

31-mar-23

Problem:1

using System;

```
class Program
{
    static void Main()
    {
        Console.Write("Enter a string: ");
        string input = Console.ReadLine();

        string reversed = ReverseString(input);
        Console.WriteLine("Reversed string: " + reversed);
    }

    static string ReverseString(string str)
    {
        char[] charArray = str.ToCharArray();
        Array.Reverse(charArray);
        return new string(charArray);
    }
}
```

Problem:2

```
public static class StringExtensions
{
    public static int CountVowels(this string str)
    {
        // Convert the string to lowercase to handle both uppercase and lowercase vowels
        string lowercaseStr = str.ToLower();

        int vowelCount = 0;

        foreach (char c in lowercaseStr)
        {
            if (IsVowel(c))
            {
                vowelCount++;
            }
        }

        return vowelCount;
    }

    private static bool IsVowel(char c)
    {

```

```

        // Check if the character is a vowel
        return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
    }
}

```

Problem 3

```

using System;
using System.Collections.Generic;

class Program
{
    static void Main(string[] args)
    {
        List<string> names = new List<string> { "RAJ", "raj", "Avi", "Kirti" };

        Dictionary<string, int> nameCount = CountNames(names);

        foreach (KeyValuePair<string, int> pair in nameCount)
        {
            Console.WriteLine($"{pair.Key} - {pair.Value}");
        }
    }

    static Dictionary<string, int> CountNames(List<string> names)
    {
        Dictionary<string, int> nameCount = new Dictionary<string,
int>(StringComparer.OrdinalIgnoreCase);

        foreach (string name in names)
        {
            if (nameCount.ContainsKey(name))
            {
                nameCount[name]++;
            }
            else
            {
                nameCount[name] = 1;
            }
        }

        return nameCount;
    }
}

```

Problem 4

```

using System;

```

```

class AppMath
{
    public static int Add(params int[] numbers)
    {
        int sum = 0;
        foreach (int num in numbers)
        {
            sum += num;
        }
        return sum;
    }

    public static double Add(params double[] numbers)
    {
        double sum = 0;
        foreach (double num in numbers)
        {
            sum += num;
        }
        return sum;
    }

    public static decimal Add(params decimal[] numbers)
    {
        decimal sum = 0;
        foreach (decimal num in numbers)
        {
            sum += num;
        }
        return sum;
    }

    public static int Multiply(params int[] numbers)
    {
        int result = 1;
        foreach (int num in numbers)
        {
            result *= num;
        }
        return result;
    }

    public static double Multiply(params double[] numbers)
    {
        double result = 1;
        foreach (double num in numbers)
        {

```

```

        result *= num;
    }
    return result;
}

public static decimal Multiply(params decimal[] numbers)
{
    decimal result = 1;
    foreach (decimal num in numbers)
    {
        result *= num;
    }
    return result;
}
}

```

Assignment 3-Apr-2023

Problem 1

```

using System;

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

        Console.WriteLine("The string you entered is: " + input);
    }
}

```

Problem 2

```

using System;

class Program
{
    static void Main(string[] args)
    {
        Console.Write("Input the string: ");
        string input = Console.ReadLine();

        int length = 0;
        foreach (char c in input)
        {
            length++;
        }
    }
}

```

```

        Console.WriteLine("Length of the string is: " + length);
    }
}

```

Problem 3

using System;

class Program

```

{
    static void Main(string[] args)
    {
        Console.Write("Input the string: ");
        string input = Console.ReadLine();

        Console.Write("The characters of the string are: ");
        foreach (char c in input)
        {
            Console.Write(c + " ");
        }
        Console.WriteLine();
    }
}

```

Problem 4

using System;

class Program

```

{
    static void Main(string[] args)
    {
        Console.Write("Input the string: ");
        string input = Console.ReadLine();

        Console.Write("The characters of the string in reverse are: ");
        for (int i = input.Length - 1; i >= 0; i--)
        {
            Console.Write(input[i] + " ");
        }
        Console.WriteLine();
    }
}

```

Problem 5

using System;

class Program

```

{
    static void Main(string[] args)
    {
        Console.Write("Input the string: ");
        string input = Console.ReadLine();

        int alphabetCount = 0;
        int digitCount = 0;
        int specialCharCount = 0;

        foreach (char c in input)
        {
            if (Char.IsLetter(c))
            {
                alphabetCount++;
            }
            else if (Char.IsDigit(c))
            {
                digitCount++;
            }
            else
            {
                specialCharCount++;
            }
        }

        Console.WriteLine("Number of Alphabets in the string is: " + alphabetCount);
        Console.WriteLine("Number of Digits in the string is: " + digitCount);
        Console.WriteLine("Number of Special characters in the string is: " + specialCharCount);
    }
}

```

Assignment 10-Apr-2023

(Anmol) Problem 1:

using System;

using System.IO;

class Program

```

{
    static void Main()
    {
        // File creation and writing
        string fileName = "example.txt";
        string content = "Hello, World!";

        // Create and write to the file
        File.WriteAllText(fileName, content);
    }
}

```

```

// File opening and reading
string fileContent = File.ReadAllText(fileName);
Console.WriteLine("File content: " + fileContent);

// Appending a line to a text file
string newLine = "This is a new line.";
AppendLineToFile(fileName, newLine);

// File reopening and reading after appending
fileContent = File.ReadAllText(fileName);
Console.WriteLine("Updated file content: " + fileContent);
}

static void AppendLineToFile(string fileName, string line)
{
    // Open the file in append mode and write the new line
    using (StreamWriter file = File.AppendText(fileName))
    {
        file.WriteLine(line);
    }
}
}

