2100090159
M. Dinesh Gopi

PRE-LAB

1.What are the Advantages of Extension Methods?

Solution:
1. Enhanced Readability: Extension methods improve the readability of code by allowing developers to call methods as if they were part of the original class, even though they are defined externally.

2. No Modification of Existing Code: They allow developers to add new functionality to existing types without modifying the original code, making it easier to maintain and update codebases.

3. Code Reusability: Extension methods promote code reusability by allowing the same method to be used across multiple classes without duplicating code.

4.Integration with LINQ: Extension methods integrate seamlessly with LINQ (Language Integrated Query), enabling developers to create custom query operators that can be used with LINQ queries.

5. Support for Framework Classes:They can be used to add functionality to classes that are part of the .NET Framework or other third-party libraries, which is especially useful when developers do not have access to the source code of these classes.

IN-LAB:

Task1:Develop a MyExtension **class, which declares the following extension methods:**

- the **SummaDigit** method, which extends the Int32 type and returns the sum of the digits of an arbitrary integer.

```
Example 1: n = 1274   result = 14 (14 = 1 + 2 + 7 + 4)
```

- the **SummaWithReverse** method, which extends the UInt32 type and returns the sum of the original positive integer with the number obtained from the original by rearranging all digits in reverse order

```
Example 2: n = 132    result = 363 (363 = 132 + 231)
```

- the **CountNotLetter** method, which extends the String type and returns the number of characters in the string that are not Latin letters.

```
Example 3: s = "I like C#"   result = 3 (there are two spaces
and a "sharp" character in the line)
```

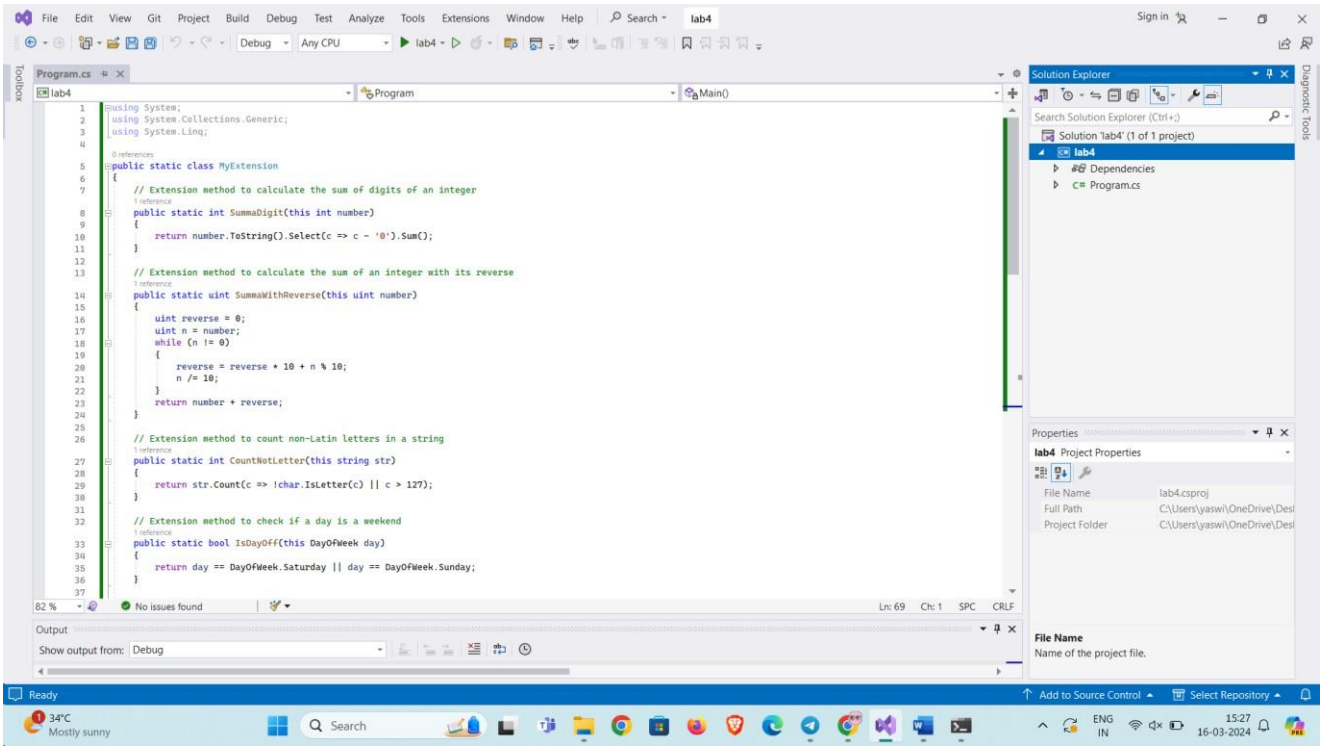
- the **IsDayOff** method, which extends the DayOfWeek type and returns the boolean value true if it is a weekend (Saturday or Sunday) or the boolean value false if it is a weekday.

