

COMPUTING · GRADES 4–5

Debugging

Why your code doesn't work — and how to fix it

Why 'bug'?

In 1947, an engineer named Grace Hopper found an actual MOTH stuck in a computer that was malfunctioning. She taped it to her notebook and labeled the problem a 'bug'. The name stuck. Today, any error in code is called a bug, and fixing one is called **DEBUGGING**.

Three common bugs

Type	What it is	How to spot
Syntax error	You wrote the code in a way the computer can't read.	Computer refuses to run. Often shows a line number.
Logic error	Code RUNS but does the wrong thing.	Output is unexpected. e.g. you wanted to add but used minus.
Runtime error	Code starts running but crashes when it hits something it can't handle.	e.g. dividing by zero, accessing a list item that doesn't exist.

Five debugging steps

- **READ** the error message carefully (don't panic — they usually point at the line)
- **PRINT** the values of variables to see what the computer 'thinks' is happening
- **ASK FOR HELP** — fresh eyes spot bugs faster than tired ones
- Check the **OBVIOUS** first — typos, missing brackets, capital vs lowercase
- **TALK** through the code line by line, out loud, like you're teaching it (rubber-duck debugging)

