Practice Problems of OOP

Q14:Create base classes Athlete and Musician with appropriate attributes and methods related to their respective domains (e.g., name, sport for Athlete, name, instrument for Musician, methods like train() for Athlete and perform() for Musician). Create a derived class Performer that inherits from both Athlete and Musician using multiple inheritance.

Q15: You are tasked with designing a plugin system in Python. Plugins can have various functionalities such as data processing or networking capabilities. How would you utilise multiple inheritance to create a flexible plugin architecture where plugins can inherit from multiple base classes to combine functionalities seamlessly?

Q16: Define three classes A, B, and C. B and C should both inherit from A. Add a method identify in A that prints "I am A". Override identify in B to print "I am B" and in C to print "I am C". Define a class D that inherits from both B and C.Instantiate a D object and call the identify method.Use super() to ensure that the identify method in class A is called from within D's identify method.

Q17: Define the classes Media, Audio, Video, and AudioVideo according to the specifications.Ensure that the AudioVideo class properly resolves the diamond problem by using super(). Create an instance of AudioVideo. Call the play, stop, adjust_volume, adjust_brightness, and display_info methods on the instance.

Q18: write a program that counts the number of objects created of a particular class.

Q19: Create a class Student with a class variable school_name set to "ABC School". Add instance variables name and age. Change the class variable and instance variables and observe the effects on different instances.

Q20: Create a class Employee with a class attribute num_of_employees to keep track of the number of employees. Implement a class method increment_employee_count to increase the employee count whenever a new employee is created.

Q21: Design a class Student with class variables total_students and total_grades. Implement a class method update_statistics to increments total_students, and adds new_grade to total_grades. Test the method by creating several instances and updating the statistics.

Q22: Define a class Circle with a class method from_diameter that creates a Circle instance given the diameter.

Q23: Define a class Book with an __init__ method that takes title, author, and pages. Then, add a class method from_dict that creates a Book instance from a dictionary containing these keys.

Q24: Define a class Temperature with a class method from_fahrenheit that converts a temperature from Fahrenheit to Celsius and creates an instance.

Q25: Define a class Employee with attributes name, position, and salary. Implement a class method from_string that creates an Employee instance from a string formatted as "Name-Position-Salary".