**Scenario-Based Questions for Python and OOP Concepts**

Q.No.1:- You are working on a Library Management System where books are represented as objects. How would you create a Book class with attributes like title, author, and available (defaulting to True)? Can you implement methods to borrow and return a book and demonstrate borrowing and returning one using two book objects?

Q.No.2:- In a Student Management System, all students belong to the same university. How would you design a Student class with instance variables for name and age, along with a class variable university set to a default value like "ABC University"? Can you create a method that updates the university name for all students and show how changing this class variable affects all instances?

Q.No.3:- Your company needs a way to track how many employees have been created. How would you design an Employee class with attributes for name and salary, while also maintaining a class variable employee\_count to keep track of total employees? Can you implement a class method total\_employees() that returns the count and demonstrate this by creating multiple employee objects?

Q.No.4:- Imagine you are building a Math Utility class that provides common calculations. How would you implement a MathOperations class with a static method add\_numbers(a, b) that returns their sum? Can you call this method using both the class name and an object of the class?

Q.No.5:- In a Vehicle Rental System, different types of vehicles share common characteristics but also have unique features. How would you design a Vehicle class with brand and model attributes and a method display\_info()? Can you create a Car class that inherits from Vehicle and adds number\_of\_doors, and a SportsCar class that further inherits from Car and introduces a top\_speed attribute? How would you demonstrate this multilevel inheritance by creating a SportsCar object?

Q.No.6:- For a secure Banking System, account details must be protected. How would you create a BankAccount class with a private variable \_\_balance and implement deposit(amount), withdraw(amount), and get\_balance() methods? Can you ensure that withdrawals do not exceed the available balance and demonstrate this through deposit and withdrawal operations?

Q.No.7:- You need to develop a message logger that tracks how many messages have been logged. How would you implement a logger() function containing an inner function log\_message() that uses a nonlocal variable to track the count? Can you call log\_message() multiple times and display the message count?

Q.No.8:- Imagine you want to measure the time taken by certain functions automatically. How would you create a decorator @timer that records and prints the execution time of a function? Can you apply this decorator to a slow\_function() that uses time.sleep() to simulate a delay?

Q.No.9:- Handling large amounts of data efficiently is important. How would you implement a generator function generate\_numbers(n) that yields numbers from 1 to n instead of storing them in memory? Can you iterate over the generator and print the numbers without using a list?

Q.No.10:- When reading files, it is crucial to ensure they are always closed properly. How would you write a function read\_file(filename) that opens a file, reads its contents, and closes it using a finally block? Can you test this function with both an existing and a non-existent file?

Q.No.11:- In a User Authentication System, you need to store user details persistently. How would you design a User class with username and password, and use the pickle module to serialize and save a user object? Can you deserialize it later and print the details?

Q.No.12:- Finally, working with images often requires reading and writing binary files. How would you write a Python script that reads an image file in binary mode and saves a copy? Can you then display both the original and copied images using matplotlib?