Set D Solution

1. **Write a program to create a class User with attributes Name, Age, Email, and City. Create a constructor to initialize these attributes and methods to update the user's details. Implement functionality to input the details of 5 users and display the details of the user with the highest age.**

```
using System;
class User
{
   public string Name;
   public int Age ;
   public string Email;
   public string City;

   public User(string name, int age, string email, string city)
   {
      Name = name;
      Age = age;
      Email = email;
      City = city;
   }

   public void UpdateDetails(string name, int age, string email, string city)
   {
      Name = name;
      Age = age;
      Email = email;
      City = city;
   }
}
class Program
{
   static void Main(string[] args)
   {
      User[] users = new User[5];
      for (int i = 0; i < 5; i++)
      {
         Console.WriteLine("Enter details for user " + (i + 1));
         Console.Write("Name: ");
         string name = Console.ReadLine();
         Console.Write("Age: ");
         int age = int.Parse(Console.ReadLine());
         Console.Write("Email: ");
         string email = Console.ReadLine();
         Console.Write("City: ");
         string city = Console.ReadLine();
         users[i] = new User(name, age, email, city);
      }

      User oldestUser = users[0];
      foreach (var user in users)
```

```
        {
            if (user.Age > oldestUser.Age)
            {
                oldestUser = user;
            }
        }

        Console.WriteLine($"User with highest age: {oldestUser.Name}, Age: {oldestUser.Age}");
    }
}
```

2. Write a program in C# to check whether a given number is an Armstrong number or not.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Console.Write("Enter a number to check if it's an Armstrong number: ");
        int number = int.Parse(Console.ReadLine());
        int temp = number, sum = 0, digits = number.ToString().Length;
        while (temp != 0)
        {
            int digit = temp % 10;
            sum += (int)Math.Pow(digit, digits);
            temp /= 10;
        }
        if (sum == number)
            Console.WriteLine($"{number} is an Armstrong number.");
        else
            Console.WriteLine($"{number} is not an Armstrong number.");
    }
}
```

3. Create a program to manage student grades. The program should:

1. **Accept marks for three subjects .**
2. **Throw a custom exception InvalidMarksException if the user enters marks outside the range of 0 to 100.**
3. **Calculate the total and percentage of marks only if all inputs are valid.**

```
using System;
class InvalidMarksException : Exception { }
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Enter marks for three subjects:");
```

```
            int[] marks = new int[3];
            int total = 0;
            for (int i = 0; i < 3; i++)
            {
                Console.Write($"Subject {i + 1} marks: ");
                marks[i] = int.Parse(Console.ReadLine());
                if (marks[i] < 0 || marks[i] > 100)
                    throw new InvalidMarksException();
                total += marks[i];
            }
            double percentage = (double)total / 3;
            Console.WriteLine($"Total Marks: {total}, Percentage: {percentage}%");
        }
        catch (InvalidMarksException)
        {
            Console.WriteLine("Error: Invalid marks entered. Marks must be between 0 and 100.");
        }
    }
}
```

4. Write a program to accept a paragraph of text from the user and perform the following operations:

1. Find and display the longest word.

2. Replace all occurrences of a specific word with another word.

Accept another word from user.

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.Write("Enter a paragraph: ");
        string paragraph = Console.ReadLine();
        string[] words = paragraph.Split(' ');

        string longestWord = "";
        foreach (var word in words)
        {
            if (word.Length > longestWord.Length)
                longestWord = word;
        }
        Console.WriteLine($"Longest word: {longestWord}");

        Console.Write("Enter a word to replace: ");
        string wordToReplace = Console.ReadLine();
        Console.Write("Enter the new word: ");
        string replacementWord = Console.ReadLine();
```

```
        string updatedParagraph = paragraph.Replace(wordToReplace, replacementWord);
        Console.WriteLine("Updated paragraph: " + updatedParagraph);
    }
}
```

5. Write a program to calculate the area of different shapes using method overloading:

1. **Circle (requires radius).**
2. **Rectangle (requires length and width).**
3. **Triangle (requires base and height).**

```
using System;
class Shape
{
    public double Area(double radius) => Math.PI * radius * radius;
    public double Area(double length, double width) => length * width;
    public double Area(double baseLength, double height) => 0.5 * baseLength * height;
}
class Program
{
    static void Main(string[] args)
    {
        Shape shape = new Shape();
        Console.WriteLine("Area of Circle (radius 5): " + shape.Area(5));
        Console.WriteLine("Area of Rectangle (length 5, width 10): " + shape.Area(5, 10));
        Console.WriteLine("Area of Triangle (base 5, height 10): " + shape.Area(5, 10));
    }
}
```

6. Create a program that demonstrates the use of a copy constructor with a Time object. The Time class should store time in hours, minutes, and seconds, and provide functionality for adding times.

```
using System;
class Time
{
    public int Hours ;
    public int Minutes ;
    public int Seconds ;

    public Time(int hours, int minutes, int seconds)
    {
        Hours = hours;
        Minutes = minutes;
        Seconds = seconds;
    }

    public Time(Time time)
    {
        Hours = time.Hours;
        Minutes = time.Minutes;
```

```
        Seconds = time.Seconds;
    }

    public void AddTime(Time time)
    {
        this.Seconds += time.Seconds;
        if (this.Seconds >= 60)
        {
            this.Seconds -= 60;
            this.Minutes++;
        }
        this.Minutes += time.Minutes;
        if (this.Minutes >= 60)
        {
            this.Minutes -= 60;
            this.Hours++;
        }
        this.Hours += time.Hours;
    }

    public void DisplayTime()
    {
        Console.WriteLine($"Time: {Hours}:{Minutes}:{Seconds}");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Time time1 = new Time(2, 50, 30);
        Time time2 = new Time(1, 20, 45);
        Time time3 = new Time(time1); // Copy constructor
        time3.AddTime(time2);
        time3.DisplayTime();
    }
}
```

7. Write a program to use a Stack to reverse a sentence entered by the user.

```
using System;
using System.Collections.Generic;
class Program
{
    static void Main(string[] args)
    {
        Stack<string> stack = new Stack<string>();
        Console.Write("Enter a sentence: ");
        string sentence = Console.ReadLine();
        string[] wordsInSentence = sentence.Split(' ');

        foreach (var word in wordsInSentence)
        {
```

```
            stack.Push(word);
        }

        Console.WriteLine("Reversed Sentence:");
        while (stack.Count > 0)
        {
            Console.Write(stack.Pop() + " ");
        }
    }
}
```