Assignments

- Create an extension method ToTitleCase() for the string class that converts the first letter of each word to uppercase.
- Create an extension method AverageExceptZero() for List<int> that calculates the average excluding zero values.

Sample Date :
List<int> numbers = new List<int> { 10, 0, 20, 30, 0 };
Console.WriteLine(numbers.AverageExceptZero()); // Output : 20

- Create a base class Animal with a virtual method Speak(). Then create two derived classes: Dog and Cat, that override the Speak() method.
- Create a base class Vehicle with a method ShowType() that prints "This is a vehicle". Then create a derived class Car that hides the ShowType() method using the new keyword and prints "This is a car".
  Call the method using both base class and derived class references to see the difference.

- Banking App
  - Create an interface IAccount with methods:
    - void Deposit(double amount);
    - void Withdraw(double amount);
    - double GetBalance();
  - Then implement it in two classes:
    - SavingsAccount
    - CurrentAccount
  - Create an interface IPaymentService with method:
    - void MakePayment(double amount);
  - Then create implementations:
    - CreditCardPayment
    - UPIPayment
    - NetBankingPayment
  - Create a class PaymentProcessor that depends only on the interface, not on the concrete class. Use Constructor for injecting the dependency.
- Notification App
  - Create interface INotificationService with:
    - void Notify(string message);
  - Implement SMSNotifier, EmailNotifier, and PushNotifier.
  - Create a class AppointmentService that depends on the interface to notify patients after booking.