



F4.3 Local and global variables

Functions and structured code · GCSE · OCR J277 2.2.3, AQA 8525 3.2.10,
Edexcel 1CP2 6.6.3 · about 15 min

What this lesson is about

Scope: where a variable exists, and why passing values beats global.

- Local variables stay inside
- Same name, different variables
- Reading a global works; changing it does not
- Fix 1: pass it in and return it
- Fix 2: the global keyword

Questions

6 marks in all

1. A function sets `gap = 10` inside its body. After calling it, the program runs `print(gap)`. What happens? [1 mark]

- ☐ A NameError: gap only exists while the function runs
- ☐ B It prints 10
- ☐ C It prints gap
- ☐ D It prints nothing

Answer: A. A variable made inside a function belongs to it. If the outside needs the value, return it.

2. What does this program print? [1 mark]

```
speed = 80

def gentle():
    speed = 30
    print(speed)

gentle()
print(speed)
```

Answer:

```
30
80
```

The speed inside the function is a new local variable, so the global speed is unchanged.

3. `beeps = 0` outside, and inside a function `beeps = beeps + 1` with no `global` line. What happens when the function runs? [1 mark]

- ☐ A UnboundLocalError: assigning makes beeps local, and it has no value yet
- ☐ B beeps becomes 1
- ☐ C The global beeps becomes 1
- ☐ D NameError

Answer: A. Assigning to a name anywhere in a function makes it local throughout that function.

4. Which are reasons local variables are good practice?

[1 mark]

Tick every answer that is true.

- ☐ A A subprogram can be understood and tested on its own
- ☐ B The same names can be used safely in different subprograms
- ☐ C The memory is freed when the subprogram ends
- ☐ D They make the program run without errors

Answer: A, B, C. Local variables keep subprograms self-contained. They do not stop errors on their own.

5. What does this program print?

[1 mark]

```
count = 0

def add():
    global count
    count = count + 1

add()
add()
print(count)
```

Answer:

2

global count means the function changes the global variable, so two calls make it 2.

6. What is the scope of a variable?

[1 mark]

- ☐ A The part of the program where it can be used
- ☐ B The largest value it can hold
- ☐ C Its data type
- ☐ D How long its name is

Answer: A. A local variable's scope is its subprogram; a global variable's is the whole program.

The task: count the beeps

This program should beep three times and print `beeps: 3`, but it crashes. Fix it, with parameters and `return` or with `global`.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

beeps = 0

def beep():
    tone(880, 0.1)
    wait(0.1)
    beeps = beeps + 1

beep()
beep()
beep()
print("beeps:", beeps)
```

The hint students can ask for: A function cannot change a variable made outside it unless you say so. Either pass the count in and hand the new one back, or declare the outside variable `global` at the top of the function. Pick one and keep it consistent.

A solution

```
from bugbot import *
connect()
def beep(count):
    tone(880, 0.1)
    wait(0.1)
    return count + 1

beeps = 0
beeps = beep(beeps)
beeps = beep(beeps)
beeps = beep(beeps)
print("beeps:", beeps)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.