

Assignment – 01

Yash Yadav(107)

1.Calculator application

create a simple calculator application that can perform basic arithmetic operations (addition, subtraction, multiplication, and division). The application should take input from the user and display the result.

The screenshot shows a GitHub Copilot Walkthrough interface. On the left, there's a sidebar titled "Grading" with a progress bar at 68%. The main area displays a C# code editor with the file "Program.cs". The code implements a basic calculator with addition, subtraction, multiplication, division, and modulus operations. It uses `Console.ReadLine` for input and `Console.WriteLine` for output. The code is annotated with comments explaining its logic. The status bar at the bottom indicates "No issues found" and shows the current line and character count as "Ln: 2 Ch: 38 SPC CRLF".

```
using System;
using System.Security.Authentication;
using System.Security.Cryptography.X509Certificates;
using System.Linq;
public class Class1
{
    static void Main(string[] args)
    {
        Console.WriteLine("Basic Calculator");
        double a, b;
        int n;
        Console.WriteLine("Enter first number: ");
        a = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Enter second number: ");
        b = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Enter Your Choice : ");

        Console.WriteLine("1. Addition");
        Console.WriteLine("2. Subtraction");
        Console.WriteLine("3. Multiplication");
        Console.WriteLine("4. Division");
        Console.WriteLine("5. Modulus");
        n = Convert.ToInt32(Console.ReadLine());
        switch (n)
        {
            case 1:
                double sum = a + b;
                Console.WriteLine("The sum is: " + sum);
                break;
            case 2:
                double difference = a - b;
                Console.WriteLine("The difference is: " + difference);
                break;
            case 3:
                double product = a * b;
                Console.WriteLine("The product is: " + product);
                break;
            case 4:
                if (b != 0)
                {
                    double quotient = a / b;
                    Console.WriteLine("The quotient is: " + quotient);
                }
                else
                {
                    Console.WriteLine("Cannot divide by zero.");
                }
                break;
            default:
                Console.WriteLine("Invalid Choice");
                break;
        }
    }
}
```

```
Basic Calculator
Enter first number:
55
Enter second number:
5
Enter Your Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
4
The quotient is: 11

C:\Users\yadav\source\repos\Gradeing1\Gradeing1\bin\Debug\net8.0\Gradeing1.exe (process 38040) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

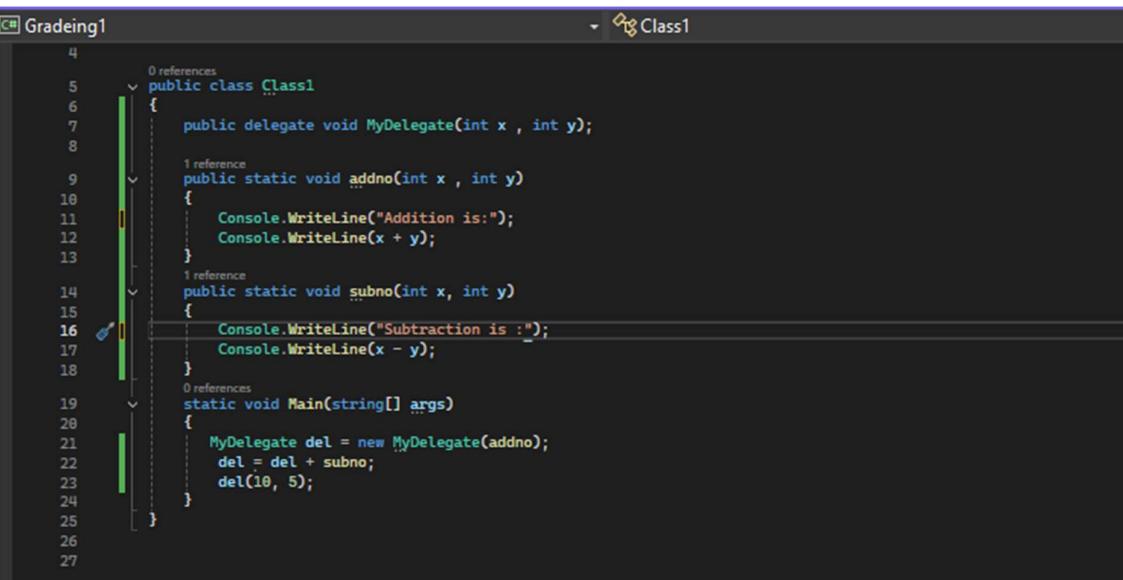
```
Microsoft Visual Studio Debug + X - □ ×

Basic Calculator
Enter first number:
55
Enter second number:
2
Enter Your Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
3
The product is: 110

C:\Users\yadav\source\repos\Gradeing1\Gradeing1\bin\Debug\net8.0\Gradeing1.exe (process 27580) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

