1. Data Abstraction, Encapsulation, Inheritance, Polymorphism, Generalization, Composition, and Aggregation.

Problem:

You are tasked with designing a simple car rental system. The system should be able to handle different types of vehicles. The system should also be able to handle different rental rates based on the type of vehicle, the rental duration, and any additional services that the customer may request.

Requirements:

- The system should allow customers to rent different types of vehicles.
- The system should track the rental duration and calculate the total rental cost based on the type of vehicle and any additional services requested.
- The system should allow customers to return their rented vehicles.
- The system should be able to generate a report of all the rented vehicles and their respective rental costs.

Design:

- Create an abstract class called Vehicle that defines the basic properties of a vehicle, such as the make, model, year, and rental rate. This class should also include methods for setting and getting these properties. Define an abstract method getType() which is not implemented in this class but must be implemented in any concrete subclass of Vehicle, leaving the details of the implementation to each subclass (**abstraction**).
- Create subclasses for each type of vehicle, such as Car, SUV. Each subclass should inherit the properties and methods of the Vehicle class (**generalization, inheritance**) and should also include any additional properties and methods specific to that type of vehicle (Car - numDoors, fuelType; SUV - numSeats, fourWheelDrive).
- Use **encapsulation** to protect the internal state of the Vehicle and its subclasses. This can be achieved by making the properties private and using public methods for accessing and modifying them.
- Create a class called Rental that represents a rental transaction. This class should contain a reference to the rented vehicle (**composition, polymorphism**), the rental start date, the rental duration, and any additional services requested by the customer (GPS, ChildSeat, RoofRack etc). Define a function, calculateRentalCost(), which calculates total cost using rental rate, duration, and additional services if any.
- Create a class called RentalSystem containing a list of all the rented vehicles (**aggregation**) and should include methods for renting a vehicle, returning a vehicle, and generating a report of all the rented vehicles, their respective rental costs and total revenue.

2. Enum, loops and data types

Problem:

Write a program to calculate the average temperature of a city for a given month.

Requirements:

- The program should prompt the user to enter the name of the city and the month for which they want to calculate the average temperature.
- The program should then prompt the user to enter the high temperature and low temperature for each day of the month. The user should be able to input temperatures in Fahrenheit or Celsius. It should accept symbols as well (C or F).
- The program should convert all temperatures to Celsius for calculations.
- The program should then calculate the average temperature for the month and display it to the user in both Fahrenheit and Celsius.

3. String handling

Problem:

Suppose you have a string that represents a phone number in the format "(XXX) XXX-XXXX", where each X represents a digit from 0 to 9. Write a program that does the following:

- Asks the user to enter a phone number in the correct format.
- Check if the phone number is in the correct format. If it is not, display an error message and ask the user to enter a phone number again.
- If the phone number is in the correct format, remove the parentheses, space, and dash from the string and display the resulting phone number in the format "XXX-XXX-XXXX".
- Count the number of digits in the phone number and display the result.
- Convert the phone number to an array of integers where each element represents a digit. Display the resulting array.

4. Exception handling

Problem:

You have been tasked with designing a program that will calculate the average of an array of integers. However, the program must be able to handle the following exceptions:

- An empty array
- An array with a negative integer value

Your program should prompt the user to enter the size of the array, and then prompt them to enter the values of the array. If any of the above exceptions occur, your program should catch them and output an appropriate error message. Otherwise, your program should calculate and output the average of the values in the array.

5. Additional string handling problems

1. Write a program that reads in a sentence from the user and prints out the number of words in the sentence.
2. Given a string, write a program that determines whether the string is a palindrome (reads the same forwards and backwards).
3. Write a program that takes in two strings and checks if they are anagrams of each other (contain the same letters in the same quantities).
4. Write a program that reads in a string and replaces all occurrences of a given substring with a different substring.
5. Given a string, write a program that returns the longest substring of the string that does not contain any repeating characters.
6. Write a program that reads in a string and reverses the order of the words in the string.
7. Given two strings, write a program that determines whether the first string is a substring of the second string.
8. Write a program that reads in a string and converts all uppercase letters to lowercase and all lowercase letters to uppercase.
9. Given a string, write a program that finds and prints all the substrings of the string that are palindromes.
10. Write a program that reads in two strings and returns the length of the longest common substring (i.e., the longest substring that appears in both strings).