstractly and quantitatively | ☐ Attendto precision |
| ☒ Construct viable arguments and critique the reasoning of others | ☒ Look for and make use of structure |
| ☐ Model with mathematics | ☐ Look for and express regularity in repeated reasoning |

|  |
| --- |
| **Unit Academic Standards (NGSS, OLS and/or CCSS):** |

Standard for Mathematical Practice #2 Reason Abstractly and Quantatively; Standard for Mathematical Practice #5 Use Appropriate Tools Strategically; CCSS High School Modeling Domain: *“When making mathematical models, technology is valuable for varying assumptions, exploring consequences, and comparing predictions with data.”*

|  |
| --- |
| **Materials**: (Link Handouts, Power Points, Resources, Websites, Supplies) |

* Pre-test
* Reading Activity Walkthrough
* Variable Activity Walkthrough
* Choose your own adventure book or online game
* Notes for game/book worksheet
* Mind reading binary worksheet
* Quiz

|  |
| --- |
| **Teacher Advance Preparation:** |

* On Google Docs, make note of who wrote the questions for their homework assignment. Perhaps even mark as a grade.
* Prepare some pseudo-code regarding variables and basic operations.
* When choosing the 2nd person book or game, run it through to see the time it might take. Ask whether or not you want to make it team or individually based. It might be easier if you can get a couple copies of each book/game.
* Students struggle with the idea of Boolean logic and Loops. Really try to use the formative assessments to gauge understanding.

|  |
| --- |
| **Activity Procedures:** |

Day 1 (Ordering and Variables):

1. Begin by having the students take the pre-test. Assure them that you do not expect them to know a whole lot, but to write down everything that they know about each question. This shouldn’t take more than 10 minutes.
2. Bring up their guiding questions and review through them all. Compile them onto one big document in front of them. Ask if there are any more to add after taking the pre-test. Remind them of the challenge and what they are working towards. This should only take a couple minutes.
3. Hand out mapping cards. This should be prearranged for specific students based on the map of your room. Have them run through it—the activity should take about 2 minutes
4. Discuss the consequences of “following the recipe” vs. “jumping to the end.” Try to lead them down the idea of syntax. This should take 5-10 minutes
5. Do the variable activity. This shouldn’t take longer than 10 minutes.
6. Clarify meaning in broader scope, using syntax. Bring up pseudo-code and see if the students can start to read it. Go through a number of examples.
7. Exit slip when there is about 5 minutes left.

Day 2 (Boolean logic):

1. Hand out the books to each group. Explain quickly how second-person books work. Have the students work together to try and build a flow chart. Let them know it will be very difficult to design a full web in the time, but go as far as they can in 15 minutes. Two people should be reading and two should be writing/summarizing the material.
2. After receiving their flow charts, do the Mind-reading binary activity. This shouldn’t take more than 10 minutes.
3. Ask what these two have in common – two choices.
4. Write up flow chart, excel if-statements, and pseudo-code to compare—10 minutes
5. More examples, if time allows
6. Exit slip when there is about 5 minutes left

Day 3 (Loops)

1. Have prebell on the board be “write ‘I will understand loops’ 15 times.” Afterwards, see if any students have an idea as to what a loop is. Lead a discussion as to the importance of computers doing the same task over and over again for us. This should not last longer than 10 minutes.
2. Take the time to show the pseudo code of a “for” loop. Give a couple of examples (any brute force examples are good). 10-15 minutes.
3. Have students come up with their own example of when a for loop would be used. Write ideas on board and then see if the class can build pseudo code for the tasks outlined. 5 minutes of coming up with their own idea, 10-15 minutes of discussion and construction.
4. See if there are any topics that need to be reviewed before the quiz. If there is enough time, provide quiz to students. If not, assure students the quiz will take place next class period, first thing.

**Formative Assessments:** Link the items in the Activities that will be used as formative assessments.

Worksheets, discussion, team activities.

**Summative Assessments:** These are optional; there may be summative assessments at the end of a set of Activities or only at the end of the entire Unit.

If there isn’t enough time on day 3 of this lesson for our quiz for the week, it will be given first thing the next day, before we begin the second lesson.

|  |
| --- |
| **Differentiation:** Describe how you modified parts of the Lesson to support the needs of different learners.Refer to Activity Template for details. |

Students began to naturally sort themselves into groups that had varying levels of expertise and experience in code. Each group had a member who was somewhat familiar with the process. Most of this lesson used whole class instruction, and, keeping in mind the student with a 504, I was able to get most students to participate.

|  |
| --- |
| **Reflection:** Reflect upon the successes and shortcomings of the lesson. |

Students had fun with the mapping game – it led to a great discussion of how conversations are had between humans and how difficult it is to speak to a computer. The transition from that to talking about syntax and pseudo-code worked really well. That variable activity started well and then the pseudo-code that we started writing intimidated them. There was still an understanding gap that I will have to reflect on and adjust as the years go by as to how to describe what variables behave like in coding.

The day 2’s flow charts went well – students really understood the tie in between right/wrong, yes/no, true/false, 1/0. Students took a real interest in the binary number system, even though there wasn’t a whole lot of need for it for their activities or challenges. The design of the second person novel went poorly because the links were blocked on their chromebooks. We instead had to do it as a class, which lost some of it’s magic. Next time, I would suggest teachers print out the books or use library books, just in case.

Day 3 was compressed into the next lesson’s activities because we lost a bell of school because of picture retakes. Students were not able to discuss and develop their own pseudo code as much as I would have liked, but it still was a good in-class discussion on it. They were once again amused by the pre-bell.