

Kit's Code-Cracking Log: The Debugging Detective

Learning objective: To identify, explain, and correct errors (bugs) within an algorithm or a sequence of instructions.

Use this log to act as a digital detective. When you find an error in your code, describe the bug, explain why it happened, and write the corrected version. Work methodically to ensure your program runs exactly as intended.

Word bank: Algorithm · Sequence · Syntax error · Logic error · Predict · Outcome · Refine · Efficiency

1. The Unexpected Outcome: Describe the exact moment your program stopped working or produced the wrong result. What should have happened instead? (2 marks)

2. Bug Identification: Is this a 'syntax error' (a spelling or structural mistake) or a 'logic error' (the code runs, but the result is wrong)? Explain your reasoning. (3 marks)

3. The Investigation: Explain why the bug occurred. Did you miss a step in your sequence, or was a command misinterpreted by the system? (3 marks)

4. The Solution: Write out the corrected sequence of instructions. How have you modified your code to ensure it now functions correctly? (3 marks)

5. Predict and Test: What might happen if you changed one of your variables or conditional statements in this code? Justify your prediction. (4 marks)

6. Reflection: Compare and contrast your original attempt with your debugged version. Why is the second version more efficient or reliable? (3 marks)

Extension challenge: Imagine you are teaching a friend how to avoid this specific bug in the future. Write a 'Best Practice' guide consisting of three rules they must follow when writing code to ensure their programs are 'bug-proof'.