**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)

|  |  |  |
| --- | --- | --- |
| **Unit 4 Lesson 1: Decomposition** *5th &6th Grade* | | |
|  |  |  |
| **Concept: Decomposition** | | |
| **Vocabulary:**  • Patterns, sequence, algorithm, abstraction (review)  • Decomposition  • Summaries  • Events and Loops | | |
| **Narrative/Summary:**  In this lesson, students will use a summary strategy to decompose a story in Coco and use events and loops in Scratch. | | |
| **Lesson Objectives (learning targets): I can…**   * Review CS Vocabulary * Identify features in Coco Level 5 Identify other types of writing (SUMMARIES) that could be used in Coco Level 5 * Identify and use Start blocks and “If/Then” block * Brainstorm new writing ideas with a partner | | |
| **VDOE ELA Standard(s)** | **VDOE Computer Science Standard(s)** | |
| The student will write in a variety of forms to include narrative, descriptive, opinion, and expository.  a) Engage in writing as a process.  b) Identify audience and purpose.  c) Use a variety of prewriting strategies.  d) Use organizational strategies to structure  writing according to type.  g) Use transition words to vary sentence  structure. | The student will break down (decompose) a larger problem into smaller sub-problems, independently or collaboratively. | |

|  |
| --- |
| **Materials** |
| **Lesson materials:**   * Chromebook/Laptop * Internet Access * Read aloud of [Little Red Riding Hood](https://www.youtube.com/watch?v=r_akuIFFpws) or another [story of your choice](https://www.dropbox.com/scl/fi/i0q9o2dm8e0w34yfmutst/simple-stories-for-summarizing.doc?dl=0&rlkey=rv8g9vnaisur6msqb7v3j0a7y) to summarize using SWBST * Teacher slide deck * [Coco Link](https://wego.gmu.edu/scratchgo/login.php) * [Scratch link](https://scratch.mit.edu/) * SWBST [Practice Graphic Organizer](https://www.dropbox.com/scl/fi/jm43x1gqg71rq204yk4cq/SWBST_Practice-graphic-organizer.docx?dl=0&rlkey=bni0elmi28e8ikw8hwfovryzr)   **Supplemental resources:** |

|  |
| --- |
| **Lesson Structure and Activities** |
| **Note for Teachers:**   * **Prior to beginning this Unit,** be sure to assign your students a story in CoCo, using **Level 5.** * **Please use the following naming strategy for assigning the story in CoCo:**   + “Unit # + Descriptor”, for example, “Unit 4 Summary” * **Students should use the same naming strategy for their final Scratch Project:**    + “Student Name + Unit # + Descriptor”, for example, “Johnny Unit 4 Summary” |
| **Warm-up/ Introduction:**  **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.**   * (Optional) Read aloud the summary and standard, the materials and resources needed for this lesson, and the lesson objectives (slides 1-4). * Review Computational Thinking vocabulary by having students free-write for 5 minutes about one CT skill that helps them think like a computer scientist (slide 5). Introduce the term “decomposition”, ask students to share what they think decomposition means. (slides 6-7) * Introduce definition → (slides 8-9)   + Decompose (v): to break a problem down into smaller pieces   + Decomposition (n): breaking a large problem into smaller parts * Discuss as a class additional examples of decomposition in computer science and the real world (See examples below and in slides 9-10)   + Cleaning your room   + In nature: decomposers   + Solving Math problems   + Breaking down a calendar year into months and months into days   + Group dances–whether we do one we all know or make up our own! (like the macarena or maybe one from social media)   + **In CS: Breaking down a complex problem into smaller, more manageable parts** |
| **Direct Instruction & Guided Practice:**   * Explain that in literacy, we use decomposition quite often: (slide 11)   + When we sound words out (ex. mis-com-mu-ni-cate)   + When we edit and fix mistakes   + When we create a set of instructions   + When we share the main idea   + **When we create a summary of a story** * Ask students to share what a **summary** is. (slides 12-13)   + Define→ A summary restates the main ideas of a story in your own words.   + Explain that when we are summarizing, we are decomposing a story into the main parts. * Introduce summarizing strategy: Somebody Wanted But So Then (SWBST) (slide 14)   + Somebody: Character   + Wanted: Goal/motivation   + But: Conflict/problem   + So: Resolution   + Then: How does it end? * **Using a story or text of your choice, guide and model how to use SWBST with the class** OR (Optional) Stop and listen/[watch](https://www.youtube.com/watch?v=r_akuIFFpws) the story of Little Red Riding Hood. (slide 14) * Model how to use level 5 of CoCo by using it to write a summary of the story you choose (or Little Red Riding Hood) using the SWBST strategy. You may also wish to use [this video model](https://www.dropbox.com/s/s1epeuo6dyxmp5n/SWBST%202%20min..webm?dl=0) (slides 15-16) * Instruct students to get with a partner to practice summarizing a familiar story or one that the teacher assigns. They may have a story or text they have been reading or the teacher can provide one. (slide 17)   + They may use the [paper graphic organizer attached here](https://www.dropbox.com/scl/fi/jm43x1gqg71rq204yk4cq/SWBST_Practice-graphic-organizer.docx?dl=0&rlkey=bni0elmi28e8ikw8hwfovryzr) |
| **Direct Instruction & Independent Practice in Scratch:**   * Re-introduce Events: (slides 18-21)   + Events in computer science are the triggers for making action happen, like selecting the play button on any screen. Events in Scratch are represented by the yellow codes including: when flag clicked, when sprite clicked, when key pressed     1. Model New Event blocks (when sprite clicked, when key pressed)     2. Instruct students to try each out on their own in Scratch.        1. Instruct students to create an animation of a sport or outside activity they like to do with two Sprites. Use one of the new event blocks for each Sprite. Share with a partner! * Discuss conditionals: In computer science, a condition is something that must be true in order for something to happen. A condition is said to "evaluate to true" or "evaluate to false." In Scratch, any block whose label says "if," "when," or "until" is a sort of conditional construct. (slide 22) * Introduce If/Then Block: (slides 23-24)   + The if () then block is a control block in Scratch. If the condition written inside the block is true, the blocks inside it will run. If the condition is false, the blocks inside the block will be ignored. The condition is only checked once; if the condition turns to false while the script inside the block is running, it will keep running until it has finished. * Model how to use the If/Then Block (slide 24)   + Video * Instruct students to log into Scratch and try to write an algorithm that uses the If/Then Block (slide 25)   + Create an animation using the “If/Then” block. Have your sprite ask question. Using “If/Then” change the backdrop to green if the answer was correct. Once you are done coding, have a partner try your animation and answer the question! |
| **Wrap up:**   * Have students get with a partner or small group and discuss when they may want to use a If/Then block in their coding. (slide 26) |
| **Assessment Strategy:** Evaluate students’ written story with a teacher-made rubric or focusing on a target skill based on the student’s area of need. |