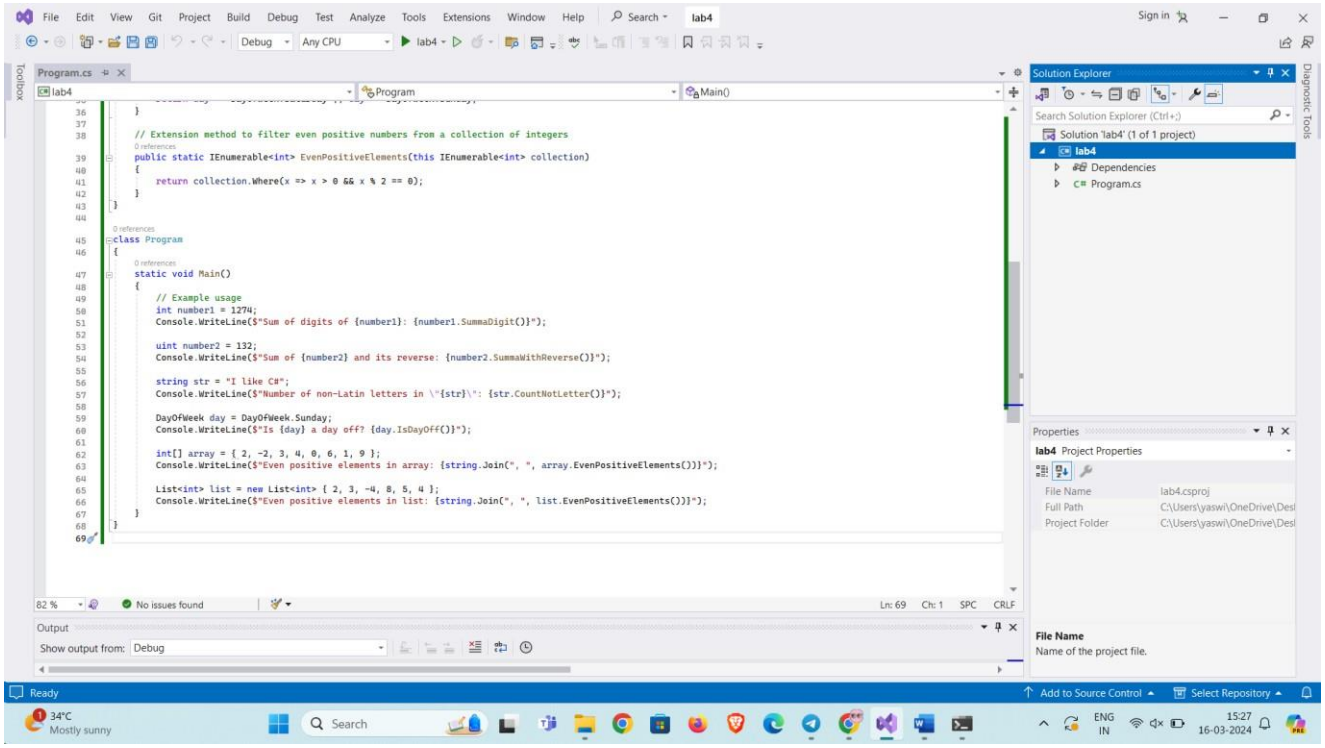
```
Example 4: day = DayOfWeek.Sunday  result = true
```

- the **EvenPositiveElements** method, which extends the IEnumerable <int> type and returns only even positive numbers from a set of integers

```
Example 5: int[] mas = { 2, -2, 3, 4, 0, 6, 1, 9 }     result = 2 , 4, 6
Example 6: for List<int> list = new List<int>{ 2, 3, -4, 8, 5, 4 }    result = 2, 8, 4
```

