Python Programming

Level 2 (Medium)

https://cedricf6.github.io/python-course/

Python Programming - Level 2 (Medium) - Questions

Section A: Output, Variables & Type Conversion

1. What is the output of the following code snippet?

```
result = 15 / 4 + 2 ** 3
print(int(result))
```

Answer:

---

2. The following code is intended to calculate the average of two numbers entered by the user. Identify and correct the **logical error**.

```
num1 = input("Enter first number: ")
num2 = input("Enter second number: ")
average = (num1 + num2) / 2
print(average)
```

Answer:

---

3. What is the value and data type of the variable `final_value` after the following operations?

```
a = "10"
b = 5.5
final_value = float(a) + b
```

Answer:

Section B: Operators & Precedence

4. What is the value of `output` after the following operation is executed?

```
x = 20
output = x % 7 * 2
```

Answer:

---

5. Rewrite the following expression using an appropriate assignment operator:

`total = total - discount`

Answer:

---

6. Assuming `a = 5`, `b = 2`, and `c = 3`, what is the result of the following boolean expression? `not (a < b) and c % b == 1`

Answer:

Section C: Complex Decision Statements

7. What is the output of the following nested `if` structure?

```
score = 82
if score >= 50:
    if score >= 75:
        print("Distinction")
    else:
        print("Pass")
else:
    print("Fail")
```

Answer:

---

8. Complete the following code snippet using a **membership operator** to check if the character `char` is a vowel.

```
char = 'e'
vowels = "aeiouAEIOU"
if char ____ vowels: # Fill in the blank with the operator
    print("Vowel")
```

Answer:

---

9. The following code should print "Eligible" if a person is between 18 and 65 years old (inclusive). Identify the **logical error**.

```
age = 17
if age >= 18 or age <= 65:
    print("Eligible")
else:
    print("Not Eligible")
```

Answer:

Section D: Advanced Iteration (Loops)

10. What is the output of the following `for` loop using the `range()` function with a step?

```
for i in range(2, 10, 3):
    print(i, end=' ')
```

Answer:

---

11. How many times will the string "Processing..." be printed?

```
count = 5
while count != 0:
    print("Processing...")
    count //= 2
```

Answer:

---

12. What is the output of the following loop that contains a `continue` statement?

```
for num in range(1, 6):
    if num == 3:
        continue
    print(num, end=' ')
```

Answer:

Section E: Functions & Return Values

13. What is printed when the following program runs?

```
def calculate(x, y):
    return x * y - 2

result = calculate(3, 4)
print(result)
```

Answer:

14. The following function is supposed to check if a number is even. What should the return statement be to make this function work correctly?

```
def is_even(number):
    if number % 2 == 0:
        return ____ # What should be returned?
    else:
        return ____ # What should be returned?
```

Answer:

15. What does the following import statement allow a programmer to do?

```
from random import choice as pick_one
```

Answer:

Section F: String Manipulation

16. What is the value of the variable `new_text`?

```
text = "Mississippi"
new_text = text.replace("ss", "zz", 1)
```

Answer:

---

17. What is the value of the variable `result`?

```
phrase = "Hello World"
result = phrase.find("W") + len(phrase.lower())
```

Answer:

---

18. What is the output of the following string slicing operation?

```
word = "Encyclopedia"
print(word[4:8:2])
```

Answer:

Section G: Data Structures in Action

19. What is the output of the following code that manipulates a list?

```
data = [1, 2, 3]
data.insert(1, 99)
data.reverse()
print(data)
```

Answer:

---

20. What is the final state of the list `numbers`?

```
numbers = [5, 2, 8, 2, 1]
numbers.remove(2)
numbers.sort()
print(numbers)
```

Answer:

---

21. What is the output of the following code using a **tuple**?

```
dimensions = (200, 50)
# dimensions[0] = 250 # This line is commented out. What if it wasn't?
print(dimensions[1])
```

Answer:

---

22. What is the output of the following code using a **dictionary**?

```
student = {"name": "Ben", "grade": 87}
student["grade"] = 92
student["class"] = "9B"
print(student.get("name"), student.get("class"))
```

Answer:

---

23. What does the `.items()` method return when used on a dictionary?

Answer:

Section H: Random Module & Combined Concepts

24. What is a possible output of the following code?

```
import random
options = ["Rock", "Paper", "Scissors"]
selection = random.choice(options)
print(f"The computer chose: {selection[0]}")
```

Answer:

---

25. The `random.shuffle()` function is used to randomize the order of items in a **list**. Why would it produce an error if used on a tuple?

