Grade 8 Computing

Unit 4 · Functions: Building Blocks of Programs

ANSWER KEY · FOR TEACHERS

Full marks 30

Give the marks for the points in the marking scheme; an answer in other words that says the same is right.

1 Key words

K · 4

1. decomposition
2. function
3. definition
4. call
5. parameters
6. arguments
7. None
8. docstring

Each carries 0.5 mark.

2 Label the diagram

U · 2

1. house
2. roof a triangle
3. trunk a rectangle
4. tree

Each carries 0.5 mark.

3 Multiple choice

i. **b.** `def square(size):` — K · 1

`def`, the name, the parameters in brackets and a colon. `square(size)` is a call, and Python has no `function` or `define` keyword.

ii. **b.** Python stops with a TypeError: one argument is missing. — U · 1

`bill` needs two arguments and got one: `TypeError: bill() missing 1 required positional argument: 'samosas'`. Only a parameter with a default, like `samosas=0`, may be left out.

iii. **a.** one, the function — U · 1

The 20 calls all run the same lines, the function's body. Fix it there, once, and all 20 calls are fixed. That is one of the main reasons to write functions.

iv. **a.** Hari is the first name in the list. — U · 1

The function returns the position where it found the target, and positions start at 0: Hari is the first name. "Not found" is −1.

v. **c.** It documents the function: `help()` shows it. — K · 1

A docstring is a string on the first line of a function's body. Python keeps it as the function's documentation and `help()` shows it; the function runs exactly as before.

4 What does it print?

MARKING SCHEME — A · 3

- ✓ 9: `double(4)` is replaced by the 8 it returns, and 8 + 1 is 9.
- ✓ 6, printed by `show` itself during the call on line 9.
- ✓ `None`: `show` has no `return`, so **b** gets `None`.

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

5 Trace the program

ANSWER A · 4

Call on line	`cups`	`samosas`	`total` returned	`bill` after the line	Output
6	2	1	85	85	
7	1	3	105	190	
8				190	190

MARKING SCHEME

- ✓ Line 6: the arguments 2 and 1 go into **cups** and **samosas**, in order.
- ✓ Line 6 returns 2 × 30 + 1 × 25 = 85, and **bill** becomes 85.
- ✓ Line 7 returns 1 × 30 + 3 × 25 = 105, and **bill** becomes 190.
- ✓ Output `190`, once, at line 8.

6 Find and fix the bug

MARKING SCHEME A · 3

- ✓ 40 is printed by the `print` inside the function, during the call.
- ✓ The function returns nothing, so **a** is `None`, and `None` can't be added to 10.
- ✓ `return length * width` on line 2; the program then prints 50.

7 Reading documentation

MARKING SCHEME A · 3

- ✓ i. Only **height** must be given; **x**, **y** and **outline** are optional (they have defaults).
- ✓ ii. `draw_flag(200, -100, -50)`.
- ✓ iii. `width = draw_flag(100, outline=False)` (or `draw_flag(100, 0, 0, False)`), the return value kept in a variable.

8 Write a program

ANSWER A · 4

```
def highest(marks):
    best = marks[0]
    for i in range(1, len(marks)):
        if marks[i] > best:
            best = marks[i]
    return best

print(highest([52, 38, 71, 45]))
```

```
OUTPUT
71
```

Starting **best** at `marks[0]` also works for lists of negative numbers; starting at 0 is accepted for marks. A `for m in marks:` loop is right too.

MARKING SCHEME

- ✓ `def highest(marks):` with a parameter for the list.
- ✓ A "highest so far" variable, set before the loop, and changed inside an if when a bigger item comes.
- ✓ `return` with the answer after the loop (not `print` inside the function).
- ✓ A call that prints the answer, 71.

9 Think

ANSWER

Write the four lines once, as a function such as `receipt_line(item, price)`, and replace each copy with a one-line call. If the receipt's layout changes, or a bug is found, it is

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

changed once, in the function, and all seven places follow; with copies it is easy to fix six and forget the seventh. The program is also shorter and easier to read, and the function can be tested on its own and used again elsewhere.

MARKING SCHEME

- ✓ One place to fix or change: the function is changed once and all seven places follow (copies can be missed).
- ✓ Any one more: a shorter, clearer program; the function can be tested on its own; it can be reused elsewhere.

10 Activity: Look it up, then use it

(not marked)

WHAT STUDENTS SHOULD FIND

dot draws a filled circle at the turtle's position, with its center there: `dot(size=None, *color)`. Nothing is needed; without a size it draws a small dot, and without a color it uses the pen's color. **write** writes text where the turtle is: `write(arg, move=False, align='left', font=('Arial', 8, 'normal'))`; only the text, **arg**, is needed. One possible sun:

```
import turtle
t = turtle.Turtle()
t.dot(100, "gold")
t.penup()
t.goto(0, -80)
t.write("Sun", align="center")
turtle.done()
```

Marks by level	Knowledge	Understanding	Application	Higher ability	Total
Marks	6	5	17	2	**30**

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking