



Bit by Bit Computer Science

5 Python Programming Mistakes Every Student Makes

OCR • AQA • Edexcel

🕒 25 MINUTES TO COMPLETE

- ✓ 5 Real Student Mistakes, Explained
- ✓ The Fix & an Exam Tip for Each One
- ✓ Student Debug Tasks Throughout
- ✓ Final Challenge: Fix All 5 at Once
- ✓ Works for OCR, AQA & Edexcel Python

Why These 5 Mistakes?

These are the five errors that come up again and again in GCSE Python programming — in class, in homework, and in the exam itself. Every one of them is easy to make and, once you know what to look for, easy to avoid.

- ✓ Using `=` instead of `==`
- ✓ Forgetting that **`input()`** returns a string
- ✓ Off-by-one errors with **`range()`**
- ✓ Incorrect indentation
- ✓ Case-sensitive variable name typos

Who This Is For

Written for **GCSE Computer Science students** (Year 10–11) studying Python programming under OCR (J277), AQA (8525) or Edexcel (1CP2). Suitable for whole-class teaching, independent revision, or as a quick homework task.

Suggested Timing (25 Minutes)

Time	Activity
10 min	Mistakes 1 & 2, with student tasks (p.3)
7 min	Mistakes 3 & 4, with student tasks (p.4)
3 min	Mistake 5 (p.5)
5 min	Final Challenge: debug all 5 (p.5)

HOW TO USE THIS BOOKLET

Read each mistake's broken code first and try to spot the problem yourself before reading the fix. Then complete the Student Task in your own book or on paper - these practise the debugging and code-correction skills assessed across GCSE Computer Science specifications. If possible, type the corrected code into Python afterwards to check that it works. Answers for every task are on page 6.

Mistakes 1 & 2

1 Using = Instead of ==

```
if score = 10:  
    print("Perfect!")
```

A single `=` assigns a value to a variable — it doesn't compare two values. Python won't even run this line; it raises a **SyntaxError**.

```
if score == 10:  
    print("Perfect!")
```

Exam Tip: Use `==` to compare two values. Whenever your code won't run and the error points at an `if` line, check for a stray single `=` first — it's a very common syntax error in GCSE Python.

STUDENT TASK

Circle the mistake below, then write the corrected line in your book:

```
if grade = "A":  
    print("Excellent")
```

2 Forgetting `input()` Returns a String

```
age = input("Enter your age: ")  
if age >= 18:  
    print("You can vote")
```

`input()` always returns a string, even if the user types a number. Comparing a string to an integer with `>=` raises a **TypeError**.

```
age = int(input("Enter your age: "))  
if age >= 18:  
    print("You can vote")
```

Exam Tip: When numeric input is required, exam questions often expect you to convert the value using `int()` or `float()`. Check the required data type carefully, and get into the habit of converting every numeric `input()` straight away.

STUDENT TASK

Rewrite this line so **height** is correctly stored as a decimal number:

```
height = input("Enter your height in metres: ")
```

Mistakes 3 & 4

3 Off-by-One Errors with range()

```
for i in range(1, 5):  
    print(i)  
# Student expected: 1 2 3 4 5
```

range(1, 5) stops *before* the second number, so it only produces 1, 2, 3, 4 — the 5 never appears.

```
for i in range(1, 6):  
    print(i)
```

Exam Tip: Before you trust a loop, always ask: "does my loop need to include the last number?" If yes, add 1 to the stop value in **range()**.

STUDENT TASK

How many times does this loop actually run? Write your answer, then check it by tracing the code line by line.

```
for i in range(3, 8):  
    print(i)
```

4 Incorrect Indentation

```
for i in range(3):  
print(i)
```

Python uses indentation to show which lines belong inside a loop or if statement. Because **print(i)** isn't indented, Python doesn't see it as part of the loop and raises an **IndentationError**.

```
for i in range(3):  
    print(i)
```

Exam Tip: Indent the line that belongs inside the loop by exactly one level (4 spaces). Never mix tabs and spaces in the same program — your editor may display indentation inconsistently or report an error.

STUDENT TASK

This code should print "Go!" three times. Rewrite it with correct indentation:

```
for i in range(3):  
print("Go!")
```

Mistake 5 & Final Challenge

5 Case-Sensitive Variable Name Typos

```
Score = 0
score = score + 10
print(Score)
```

Python treats **Score** and **score** as two completely different variables. The second line updates **score**, but the last line prints **Score** — which is still 0.

```
score = 0
score = score + 10
print(score)
```

Exam Tip: Pick a naming style (e.g. all lowercase with underscores) at the start of every program and use it consistently throughout — this single habit prevents a huge number of exam errors.

STUDENT TASK

Find the two different variable names being used by mistake, then rewrite the code so they match:

```
Total = 0
total = total + 5
print(Total)
```

FINAL CHALLENGE · 5 MINUTES

This program should ask for the user's name and age, print every age from 1 up to the user's current age, then print a special message if they have just turned 13.

```
Name = input("Enter your name: ")
age = input("Enter your age: ")

for i in range(1, age):
    print(name, "was", i)

if age = 13:
    print(name, "is now a teenager!")
```

STUDENT TASK — DEBUG ALL FIVE

This code contains all five mistakes from this booklet. Working on your own or with a partner, find and fix every one, then rewrite the corrected program underneath. *Hint: there's a missing type conversion, an off-by-one error in the range, an indentation error, an = instead of ==, and a case-sensitive variable name typo.* Answers are on page 6.

Answers

Mistakes 1-5: Student Task Answers

1. Comparison operator

```
if grade == "A":  
    print("Excellent")
```

A single = assigns; == compares.

2. Converting input()

```
height = float(input("Enter your height in metres: "))
```

Decimal values need float(), not int().

3. range() off-by-one

```
for i in range(3, 8):  
    print(i)
```

The loop runs **5 times** (3, 4, 5, 6, 7) — range(3, 8) stops before 8.

4. Indentation

```
for i in range(3):  
    print("Go!")
```

The line inside the loop must be indented.

5. Case-sensitive typo

```
total = 0  
total = total + 5  
print(total)
```

Total and total were being treated as different variables.

Final Challenge: Corrected Program

```
name = input("Enter your name: ")  
age = int(input("Enter your age: "))  
  
for i in range(1, age + 1):  
    print(name, "was", i)  
  
if age == 13:  
    print(name, "is now a teenager!")
```

WHAT CHANGED

age is converted with **int()** so it can be used in **range()** and compared with numbers · **range(1, age + 1)** now includes the final year · **print(name, "was", i)** is indented so it runs inside the loop · **if age == 13:** uses the comparison operator, not assignment · **name** is spelled consistently throughout, matching the variable defined on the first line.

Scan for resources

Enjoyed this booklet?

Follow @bitbybitsuk on Instagram and TikTok for more free GCSE Computer Science revision. Visit www.bitbybits.com for 1-to-1 tutoring and further OCR, AQA and Edexcel resources.