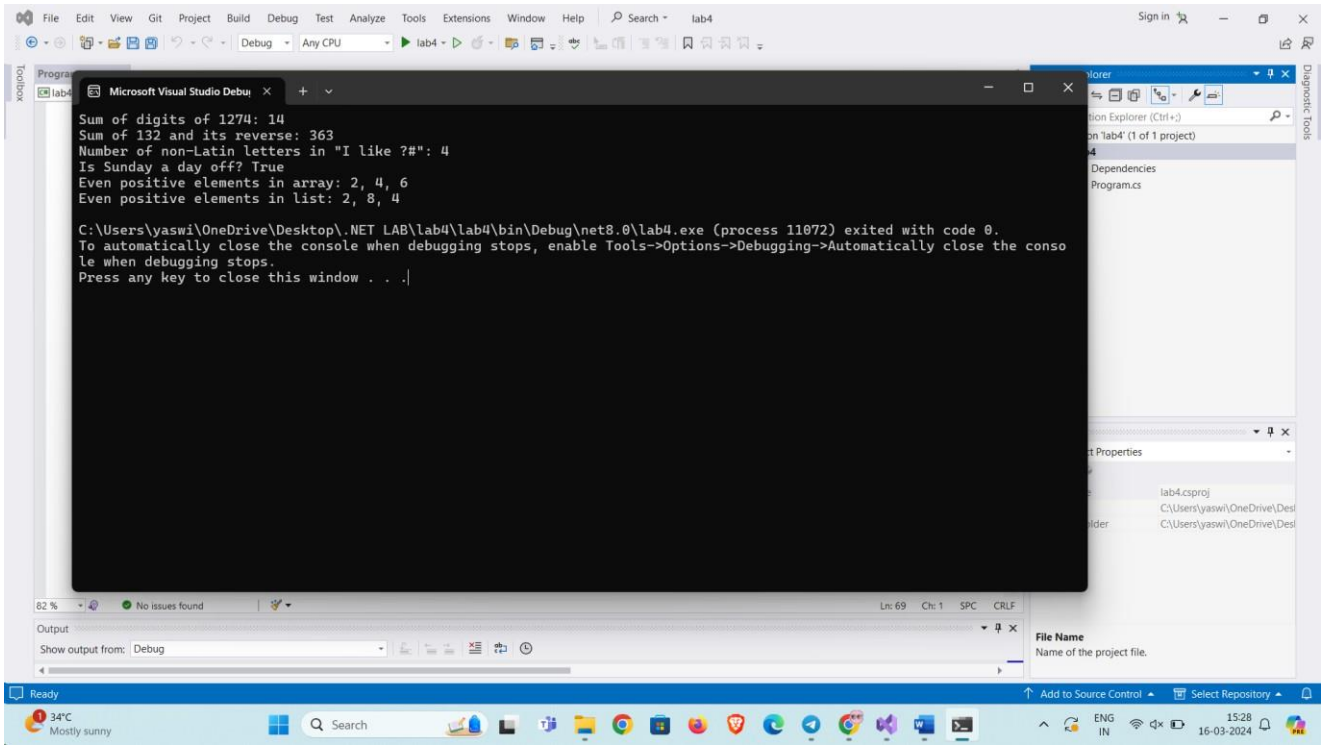
Solution:





```
    }

    // Extension method to filter even positive numbers from a collection of integers
    public static IEnumerable<int> EvenPositiveElements(this IEnumerable<int> collection)
    {
        return collection.Where(x => x > 0 && x % 2 == 0);
    }
}

class Program
{
    static void Main()
    {
        // Example usage
        int number1 = 1274;
        Console.WriteLine($"Sum of digits of {number1}: {number1.SummaDigit()}");

        uint number2 = 132;
        Console.WriteLine($"Sum of {number2} and its reverse: {number2.SummaWithReverse()}");

        string str = "I like C#";
        Console.WriteLine($"Number of non-Latin letters in \"{str}\": {str.CountNotLetter()}");

        DayOfWeek day = DayOfWeek.Sunday;
        Console.WriteLine($"Is {day} a day off? {day.IsDayOff()}");

        int[] array = { 2, -2, 3, 4, 0, 6, 1, 9 };
        Console.WriteLine($"Even positive elements in array: {string.Join(", ", array.EvenPositiveElements())}");

        List<int> list = new List<int> { 2, 3, -4, 8, 5, 4 };
        Console.WriteLine($"Even positive elements in list: {string.Join(", ", list.EvenPositiveElements())}");
    }
}
```

