Category	Instruction Delivery	Instructional Steps
Texas Essential Skills and Knowledge	Analyze debugging techniques and apply them to an algorithm.	**Engage** The teacher begins by displaying a simple everyday algorithm students recognize (for example, steps for making a peanut butter sandwich or logging into a device) that intentionally contains mistakes or missing steps. Students attempt to follow the flawed algorithm mentally and quickly realize it does not work. The teacher asks guiding questions such as, “Where did the process break down?” and “How could we systematically find the problem?” This creates curiosity and establishes that errors are expected and solvable. **Explore** Students work collaboratively in pairs to examine a short, intentionally flawed pseudocode algorithm (for example, calculating the average of three numbers but missing a step or using incorrect logic). Without direct instruction on methods yet, students attempt to identify what is wrong by testing the algorithm with sample inputs. They record observations about where the output fails. The teacher circulates, prompting students to try different inputs and to track each
Performance Measure & Assessment	Students debug a flawed pseudocode algorithm and explain corrections.	
Learning Objectives	Students will analyze multiple debugging techniques (such as tracing, isolating errors, using test cases, and peer review) and apply those techniques to identify and correct errors in a flawed algorithm, demonstrating understanding by producing a corrected algorithm and explaining the debugging process used.	
Objective Statement	“Today you will analyze different debugging techniques and apply them to fix an algorithm, so that you can understand how programmers find and correct errors in code.”	
Purpose of Learning	“Debugging is a critical skill in computer science and problem-solving because algorithms rarely work perfectly on the first attempt. By learning systematic debugging techniques, students develop persistence, logical reasoning, and attention to detail. These skills transfer beyond programming into mathematics,	

	engineering, and everyday problem-solving situations where identifying and fixing mistakes is necessary.”	step, encouraging discovery of informal debugging behaviors such as tracing. **Explain** The teacher formally introduces debugging techniques, explicitly naming strategies students may have already begun using: ● Step-by-step tracing (dry run) ● Using test cases with known outputs ● Isolating sections of the algorithm ● Peer review or pair debugging ● Checking logic conditions Students revisit the algorithm from the Explore phase and apply at least two named techniques. The teacher models thinking aloud while debugging a sample algorithm, demonstrating how a programmer systematically tests and revises. **Elaborate** Students receive a new, more complex flawed algorithm related to a real-world scenario (such as calculating total cost with tax and discount). Individually or in small groups, they must: ● Analyze the algorithm
Explicit Connection to Earlier Learning	Previously, students learned how to design algorithms using sequencing, selection, and iteration. They practiced writing step-by-step solutions and representing processes logically. This lesson builds on that foundation by helping students recognize that even well-planned algorithms can contain errors and that computer scientists use structured techniques to locate and fix those errors. Students will connect their prior knowledge of algorithm structure to the process of analyzing where logic breaks down.	
Supports Available	Accessibility: Color-coded errors Vocabulary supports	

		● Select appropriate debugging techniques ● Apply those techniques ● Produce a corrected version ● Write a brief explanation of the debugging process used Extension opportunities are provided for advanced learners, such as creating their own flawed algorithm for another group to debug. **Evaluate** Students demonstrate mastery by submitting the corrected algorithm and a reflection explaining: ● What error(s) were found ● Which debugging techniques were used ● Why those techniques were effective The teacher assesses both the accuracy of the corrected algorithm and the student's ability to articulate the debugging process, ensuring alignment to the TEKS expectation of analyzing and applying techniques.