ANSWERS

F4.2 Parameters and return values

Functions and structured code · GCSE · OCR J277 2.2.3, AQA 8525 3.2.10, Edexcel 1CP2 6.6.2 · about 15 min

What this lesson is about

Information in and an answer out; procedures and functions.

- Parameters: information in
- Several parameters
- return: an answer out
- Printing is not returning
- Procedures and functions

Questions

7 marks in all

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

```
def double(n):
    return n * 2

print(double(4))
print(double(10))
```

Answer:

```
8
20
```

return hands the value back to the call, and print shows it.

2. In `def square(size):`, what is `size`? [1 mark]

- ☐ **A** A parameter: it gets its value from the call
- ☐ **B** An argument
- ☐ **C** A return value
- ☐ **D** A comment

Answer: A. size is a parameter. In square(15), the 15 is the argument that gives it its value.

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

```
def greet(name, times):
    for i in range(times):
        print("hi", name)

greet("Ada", 2)
```

Answer:

```
hi Ada
hi Ada
```

The values in the call fill the parameters in the same order.

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

```
def double_print(n):
    print(n * 2)

def double_return(n):
    return n * 2

a = double_print(5)
b = double_return(5)
print(a, b)
```

Answer:

```
10
None 10
```

double_print shows 10 but returns nothing (None); double_return gives 10 back to be stored.

5. In `area(20, 30)`, what are 20 and 30? [1 mark]

- ☐ **A** Arguments
- ☐ **B** Parameters
- ☐ **C** Return values
- ☐ **D** Local variables

Answer: A. The values in a call are arguments; the names in the definition are parameters.

6. What is the difference between a function and a procedure? [1 mark]

- ☐ **A** A function returns a value; a procedure does not
- ☐ **B** A procedure has parameters; a function cannot
- ☐ **C** A function is written in capitals
- ☐ **D** There is none in any language

Answer: A. Python uses def for both; OCR's Reference Language has separate words, and AQA uses SUBROUTINE for both.

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

```
def check(n):
    if n > 10:
        return "big"
    return "small"

print(check(15), check(3))
```

Answer:

```
big small
```

return ends the function straight away, so only one of the two returns runs for each call.

The task: any polygon

Write `polygon(sides, length)` that prints `polygon with <sides> sides` and drives the shape. Call it for a pentagon (5 sides, 20 cm) and then a triangle (3 sides, 20 cm). Finish facing roughly the way you started.

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

def polygon(sides, length):
    print("polygon with", sides, "sides")
    # drive it

polygon(5, 20)
```

The hint students can ask for: The function needs the number of sides and the length. Work the turn out from the number of sides: a full turn shared between them. Then call it twice with different arguments.

A solution

```
from bugbot import *
connect()
def polygon(sides, length):
    print('polygon with', sides, 'sides')
    for i in range(sides):
        forward(60, distance=length)
        turn_right(30, angle=360 / sides)
polygon(5, 20)
polygon(3, 20)
```

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