Grade 8 Computing

Unit 1 · From Blocks to Python

ANSWER KEY · FOR TEACHERS

Full marks 33

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

1 Key words

1. text-based language
2. Python interpreter
3. comment
4. float
5. string
6. integer division
7. remainder
8. f-string

Each carries 0.5 mark. **K · 4**

2 Multiple choice

i. **b.** 105 **U · 1**

Both are strings, in quotes, so + joins them end to end: 105. To add, leave out the quotes: `print(10 + 5)` shows 15.

ii. **c.** 3 **A · 1**

23 ÷ 4 = 5 remainder 3, and `%` gives the remainder: 3. (`23 // 4` is 5, and `23 / 4` is 5.75.)

iii. **b.** `int()` was given text that isn't a whole number. **U · 1**

`int()` can only turn text like `"12"` into a number. For `"12.5"` use `float()`. A spelling mistake would be a NameError or SyntaxError, and joining a number to a string a TypeError.

iv. **b.** `float` **U · 1**

`/` always gives a float: `7 / 7` is `1.0`. `7 // 7` would give the int `1`.

v. **c.** `marks_2` **U · 1**

`marks_2` has only letters, a digit and an underscore, and doesn't start with a digit. `total marks` has a space, `2nd_place` starts with a digit, and `if` is one of Python's own keywords.

3 What does it print?

a. MARKING SCHEME **U · 3**

- ✓ `7 14`: line 4 uses the new **a**, 7, so **b** is 7 × 2 = 14.
- ✓ `0 7`: 14 doesn't go into 7 at all, so 7 is all left over.
- ✓ `7 and 14`, the f-string filled in with the two values.

b. MARKING SCHEME **A · 3**

- ✓ `444 12`: a string times 3 is the string three times; the int is multiplied.
- ✓ `False True`: the string `"4"` is never equal to the number 4, but `int("4")` is.
- ✓ `2.0 1 1`: `/` always gives a float; 3 goes into 4 once, with 1 left over.

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

4 Trace the program

ANSWER A · 4

Line	money	price	books	Output
1	500			
2	500	60		
3	500	60	8	
4	20	60	8	
5	40	60	8	
6	40	60	8	8 books, Rs 40 left

MARKING SCHEME

- ✓ `books` = 8 at line 3: 60 goes into 500 eight whole times (`//`).
- ✓ `money` = 20 at line 4: 500 − 8 × 60 (the multiplying is done first).
- ✓ `money` = 40 at line 5: the right side is worked out first, 20 + 20, and put back into `money`.
- ✓ The output `8 books, Rs 40 left`, at line 6 only.

5 Find and fix the bug

ANSWER A · 4

i. `Asha, please pay Rs 222222222222222222222222222222`: the 2 thirty times.

ii. Line 2: `input()` always gives a string, so **cups** holds the text `"2"`, and `"2" * 30` repeats it thirty times.

iii. `cups = int(input("How many cups? "))`

iv. A TypeError: **bill** is now an int, and + can't join a string to a number. `print(name + ", please pay Rs " + str(bill))`, or `print(f"{name}, please pay Rs {bill}")`. It then prints `Asha, please pay Rs 60`.

MARKING SCHEME

- ✓ i. The digit 2 written thirty times (no error message: just the wrong answer).
- ✓ ii. Line 2, because `input()` gives a string, not a number.
- ✓ iii. `int()` round `input()` on line 2 (or `int(cups)` on line 3).
- ✓ iv. TypeError, corrected with `str(bill)`, an f-string, or commas in `print`.

6 Write a program

ANSWER A · 4

```
price = 30
plates = int(input("How many plates of momo? "))
bill = plates * price
bill = bill + 20
print(f"Total with the delivery charge: Rs {bill}")
```

```
OUTPUT
How many plates of momo? 4
Total with the delivery charge: Rs 140
```

`when ▶ Run is pressed` has no line of its own: a Python program runs from its first line. `print("Total with the delivery charge: Rs", bill)` or `str(bill)` with + are right too.

MARKING SCHEME

- ✓ `price = 30`, and the answer asked with `input()` and turned into a number with `int()`.
- ✓ `bill = plates * price` (with `*`).
- ✓ **change by 20** as `bill = bill + 20` (or `bill += 20`).
- ✓ The message printed with the bill, and the output for 4: Rs 140.

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

7 Short answers

MARKING SCHEME U · 2

- ✓ = sets a variable, with an example such as `age = 13`.
- ✓ == compares two values and gives `True` or `False`, with an example such as `age == 13`.

8 Name the error

U · 2

1. `print("Namaste)` → **c.** SyntaxError
2. `pirnt("Hi")` → **e.** NameError
3. `print("Rs " + 60)` → **a.** TypeError
4. `print(int("two"))` → **b.** ValueError

Each carries 0.5 mark.

9 Think

ANSWER HA · 2

What a user types is characters, and Python can't know what the programmer means them to be: 12.5 could be a price or a version number, and 0098 a phone number whose zeros matter. A guess would sometimes be wrong, and a wrong guess gives a wrong answer with no error message, the worst kind of bug. So Python doesn't guess: the programmer says which type is wanted, with `int()`, `float()` or nothing, and anyone reading the program can see it.

MARKING SCHEME

- ✓ What is typed is characters (text), and only the programmer knows whether it should be a whole number, a decimal or text (a name, a phone number).
- ✓ A guess could be wrong without any error message; with `int()` or `float()` the programmer decides, and the program says so where everyone can read it.

10 Activity: Fix five programs

(not marked)

WHAT STUDENTS SHOULD FIND

1. `SyntaxError: '(' was never closed.` Add `)`.
2. `NameError: name 'Print' is not defined. Did you mean: 'print'?` Use a small p.
3. `TypeError: can only concatenate str (not "int") to str.` Use `str(age)` or an f-string.
4. `ValueError: invalid literal for int() with base 10: 'forty'.` Use `int("40")`, or just `40`.
5. `NameError: name 'true' is not defined. Did you mean: 'True'?` Python's true is `True`, with a capital T.

Students should notice that the kind of error comes first on the last line, and that Python often guesses the fix ("Did you mean…").

Marks by level	Knowledge	Understanding	Application	Higher ability	Total
Marks	4	11	16	2	**33**

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