|  |  |
| --- | --- |
| ***Overview*** | |
| *Topic/Theme*  Please list a brief title for the task | Programs and Algorithms |
| *Lesson/Activity Goal* | Students will be able to understand the importance of step-by-step directions to accomplish a task. |
| *Rationale and Unit Placement*  Please provide a few sentences that describe how this lesson or activity might fit within an existing unit. | This lesson aligns with basic math problems and how to work on a math problem step by step, in a specific order. This lesson also aligns with classroom management and day-to-day routines: following directions. |

|  |
| --- |
| ***Standards Alignment***  Please list the standards aligned with this task (e.g. K-2PA.2) |
| K-2.PA.3 Arrange information using concept mapping tools and a set of statements that accomplish a simple task.  K-2.NC.1 Use technology to work cooperatively and collaboratively with peers, teachers, and others. |

|  |  |
| --- | --- |
| ***Attributions*** | |
| *Created by:* | Caitlin Tucker and Alex Stopczynski |
| *Date:* | 5-23-19 |

---------------------------------------------------------------------------------------------------------------------

|  |
| --- |
| ***I. Introduction/Anticipatory Set***  How might you make connections to students’ own experiences/ideas or other content to set the stage for the lesson/activity? |
| One way to activate this lesson would be to show the students a video of exact instructions for making a peanut butter and jelly sandwich (YouTube: father with children.) Ask them what they notice, what problems did they have, were they successful, what determines a successful peanut butter and jelly sandwich? What skills would computer scientists need to create something like this? |

|  |
| --- |
| ***II. Summary Description***  Please describe the procedures or parts of the lesson/activity. If you are using an existing activity, you can include a link to the instructions. Feel free to provide any further instructions for how this lesson/activity might be adapted for be integrated with other parts of the unit. |
| Day 1  Task 1: (Unplugged) Directions for Making a Peanut Butter Jelly Sandwich  If necessary, teacher demonstrates how to make a peanut butter and jelly sandwich with no directions to give students a model for their task. Ask students to write down directions with a partner to describe these actions. Have partners get into groups of four to compare directions, looking for similarities and differences in their code. **This can be done on computers and students can “share” their directions with the class/teacher** to incorporate technology.  Students then take turns reading the code to the teacher, while the teacher becomes the “computer” and acts out the directions. Teacher uses phrases like, “does not compute” or “repeat please.”  Class discussion:  Why are specific directions important for a task?  How are computers different than humans?  Day 2  Task 2: (Unplugged) Describe to Partner How to Draw a House  Review discussion questions from previous day. Put students into pairs. Appoint students as A or B. Have partners B leave the classroom (wait in hallway if possible.) Describe the directions to partner A’s. Give directions to partner B on how to draw a house. Remind them that they can’t use the word house or any other terms associate with a house: window, door, roof and can only give one step at a time.  Call partners B back into the classroom. Complete activity. Have partners B guess what they’re drawing.  Switch roles and have students describe how to draw a boat.  Discussion questions:  What did you find most difficult? Why was this difficult?  What would you do different next time?  How is this activity compared to being a computer?  Day 3  Task 3: (Unplugged) Code blocks  Vocabulary to introduce: algorithm, sequence, programs, debug, event triggers. Connect these terms to the previous activities.  Algorithm/Sequencing: directions for peanut butter and jelly/drawings  Program: working through the sequence to get to the final product  Debug: Fixing code after you have run the sequence  Event triggers: use this term during activity with code blocks  Teacher needs code blocks printed that match code.org blocks. Laminate so teacher can write events on the blocks.  “Directions for Going to Lunch”  Whole class: teacher puts first block on the board: event trigger.  Teacher asks students: When we get ready to go to lunch, what is the first thing we do or say that lets us know it is time to start our routine?  Students answer, teacher writes on the code block. Teacher tells this is an event trigger that every code needs to get started.  Teacher continues to add code blocks while students give her the steps for the routine.  Students then “perform” the code after teacher has given the event trigger.  Class Discussion:  What is an event trigger?  How is this an example of a sequence/algorithm?  What other daily activities/routines could we do this for?  Day 4 & 5  Tasks 4 & 5: (Unplugged/Plugged) Students can get their own set of code blocks to create a sequence for another classroom routine. Then the teacher could introduce them to code.org where they would be given a task to complete. Example: make character move. |

|  |
| --- |
| ***III. Whole Group Discussion Central Questions***  What 1-2 central questions might be used with the whole class to solidify the main idea of the lesson/activity? |
| Why are sequences important? |
| What do you need to know before you complete a task? |
| What determines a successful sequence? |

---------------------------------------------------------------------------------------------------------------------

|  |  |
| --- | --- |
| ***Evidence of Proficiency***  Please list 3-5 categories and brief descriptions of what proficiency looks like for those categories (this can serve as the precursor to a rubric). You do not have to use all of the rows. | |
| Category | Description |
| 1. Sequencing | Students can create a successful step-by-step directions both verbally and written. Students can identify and correct mistakes in their program/sequence. |
| 2. Collaboration | Students work together to create a program, sharing the responsibility of thinking and coding. |
| 3. Discussion | Students were able to answer class discussion questions. |