📝

Test

Python Polymorphism & Scope - Test

Part A: Multiple Choice Questions (MCQs)

Choose the correct answer for each question:

1. What is polymorphism in Python?

   a) A way to overload functions

   b) The ability to use a common interface for different data types

   c) Inheriting from multiple classes

   d) Encapsulation of methods

2. Which of the following best illustrates polymorphism?

   a) Overloading the constructor method

   b) Inheriting a private attribute

   c) Having the same method name in different classes

   d) Using static variables

3. What will be the output of the following code?

   ```
   class Cat:
       def sound(self):
           print("Meow")

   class Dog:
       def sound(self):
           print("Bark")

   def make_sound(animal):
   ```

```
    animal.sound()

make_sound(Cat())
make_sound(Dog())
```

a) Meow Bark

b) Bark Meow

c) Error

d) None

4. Which scope does a variable declared inside a function belong to?

   a) Global scope

   b) Module scope

   c) Local scope

   d) Class scope

5. What keyword is used to refer to a global variable inside a function?

   a) extern

   b) static

   c) global

   d) outer

Part B: Practical Questions

1. **Write two classes `Bird` and `Fish`, each with a method `move()` that prints "Fly in the sky" and "Swim in the water" respectively. Write a function `describe_movement()` that takes an object and calls its `move()` method to demonstrate polymorphism.**

2. **Create a base class `Shape` with a method `draw()`. Derive two classes `Circle` and `Rectangle` that override `draw()` with different messages. Create a list of shape objects and call `draw()` in a loop to demonstrate polymorphism.**

3. **Demonstrate local and global scope with a variable `x`. Create a function that defines `x = 10` locally and prints it, while also printing the global `x = 5`. Show**

the difference in their values.

4. **Write a function that modifies a global variable inside the function using the `global` keyword. Show before and after values.**
5. **Demonstrate how method overriding in polymorphism works by defining a base class `Animal` with a method `speak()`, and two derived classes `Cow` and `Lion` that override this method. Call the method using each object.**