Kit’s Computing Challenge: Mastering Algorithms and Debugging

Learning objective: To design, write, and debug programs that accomplish specific goals by controlling or simulating physical systems, and to use sequence, selection, and repetition in programs.

Read each scenario carefully. Use your knowledge of logic, loops, and variables to solve Kit’s computing puzzles. Ensure your explanations are clear and use correct technical vocabulary.

Word bank: algorithm · debug · loop · sequence · variable · selection · decomposition · syntax

1. Kit is writing a program to help a robot navigate a maze. The robot must move forward 5 squares, turn right, and move forward 3 squares. Write this as a simple algorithm using pseudo-code. (2 marks)

2. Explain why it is important to 'decompose' a complex problem into smaller parts before you begin writing your code. Give an example of a task you could decompose. (3 marks)

3. Compare and contrast a 'fixed loop' (repeating a set number of times) and a 'conditional loop' (repeating until a condition is met). When might you use each? (4 marks)

4. Kit’s code for a game character jumping is: 'If spacebar is pressed, then jump.' Identify what type of logic is being used here (sequence, selection, or repetition) and justify your answer. (2 marks)

5. You have written a program that should calculate the total cost of items at a shop, but the output is always zero. Describe the process you would take to debug this error. (3 marks)

6. What might happen if a computer program contained an 'infinite loop' with no exit condition? Explain the impact on the system's performance. (2 marks)

Draw: Draw a flowchart that represents the logic for a simple vending machine that accepts £1 coins and dispenses a snack if the button is pressed.

Extension challenge: Design a 'Greater Depth' challenge: Create a pseudo-code script for a smart-home system that uses a variable to track the temperature. The system should trigger a heater if the temperature drops below 18°C, but stop the heater if a window is detected as 'open'. Justify your logic structure.