Technogeeks
Number of problem statements : 08
Python concepts required : Data types , if-else , loop , functions and methods , object oriented programming
Note: This assignment covers multiple domains based on dummy problem statements including Banking , Retail, Automotive , Telecom , Insurance , healthcare.

Problem statement 01:

Problem: Create a class called "BankAccount" that represents a bank account. The class should have the following features:

- A constructor method that takes the account holder's name and initial balance as parameters and initializes them.
- A method called "deposit" that takes an amount as a parameter and adds it to the account balance.
- A method called "withdraw" that takes an amount as a parameter and subtracts it from the account balance, if the account has sufficient funds.
- A method called "get_balance" that returns the current balance of the account.
- A method called "display_account_info" that prints the account holder's name and current balance.

Write the BankAccount class and demonstrate its usage by creating multiple instances of the class, performing various transactions, and displaying the account information.

Problem statement 02:

Create a class called "Circle" that represents a circle object. The class should have the following features:

- A constructor method that takes the radius as a parameter and initializes it.
- A method called "get_area" that calculates and returns the area of the circle.
- A method called "get_circumference" that calculates and returns the circumference of the circle.

Write the Circle class and demonstrate its usage by creating an instance of the class, setting a radius, and printing the area and circumference of the circle.

Problem statement 03:

Create a class called "PetrolPump" that represents a petrol pump. The class should have the following features:

- A constructor method that initializes the petrol pump with an initial quantity of petrol.
- A method called "fill_petrol" that takes the number of liters as a parameter and adds it to the existing quantity of petrol.
- A method called "dispense_petrol" that takes the number of liters as a parameter and dispenses that amount of petrol from the pump, if there is enough petrol available.
- A method called "get_petrol_quantity" that returns the current quantity of petrol in the pump.

Write the PetrolPump class and demonstrate its usage by creating an instance of the class, performing petrol filling and dispensing operations, and displaying the current quantity of petrol in the pump.

Problem statement 04:

Create a class called "InsurancePolicy" that represents an insurance policy in an insurance company.
The class should have the following features:

- A constructor method that takes the policy holder's name, policy number, and premium amount as parameters and initializes them.
- A method called "calculate_premium" that calculates and returns the premium amount for the policy based on the policy type and other factors.
- A method called "display_policy_info" that prints the policy holder's name, policy number, and premium amount.

Write the InsurancePolicy class and demonstrate its usage by creating an instance of the class, calculating the premium amount, and displaying the policy information.

Problem statement 05:

Create a class called "MobilePlan" that represents a mobile plan offered by a telecom company.
The class should have the following features:

- A constructor method that takes the plan name, data allowance, and monthly cost as parameters and initializes them.
- A method called "check_data_usage" that takes the amount of data used by a customer as a parameter and determines if it exceeds the data allowance of the plan.
- A method called "display_plan_details" that prints the plan name, data allowance, and monthly cost.

Write the MobilePlan class and demonstrate its usage by creating an instance of the class, checking data usage against the plan allowance, and displaying the plan details.

Problem statement 06:

Create a class called "Product" that represents a retail product. The class should have the following features:

- A constructor method that takes the product name, price, and quantity in stock as parameters and initializes them.
- A method called "calculate_total_cost" that calculates and returns the total cost of a specified quantity of the product, taking into account any discounts or promotions.
- A method called "display_product_details" that prints the product name, price, and quantity in stock.

Write the Product class and demonstrate its usage by creating an instance of the class, calculating the total cost for a specific quantity, and displaying the product details.

Problem statement 07:

Create a class called "Patient" that represents a patient in a healthcare system. The class should have the following features:

- A constructor method that takes the patient's name, age, and medical condition as parameters and initializes them.
- A method called "schedule_appointment" that takes the name of a doctor and a date as parameters and schedules an appointment for the patient with the specified doctor on the given date.
- A method called "display_patient_info" that prints the patient's name, age, and medical condition.

Write the Patient class and demonstrate its usage by creating an instance of the class, scheduling an appointment, and displaying the patient's information.

Problem statement 08:

Create a class called "Car" that represents a car in an automotive system. The class should have the following features:

- A constructor method that takes the car's make, model, and year as parameters and initializes them.
- A method called "start_engine" that starts the car's engine and prints a message indicating that the engine has started.
- A method called "accelerate" that takes a speed as a parameter and simulates the car accelerating to that speed.
- A method called "display_car_info" that prints the car's make, model, and year.

Write the Car class and demonstrate its usage by creating an instance of the class, starting the engine, accelerating the car, and displaying the car's information.