```

Assignment 12-Apr-2023

Problem 1

using System;

```

class Program
{
    static void Main()
    {
        Console.WriteLine("Simple Calculator");

        Console.Write("Enter the first number: ");
        double num1 = Convert.ToDouble(Console.ReadLine());

        Console.Write("Enter the second number: ");
        double num2 = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Select an operation:");
        Console.WriteLine("1. Addition (+)");
        Console.WriteLine("2. Subtraction (-)");
        Console.WriteLine("3. Multiplication (*)");
        Console.WriteLine("4. Division (/)");
        Console.WriteLine("5. Modulus (%)");

        Console.Write("Enter your choice (1-5): ");
        int choice = Convert.ToInt32(Console.ReadLine());
    }
}

```

```

double result = 0;

switch (choice)
{
    case 1:
        result = num1 + num2;
        Console.WriteLine($"Result: {num1} + {num2} = {result}");
        break;
    case 2:
        result = num1 - num2;
        Console.WriteLine($"Result: {num1} - {num2} = {result}");
        break;
    case 3:
        result = num1 * num2;
        Console.WriteLine($"Result: {num1} * {num2} = {result}");
        break;
    case 4:
        if (num2 != 0)
        {
            result = num1 / num2;
            Console.WriteLine($"Result: {num1} / {num2} = {result}");
        }
        else
        {
            Console.WriteLine("Error: Division by zero is not allowed.");
        }
        break;
    case 5:
        result = num1 % num2;
        Console.WriteLine($"Result: {num1} % {num2} = {result}");
        break;
    default:
        Console.WriteLine("Invalid choice.");
        break;
}
}
}

```

Assignment 24-Apr-2023(Anmol)

Problem 1

using System;

abstract class Employee

```

{
    public string Name { get; set; }

    public abstract double CalculateSalary();
}

```



```

}

class Developer : Employee
{
    public override double CalculateSalary()
    {
        return 5000;
    }
}

class Driver : Employee
{
    public override double CalculateSalary()
    {
        return 3000;
    }
}

class Program
{
    static void Main()
    {
        Developer developer = new Developer();
        developer.Name = "John";
        double developerSalary = developer.CalculateSalary();
        Console.WriteLine($"Salary of developer {developer.Name}: {developerSalary}");

        Driver driver = new Driver();
        driver.Name = "Mike";
        double driverSalary = driver.CalculateSalary();
        Console.WriteLine($"Salary of driver {driver.Name}: {driverSalary}");
    }
}

```

Problem 2
using System;

```

interface IShape
{
    double CalculateArea();
    double CalculatePerimeter();
}

class Rectangle : IShape
{
    public double Length { get; set; }
    public double Width { get; set; }
}

```

```

    public double CalculateArea()
    {
        return Length * Width;
    }

    public double CalculatePerimeter()
    {
        return 2 * (Length + Width);
    }
}

class Square : IShape
{
    public double Side { get; set; }

    public double CalculateArea()
    {
        return Side * Side;
    }

    public double CalculatePerimeter()
    {
        return 4 * Side;
    }
}

class Program
{
    static void Main()
    {
        Rectangle rectangle = new Rectangle();
        rectangle.Length = 7;
        rectangle.Width = 4;
        double rectangleArea = rectangle.CalculateArea();
        double rectanglePerimeter = rectangle.CalculatePerimeter();
        Console.WriteLine($"Rectangle - Area: {rectangleArea}, Perimeter: {rectanglePerimeter}");

        Square square = new Square();
        square.Side = 4;
        double squareArea = square.CalculateArea();
        double squarePerimeter = square.CalculatePerimeter();
        Console.WriteLine($"Square - Area: {squareArea}, Perimeter: {squarePerimeter}");
    }
}

```

Problem 3
using System;

```

class Program
{
    static void Main()
    {
        int rows = 5;

        for (int i = 1; i <= rows; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                Console.Write(j + " ");
            }

            Console.WriteLine();
        }
    }
}

```

Problem 4
using System;

```

class Program
{
    static void Main()
    {
        int outerIterations = 5;
        int innerIterations = 2;

        int number = 1;

        for (int i = 1; i <= outerIterations; i++)
        {
            for (int j = 1; j <= innerIterations; j++)
            {
                for (int k = 1; k <= i; k++)
                {
                    Console.Write(number + " ");
                }
                number++;
            }
        }

        Console.WriteLine();
    }
}

```

Assignment 12-May-2023 (Anmol)
Problem 1:

```

using System;

// Problem 1: Singlecast Delegate
delegate void SinglecastDelegate(string message);

// Problem 2: Multicast Delegate
delegate void MulticastDelegate(string message);

// Problem 3: Anonymous Delegate

class Program
{
    static void Main()
    {
        // Problem 1: Singlecast Delegate
        SinglecastDelegate singlecastDelegate = DisplayMessage;
        singlecastDelegate("Hello, Singlecast Delegate!");

        Console.WriteLine();

        // Problem 2: Multicast Delegate
        MulticastDelegate multicastDelegate = null;
        multicastDelegate += DisplayMessage;
        multicastDelegate += DisplayAnotherMessage;
        multicastDelegate("Hello, Multicast Delegate!");

        Console.WriteLine();

        // Problem 3: Anonymous Delegate
        Action<string> anonymousDelegate = delegate (string message)
        {
            Console.WriteLine("Hello, Anonymous Delegate!");
            Console.WriteLine("Message: " + message);
        };

        anonymousDelegate("Hello");
    }

    static void DisplayMessage(string message)
    {
        Console.WriteLine("Hello, Singlecast Delegate!");
        Console.WriteLine("Message: " + message);
    }

    static void DisplayAnotherMessage(string message)
    {
        Console.WriteLine("Hello, Another Message!");
    }
}

```

```
        Console.WriteLine("Message: " + message);  
    }  
}
```