Coding Challenges - OOPs

Task 31

You are working on a project to develop an e-commerce website. As part of the project, you need to implement a shopping cart functionality using OOP concepts in Python. Design and implement the classes for the shopping cart system with the following requirements:

- The shopping cart should be able to add products, remove products, and calculate the total price of all the products in the cart.
- Each product should have a name, price, and quantity.
- The shopping cart should be able to handle multiple instances of the same product and update the quantity accordingly.
- Design the classes and provide the python code implementation for the shopping cart system.

Task 32

You are working on a project to develop a class representing a Book. The Book class should have properties such as title, author, and year of publication. Implement the constructor for the Book class that initializes these properties when a new Book object is created. Additionally, provide a method called **getBookInfo()** that returns a string with the book's details in the format "Title: [title], Author: [author], Year: [year]".

Write the constructor and the **getBookInfo()** method for the Book class.

Task 33

You are developing a system to handle different shapes. Implement an abstract class called **Shape** with an abstract method **calculateArea()**. The **calculateArea()** method should be implemented by the concrete classes that inherit from Shape and should return the area of the specific shape.

Create two concrete classes **Circle** and **Rectangle** that extend the **Shape** class. Implement the **calculateArea()** method in both classes according to the area calculation formulas for circles and rectangles. Display the areas of a circle and a rectangle object.

Write the abstract class **Shape**, the concrete classes Circle and **Rectangle**, and the code to display the areas.

Coding Challenges - OOPs

Task 34

You are developing a class called **MathUtils** that provides various mathematical utility functions. Implement a static method called **calculateSum()** in the **MathUtils** class. The **calculateSum()** method should accept an array of numbers and return the sum of those numbers.

Write the **MathUtils** class with the static **calculateSum()** method and provide code to test the functionality.

Task 35

You are developing a system to handle different payment methods. Implement an abstract class called **PaymentMethod** with an abstract method **processPayment().** The **processPayment()** method should be implemented by the concrete classes that inherit from **PaymentMethod** and should simulate the payment processing for each specific payment method.

Create two concrete classes **CreditCardPayment** and **PayPalPayment** that extend the **PaymentMethod** class. Implement the **processPayment()** method in both classes to display a message indicating the payment method being processed.

Write the abstract class **PaymentMethod**, the concrete classes **CreditCardPayment** and **PayPalPayment**, and the code to demonstrate polymorphism.