



**Instructions:** No answer bank: write or fix the Lua code. Number the steps in order, answer the case study questions, explain the key terms, then write full-sentence answers and plan the extended answer.

### 1. Code puzzles

| Question                               | Working and answer |
|----------------------------------------|--------------------|
| Find the bug: for i = 1, 3 do print(i) |                    |
| Find the bug: if x = 5 then            |                    |
| Find the bug: print("Hi)               |                    |
| Find the bug: Local score = 0          |                    |
| Find the bug: task.wait(1              |                    |

### Did you know?

- Grace Hopper's team taped the 1947 moth into their logbook.
- Roblox Studio has breakpoints that pause code on a chosen line.
- Explaining code aloud to a toy is called rubber duck debugging.
- Red text in Output is an error; orange text is a warning.

### Background

A bug is a mistake in code that makes it do the wrong thing. Every coder makes bugs, even experts. Finding and fixing them is called debugging. Bugs are not bad news. They are puzzles, and solving them makes you a better coder.

### 2. Steps in order

Number the steps 1 to 6 (1 = first).

|   |                                               |
|---|-----------------------------------------------|
| 1 | <b>Read:</b> Read the code or error message   |
| 2 | <b>Test:</b> Run it again                     |
| 3 | <b>Guess:</b> Think about which part is wrong |
| 4 | <b>Repeat:</b> Keep going until it works      |
| 5 | <b>Notice:</b> See what went wrong            |
| 6 | <b>Fix:</b> Change one thing                  |

### 4. Key terms

| Key term      | What does it mean? Give an example. |
|---------------|-------------------------------------|
| Bug           |                                     |
| Debugging     |                                     |
| Error message |                                     |
| Line number   |                                     |
| Test          |                                     |

### 3. Case study

*Tayo's door should open when he has 5 coins. He has 5 coins but the door stays shut. His code says if coins > 5 then open the door.*

Case study: the stubborn door

1. What is the bug?

\_\_\_\_\_

2. How could print help?

\_\_\_\_\_

3. Write the fixed condition.

\_\_\_\_\_

### 5. Think it through

1. Explain rubber duck debugging.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2. Explain why line numbers in error messages are useful.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

### 6. Longer answers (full sentences)

1. Describe how to use print to find a bug.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2. What is the difference between an error and a bug that gives no error?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

3. Why change one thing at a time?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Explain in your own words: How do error messages help a coder?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

7. Extended answer

Extended answer: 'Debugging is more important than writing new code.' How far do you agree?

Plan: introduction · point for (example) · point against (example) · conclusion with your judgement.

---

---

---

---

---

---