

**What is the output of the following code?**

```
x = [1, 2, 3]
y = x
y[0] = 100
print(x)
```

- A) [100, 2, 3]
- B) [1, 2, 3]
- C) [100]
- D) Error

**What will be the output of this code?**

```
def add(x, y=0):
    return x + y

print(add(5, y=3))

print(add(5))
```

- A) 8, 5
- B) 8, 8
- C) 8, 0
- D) 8, None

**What is the value of x after the following code is executed?**

```
x = 10

def modify():
    global x
    x = 5

modify()

print(x)
```

- A) 10
- B) 5
- C) Error
- D) None

**What does the super() function do in Python?**

- A) Calls the parent class method
- B) Calls the constructor of the current class
- C) Calls a global function
- D) None of the above

**What is the output of this code?**

```
x = {'a': 1, 'b': 2, 'c': 3}

print(x.get('d', 'Not Found'))
```

- A) None
- B) Not Found
- C) Error
- D) 0

**What is the result of is comparison in Python?**

- A) Checks for value equality
- B) Checks for object identity
- C) Always returns True
- D) Checks for type equality

**Which of the following will raise an exception in Python?**

- A) `x = 1 / 0`
- B) `x = 'a' + 1`
- C) `x = int('abc')`
- D) All of the above

**What will be the output of the following code?**

```
x = 'abc'
```

```
y = 'def'
```

```
z = x + y
```

```
print(z * 2)
```

- A) 'abcdefabcdef'
- B) 'abcdef'
- C) 'abcdeffdef'
- D) 'abcabcabc'

**What is the output of this code snippet?**

```
def foo(a=[]):
```

```
    a.append(1)
```

```
    return a
```

```
print(foo())
```

```
print(foo())
```

- A) [1] [1]
- B) [1] [1, 1]
- C) [1] [2]
- D) [1, 1] [1, 1]

**What is the output of this code?**

```
class A:

    def __init__(self):

        self.a = 1

class B(A):

    def __init__(self):

        super().__init__()

        self.b = 2

b = B()

print(b.a)
```

- A) 1
- B) 2
- C) Error
- D) None

**What will be the output of the following code?**

```
def test(arg1, *args):

    print(arg1)

    print(args)

test(1, 2, 3, 4)
```

- A) 1 (2, 3, 4)
- B) 1 2 3 4
- C) (1) (2, 3, 4)
- D) Error

**What is the result of the expression `True == 1`?**

- A) True
- B) False
- C) Error
- D) None

**What will be the output of the following code?**

```
x = {'name': 'Alice', 'age': 25}

y = x

y['age'] = 26

print(x)
```

- A) {'name': 'Alice', 'age': 25}
- B) {'name': 'Alice', 'age': 26}
- C) Error
- D) None

### **Class Inheritance Example**

Create a class `Vehicle` with attributes like `brand` and `speed`. Derive two classes, `Car` and `Bicycle`, from `Vehicle`. Add an additional attribute for each (`Car` should have `fuel_type` and `Bicycle` should have `gear_count`). Instantiate both and display their information.

### **Class with Exception Handling**

Write a class `BankAccount` with methods `deposit()` and `withdraw()`. Implement exception handling for cases when withdrawal exceeds balance, and display an appropriate error message. Ensure that all transactions are logged into a file.

## **Inheritance and Method Overriding**

Create a base class Shape with a method area() that calculates the area. Derive two classes, Rectangle and Circle, and override the area() method in each to calculate the area for the respective shapes. Display the areas of both shapes.

## **Banking System Using Inheritance**

- Create a base class BankAccount with attributes account\_number, name, and balance. Add methods for deposit and withdrawal.
- Create a derived class SavingsAccount that limits the number of withdrawals to 3 per month.
- Create another derived class CurrentAccount that deducts a maintenance fee for low balances.
- And Log the exception to file.
- Write a menu-driven program to:
  - Create accounts.
  - Perform deposits and withdrawals.
  - Display account details.
  - Handle invalid operations using try-except blocks