OUTPUT:

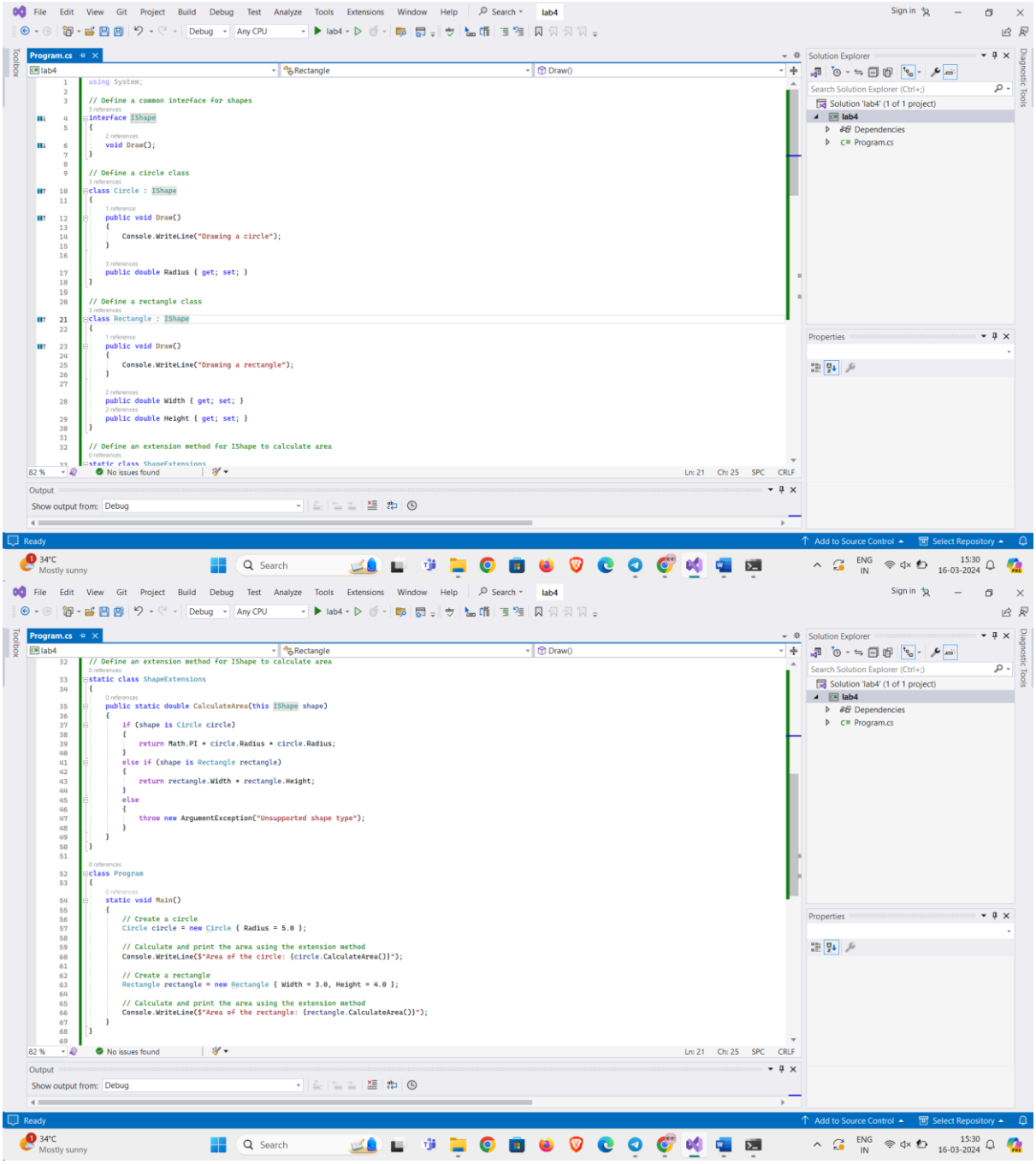


```
Sum of digits of 1274: 14
Sum of 132 and its reverse: 363
Number of non-Latin letters in "I like ?#": 4
Is Sunday a day off? True
Even positive elements in array: 2, 4, 6
Even positive elements in list: 2, 8, 4

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab4\lab4\bin\Debug\net8.0\lab4.exe (process 11072) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

POST-LAB

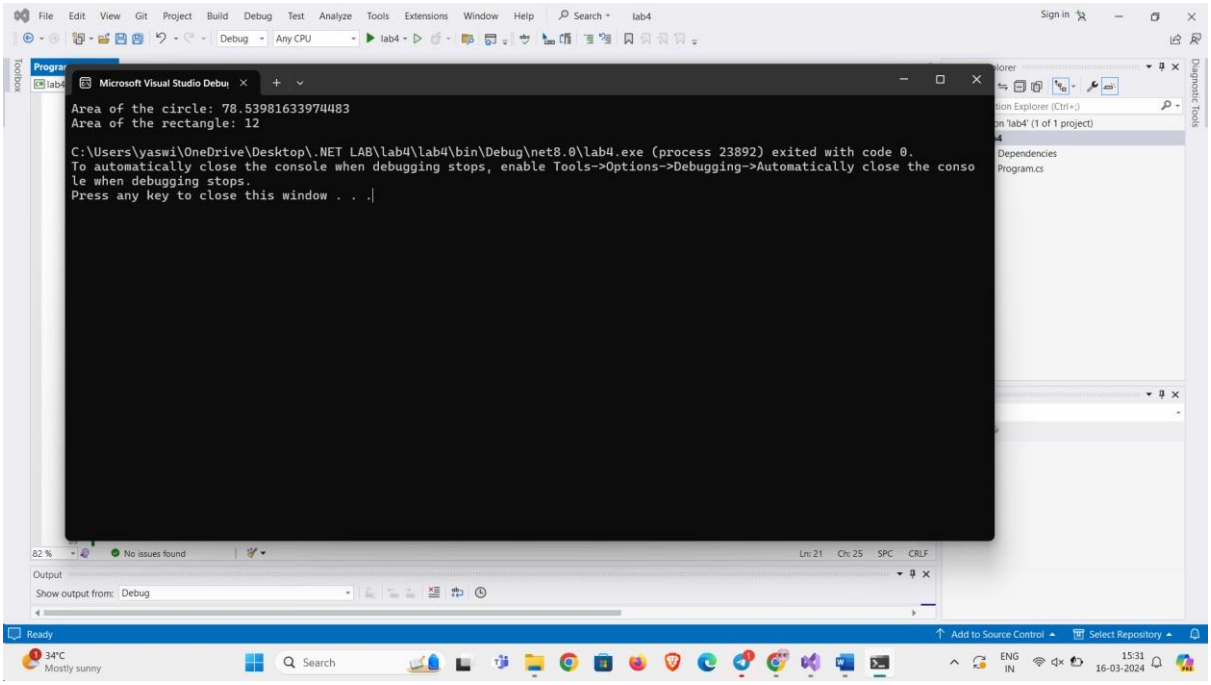
1. Why and How the Extension Methods are helpful to achieve the desired operation in an Application? Give an Example?

Solution:



