Editorial-Solution-W2A2: Functions, Math Operations, Object-Oriented Programming and Advanced Concepts

---

Question 1

Which of the following is NOT a built-in math function in Python?

- A) math.sqrt()
- B) math.pow()
- C) math.average()
- D) math.ceil()

Correct Answer: C

Explanation: Python's math module does not have a built-in average() function. The other options are valid math functions: sqrt() for square root, pow() for exponentiation, and ceil() for ceiling value.

Question 2

Scenario:

Sarah is writing a program to process sensor data. As part of the process, she needs to round down a floating-point number representing a sensor reading to the nearest integer. She decides to use the math.floor() function in Python.
What is the output of the following code?

```
import math

print(math.floor(3.7))
```

- A) 3
- B) 4
- C) 3.0
- D) 4.0

Correct Answer: A

Explanation: The math.floor() function returns the largest integer less than or equal to the given number. For 3.7, the largest integer less than or equal to it is 3.

Question 3

Scenario:

A software engineering student, Emily, is learning about object-oriented programming in Python. She is trying to understand how to access the attributes and methods of an object from within the class definition. She's confused about the correct way to refer to the instance of the class.

Which of the following is used to refer to the current instance of a class within its methods?

- A) me
- B) this
- C) self
- D) instance

Correct Answer: C

Explanation: In Python, 'self' is used as the first parameter in instance methods to refer to the current instance of the class. It's similar to 'this' in other programming languages but is explicitly passed and named in Python.

Question 4

What is the correct way to define a function in Python?

- A) function myFunction():
- B) def myFunction():
- C) define myFunction():
- D) func myFunction():

Correct Answer: B

Explanation: In Python, functions are defined using the 'def' keyword followed by the function name and parentheses. The correct syntax is 'def myFunction():'.

Question 5

Which of the following is a valid function definition with a default parameter?

- A) def greet(lang, name = "Guest"):
- B) def greet(name = "Guest", lang):
- C) def greet(lang, "Guest" = name):
- D) def greet("Guest" == name, lang):

Correct Answer: A

Explanation:

- In Python, default parameter values are assigned using the assignment operator.
- Parameters with default values must come after those without defaults in the function definition.

- Options C and D are syntactically incorrect, and option B violates the rule about the order of parameters.
- Option A correctly defines a function greet with a regular parameter lang and a default parameter name set to "Guest". If the function is called without providing a value for the name parameter, it will default to "Guest"

Question 6

What is the output of the following code?

```
def multiply(a, b=2):
   return a * b


print(multiply(3))
```

- A) 3
- B) 2
- C) 6
- D) Error

Correct Answer: C

Explanation: The function multiply() has a default parameter b=2. When called with only one argument (3), it uses the default value for b. Thus, the calculation is 3 * 2 = 6.

Question 7

What will be the output of the following code?

```
def factorial(n):
   return 1 if n == 0 else n * factorial(n-1)


print(factorial(5))
```

- A) 120
- B) 24
- C) 5
- D) Error

Correct Answer: A

Explanation: This code defines a recursive function to calculate the factorial of a number. For n=5, it calculates 5 * 4 * 3 * 2 * 1, which equals 120. The function correctly implements the mathematical definition of factorial.

Question 8

What will be the output of the following code?

```
numbers = [1, 2, 3, 4, 5]

squared = list(map(lambda x: x**2, numbers))

print(squared)
```

- A) [1, 4, 9, 16, 25]
- B) [2, 4, 6, 8, 10]
- C) [1, 2, 3, 4, 5]
- D) Error

Correct Answer: A

Explanation: The lambda function x: x**2 squares each number in the list. The map() function applies this lambda function to each element in 'numbers'. The result is a new list with each number squared: [1, 4, 9, 16, 25].

Question 9

Which of the following is NOT a valid way to call a function in Python?

- A) result = myFunction()
- B) myFunction()
- C) call myFunction()
- D) print(myFunction())

Correct Answer: C

Explanation: In Python, functions are called by using their name followed by parentheses. The keyword 'call' is not used to invoke functions. Options A, B, and D are all valid ways to call a function.

Question 10

What is the purpose of the 'return' statement in a function?

- A) To print the result of the function
- B) To end the function execution and specify the output value
- C) To return to the beginning of the function
- D) To define a new variable

Correct Answer: B

Explanation: The 'return' statement in a function is used to end the function execution and specify the output value that the function should produce. It allows the function to send a result back to the caller.

Question 11

Scenario:
A developer needs to create a function that processes unlimited product ratings. Which implementation correctly handles variable arguments? Note: Ignore the "pass" keyword.

```
# Option 1

def  process_ratings(args...):

        # logic

        pass


# Option 2

def  process_ratings(*scores):

        # logic`

        pass
```

Which syntax is valid for variable arguments?

- A) Only Option 1 works
- B) Both options work
- C) Only Option 2 works
- D) Neither works

Correct Answer: C
Explanation: Python uses *args syntax (not args...) to accept variable positional arguments.

Question 12

Scenario:

A security system checks passwords using:

```
stored_pass =  "secret123"

user_input =  "sEcRet123"

print(user_input.lower()  == stored_pass)`
```

What does this code output, and why is it problematic?

- A) True - makes comparison case-insensitive
- B) False - case mismatch remains
- C) True - lower() converts both to lowercase
- D) Error - incompatible types

Correct Answer: A
Explanation: *The .lower() converts user_input to "secret123", matching the lowercase stored_pass. This creates security risks by ignoring case sensitivity in passwords.*

Question 13

What is the purpose of the '**init**' method in a Python class?

- A) To define the class methods
- B) To initialize the class attributes
- C) To delete the class object
- D) To return a value from the class

Correct Answer: B

Explanation: The '**init**' method in Python is a special method (constructor) that is automatically called when an object of the class is created. It is used to initialize the attributes of the class.

Question 14

What is the output of the following code?

```
def greet(*names):
    for name in names:
        print(f"Hello, {name}!",end=" ")


greet("Alice", "Bob", "Charlie")
```

- A) Hello, Alice! Hello, Bob! Hello, Charlie!
- B) Hello, Alice Bob Charlie!
- C) Error
- D) Hello, ('Alice', 'Bob', 'Charlie')!

Correct Answer: A

Explanation: The function greet() uses *names to accept multiple arguments. It then iterates over these arguments, printing a greeting for each name.

Question 15

What is the output of the following code?

```
def outer(x):
    def inner(y):
        return x + y
    return inner


closure = outer(10)
print(closure(5))
```

- A) 15
- B) 10
- C) 5
- D) Error

Correct Answer: A

Explanation: This code demonstrates a closure. The outer() function returns the inner() function, which remembers the value of x. When we call outer(10), it returns inner with x=10. Then calling closure(5) is equivalent to inner(5), which returns 10 + 5 = 15.