

**Lesson created by the GMU-ODU CSforAll Team. For more information about**

**this lesson and our CSforAll initiative, contact Dr. Amy Hutchison at** [achutchison1@ua.edu](mailto:achutchison1@ua.edu)

| **Lesson 4: Algorithms** | | **Grade Level: 2** |
| --- | --- | --- |
|  |  |  |
| **Concept: Algorithms** | | |
| **Vocab:**   * Algorithms | | |
| **Summary:** In this lesson, students will sequence a story with a beginning, middle, and end, and write an algorithm for how the wolf got from house to house in the story of the “Three Little Pigs.” | | |
| **Lesson Objectives (learning targets): I can…**   * Define and give examples of “algorithms” * Write an algorithm * Debug an algorithm | | |
| **VDOE English Standard(s)** | **Computer Science Standard(s)** | |
| **Communication and Multimodal Literacies:**  2.2 The student will demonstrate an understanding of oral language structure.  a) Listen and respond to a variety of text and media.  2.8. Students will read and demonstrate comprehension of fictional texts. | 2.1 The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively a. using sequencing;  2.2 The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively a. using sequencing;  2.3 The student will analyze, correct, and improve (debug) an algorithm that includes sequencing and simple loops, with or without a computing device. | |

| **Materials** |
| --- |
| * Teacher slide deck (see website) * Student [slide deck](https://www.dropbox.com/scl/fi/wscjc9jrtgg5oka3xdb74/Student-slides_L4G2.gslides?dl=0&rlkey=yvwkzw73yutydjkocow54iufi#slide=id.g1497e9f86c5_0_141) * Video read aloud of “The Three Little Pigs”: <https://www.youtube.com/watch?v=FNYBQsay_Ek> |

| **Lesson Structure and Activities** |
| --- |
| **Warm-up/Bell Ringer Activity: (10 min)**  **NOTE: All slides for this lesson are scripted so that, if needed, you can see exact definitions and instructions for teaching this lesson in the notes at the bottom of the teacher slide deck.**   * Review key academic vocabulary and link concepts (slides 6-12)   + Computer science   + Sequencing   + Debug   + Abstraction: use the *most important* ideas to help make information easier to understand   + Decomposition: break problems apart into smaller pieces, to figure out how to solve them |
| **Direct Instruction (5-8 min)**   * Introduce new vocabulary word: “algorithm”   + Ex: The green start flag, motion blocks, and end block (slide 14) * Show “Algorithm” by BBC Learning<https://www.youtube.com/watch?v=Da5TOXCwLSg> (slide 15) * Questions to keep in mind as you watch   + What computer science words do you hear?   + Do you hear words we have talked about when we have been learning about computer science?   + Do you hear new words? |
| **Guided Instruction: (5-10 min)**   * Repeat definition of algorithm and give examples outside of computer science (slides 16-21) * Turn & talk: in pairs, direct students to discuss how they would write an algorithm for “getting ready for school” |
| **Independent Practice: (20-30 min)**  **Big Bad Wolf’s Algorithm (slides 23-30)**   * Play the 5-minute read aloud of “The Three Little Pigs” * Ask if students have any questions * Instruct students to write an algorithm to guide the Big Bad Wolf to visit each of the three little pigs’ houses following the sequence in the story.   + Note: students may choose to use either words or coding blocks to write your algorithm.   + BONUS: can you find a way to abstract any steps in your algorithm to make it shorter and more efficient? * To make this an unplugged coding practice, share individual copies of the [student slide deck](https://www.dropbox.com/scl/fi/wscjc9jrtgg5oka3xdb74/Student-slides_L4G2.gslides?dl=0&rlkey=yvwkzw73yutydjkocow54iufi#slide=id.g1497e9f86c5_0_141) with each student. Instruct them to drag and drop the code blocks where they belong on the map. There are enough code blocks on the slide to complete the algorithm without copying and pasting new blocks. * (Optional) When done, instruct students to share their algorithm with a partner. See if the partner can spot any errors and debug the code. * Share the solution (slide 29) * Ask for volunteers to share their algorithm (slide 30) |
| **Wrap up: (10 min)**   * Review algorithms (Slide 31-32)   + Ask students to explain what **algorithms** are to a friend or family member *and* think about another **algorithm** you may want to try. * Encourage students that they can become computer scientists with the skills they’ve learned (slide 33) * Show a video of a real-life computer scientist (slide 34) |
| **Assessment Strategy:** Evaluate students’ algorithms with a teacher-made rubric or focusing on a target skill based on the student’s area of need. |