Answer:

---

26. Write a single line of code using the `random` module to generate a random **even** number between 10 and 20 (inclusive).

Answer:

Section I: Error Identification & Good Practices

27. Identify the **type of error** in the following code (Syntax, Logical, or Runtime).

```
total = 10
count = 0
average = total / count # This line causes an error
print(average)
```

Answer:

---

28. The following code is poorly written. Suggest **two** improvements based on good programming practices.

```
x = 12 # x is for age
n = "Jake" # n is name
if x>18:print(n,"can vote")
```

Answer:

---

29. A programmer writes a function but forgets to include a `return` statement. If the function is called and its result is assigned to a variable, what will be the value of that variable?

Answer:

---

30. Why is it considered a good practice to use the `get()` method instead of direct square brackets `[]` when accessing a dictionary value if you are unsure the key exists?

Answer:

Python Programming - Level 2 (Medium) - Answers

1. **11** (15/4 = 3.75, 2**3=8, 3.75+8=11.75, converted to int becomes 11)

2. **The `input()` function returns strings.** The error is `(num1 + num2)`. This concatenates the strings (e.g., "5" + "3" = "53") instead of adding numbers. Correction: Convert inputs to numbers first.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
```

3. Value: `15.5`, Data Type: `float`

4. 12

5. `total -= discount`

6. True (`a < b` is False, so `not False` is True. `c % b` is 1, so `1 == 1` is True. `True and True` evaluates to True).

7. Distinction

8. `in` (The membership operator `in`)

9. **Logical Error**: The `or` operator is used incorrectly. It will print "Eligible" for *any* age because *every* number is either >=18 OR <=65 (e.g., a 100-year-old is <=65? False, but is >=18? True -> Eligible. A 10-year-old is >=18? False, but is <=65? True -> Eligible). The correct operator is `and`.

```
if age >= 18 and age <= 65:
```

10. `2 5 8` (Starts at 2, ends before 10, steps by 3)

11. **3 times**

```
(Iteration 1: count=5 -> Print, then count=5//2=2.
Iteration 2: count=2 -> Print, then count=2//2=1.
Iteration 3: count=1 -> Print, then count=1//2=0.
Loop stops as count == 0).
```

12. `1 2 4 5` (The `continue` statement skips the print statement when `num` is 3).

13. **10** (3 * 4 = 12, 12 - 2 = 10).

14. The function should return `True` for even and `False` for odd.

15. It imports the `choice` function from the `random` module and allows it to be called using the simpler alias `pick_one`.

16. `"Mizzissippi"` (Only the first occurrence of "ss" is replaced).

17. **17** ; "Hello World" has 11 characters, including the space. So len(phrase) is 11. 6 + 11 = 17

18. `"cc"` (From index 4 to 7: 'c', 'y', 'c', 'l'. Taking every 2nd character: index4='c', index6='l').

19. `[3, 2, 99, 1]` (First, insert 99 at index 1: `[1, 99, 2, 3]`. Then reverse the list).

20. `[1, 2, 5, 8]` (`remove(2)` removes the first occurrence of 2: `[5, 8, 2, 1]`. Then `sort()` arranges it in ascending order).

21. **50** ; If the commented line was active, it would cause a **Runtime Error** (TypeError) because tuples are immutable and cannot be modified.

22. `Ben 9B`

23. It returns a view object containing key-value pairs as tuples. For example, for `{'a': 1, 'b': 2}`, it would be something like `dict_items([('a', 1), ('b', 2)])`.

24. A possible output is `"The computer chose: R"` (It prints the first character of the randomly chosen string).

25. Because tuples are **immutable**; their contents cannot be changed after creation. `shuffle()` tries to change the order of items in place, which is not allowed for a tuple.

26. `random.randrange(10, 21, 2)` or `random.choice([10, 12, 14, 16, 18, 20])`

27. **Runtime Error** (Specifically, a ZeroDivisionError).

28. **1.** Use meaningful variable names (e.g., `age` instead of `x`, `name` instead of `n`). **2.** Use proper indentation and spacing for readability (e.g., put the `print` statement on a new line after the `if` condition).

29. The value of the variable will be `None`.

30. Because `get(key)` returns `None` (or a specified default value) if the key is not found, whereas using `[]` directly will raise a **KeyError** runtime error, which will crash the program if not handled.