```
using System;

// Define a common interface for shapes
interface IShape
{
    void Draw();
}

// Define a circle class
class Circle : IShape
{
    public void Draw()
    {
        Console.WriteLine("Drawing a circle");
    }

    public double Radius { get; set; }
}

// Define a rectangle class
class Rectangle : IShape
{
    public void Draw()
    {
        Console.WriteLine("Drawing a rectangle");
    }

    public double Width { get; set; }
    public double Height { get; set; }
}

// Define an extension method for IShape to calculate area
static class ShapeExtensions
{
    public static double CalculateArea(this IShape shape)
    {
        if (shape is Circle circle)
        {
            return Math.PI * circle.Radius * circle.Radius;
        }
        else if (shape is Rectangle rectangle)
        {
            return rectangle.Width * rectangle.Height;
        }
        else
        {
            throw new ArgumentException("Unsupported shape type");
        }
    }
}

class Program
{
    static void Main()
    {
        // Create a circle
        Circle circle = new Circle { Radius = 5.0 };

        // Calculate and print the area using the extension method
        Console.WriteLine($"Area of the circle: {circle.CalculateArea()}");

        // Create a rectangle
        Rectangle rectangle = new Rectangle { Width = 3.0, Height = 4.0 };

        // Calculate and print the area using the extension method
        Console.WriteLine($"Area of the rectangle: {rectangle.CalculateArea()}");
    }
}
```

OUTPUT:


Microsoft Visual Studio Debu
Area of the circle: 78.53981633974483
Area of the rectangle: 12
C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab4\lab4\bin\Debug\net8.0\lab4.exe (process 23892) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .