Practice Problems of OOP

Q1: Create a Rectangle class with attributes length and width. Add methods to calculate the area and perimeter of the rectangle. Instantiate the class and call these methods.

Q2: Write a Python program to create a calculator class. Include methods for basic arithmetic operations.

Q3: Write a Python program to create a Car class with attributes like brand, model, and year. Create an object of the Car class and display its attributes.

Q4: Create a BankAccount class that allows you to deposit, withdraw, and check the balance.

Q5: Create a class Student with attributes name, roll_number, and grades. Implement a method calculate_average that calculates the average grade of the student. Create an instance of Student and call the calculate_average method.

Q6: Create a class Employee with attributes name, employee_id, and salary. Write a constructor that initializes these attributes and a method give_raise that increases the salary by a given percentage. Create an instance of Employee and test the give_raise method.

Q7:Add a static method compare_area(rect1, rect2) to the Rectangle class that compares the areas of two rectangles rect1 and rect2 and returns the larger one.

Q8: You are developing a software system for managing tasks in a project management application. Each task has a status associated with it, such as "To Do", "In Progress", or "Done". Design a decorator status_logger that logs a message whenever the status of a task changes. Assume you have a Task class.

Q9: You are tasked with developing a software system for managing books in a library. Each book can be checked out by users if it is available. Implement a decorator check_availability that restricts access to the checkout method of the Book class based on the availability of the book.

Credit: AYESHA SALEEM

Q10:Define a class Temperature with a private attribute _celsius.
Implement getter and setter methods for _celsius using the property decorator.
Implement properties Fahrenheit and kelvin to get and set the temperature in Fahrenheit and Kelvin respectively. Ensure that updating Fahrenheit or kelvin also updates _celsius.

Q11: Create a class MathOperations with a class method to calculate the factorial of a number and a static method to check if a number is prime. Demonstrate the difference between class methods and static methods in this context.

Q12: Create a class BankAccount with methods for deposit, withdrawal, and displaying balance.Create a subclass SavingsAccount that inherits from BankAccount. Override the deposit() method in SavingsAccount to add an interest bonus on deposit. Create objects of both BankAccount and SavingsAccount and test their functionalities.

Q13:An e-commerce platform requires a product categorisation system:

1. Product: The base class with attributes like name and price.
2. Electronics: Inherits from Product, adding warranty period and brand.
3. Smartphone: Inherits from Electronics, adding operating system and camera quality. Define the three classes with appropriate attributes and methods. Implement a method in each class to display product details. Create a Smartphone object and demonstrate the use of methods from all three classes.

Credit: AYESHA SALEEM