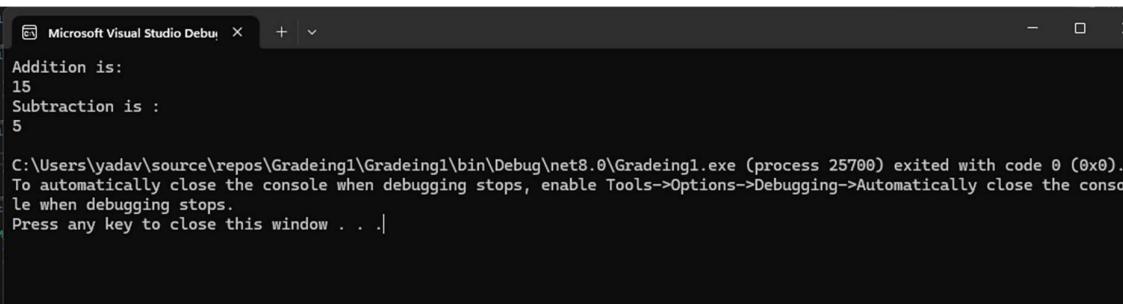
2. Delegate Task

Implement multicast delegates Using Add and Sub two methods perform addition and subtraction. The Multicast delegate is a delegate which holds a reference to more than one method. Using the delegate object variable 'del' we have to call the methods by passing the value as argument.



The screenshot shows the code editor for a C# project named "Gradeing1". The file is "Class1.cs". The code defines a multicast delegate "MyDelegate" and two methods, "addno" and "subno", which are added to this delegate. The "Main" method creates an instance of "MyDelegate", adds "addno" to it, adds "subno" to it, and then calls the delegate with arguments 10 and 5. The output window shows the results of the addition and subtraction.

```
4      0 references
5      public class Class1
6      {
7          public delegate void MyDelegate(int x , int y);
8
9          1 reference
10         public static void addno(int x , int y)
11         {
12             Console.WriteLine("Addition is:");
13             Console.WriteLine(x + y);
14         }
15
16         1 reference
17         public static void subno(int x, int y)
18         {
19             Console.WriteLine("Subtraction is :");
20             Console.WriteLine(x - y);
21         }
22
23         0 references
24         static void Main(string[] args)
25         {
26             MyDelegate del = new MyDelegate(addno);
27             del = del + subno;
28             del(10, 5);
29         }
30     }
```



The screenshot shows the "Microsoft Visual Studio Debug" window. It displays the output of the program's execution. The program prints "Addition is:", then "15", then "Subtraction is :", then "5". Below this, it shows the path to the executable and a message about automatically closing the console. At the bottom, it says "Press any key to close this window . . .".

```
Microsoft Visual Studio Debug X + v
Addition is:
15
Subtraction is :
5
C:\Users\yadav\source\repos\Gradeing1\Gradeing1\bin\Debug\net8.0\Gradeing1.exe (process 25700) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

3.C# Program to check whether the entered number is even or odd.

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "Gradeing1". The code editor window contains the following C# code:

```
Gradeing1 Class1.cs
1  using System;
2  using System.Security.Authentication;
3  using System.Security.Cryptography.X509Certificates;
4
5  2 references
6  public class Class1
7  {
8      1 reference
9      public void EvenOrOdd()
10     {
11         Console.WriteLine("Enter a Number : ");
12         int x = int.Parse(Console.ReadLine());
13         if (x % 2 == 0)
14         {
15             Console.WriteLine(x + " is Even Number");
16         }
17         else
18         {
19             Console.WriteLine(x + " is Odd Number");
20         }
21     }
22
23  0 references
24  static void Main(string[] args)
25  {
26      Class1 c = new Class1();
27      c.EvenOrOdd();
28  }
29
```

The screenshot shows the Microsoft Visual Studio Debug console window. The output is:

```
Microsoft Visual Studio Debug X + ▾
Enter a Number :
71
71 is Odd Number

C:\Users\yadav\source\repos\Gradeing1\Gradeing1\bin\Debug\net8.0\Gradeing1.exe (process 28872) exited
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatica
le when debugging stops.
Press any key to close this window . . .|
```

4. Problem Description

Create C# Program With Generic Delegate to check whether the entered number is even or odd.

```
Gradeing1 Class1 Main(string[] args)
1  using System;
2  using System.Security.Authentication;
3  using System.Security.Cryptography.X509Certificates;
4
5  public class Class1
6  {
7      public delegate void EvenOddDelegate();
8      public static void EvenOrOdd()
9      {
10         Console.WriteLine("Enter a Number : ");
11         int x = int.Parse(Console.ReadLine());
12         if (x % 2 == 0)
13         {
14             Console.WriteLine(x + " is Even Number");
15         }
16         else
17         {
18             Console.WriteLine(x + " is Odd Number");
19         }
20     }
21
22     static void Main(string[] args)
23     {
24         EvenOddDelegate del = new EvenOddDelegate(EvenOrOdd);
25         del();
26     }
27 }
28
29
```

```
C:\Users\yadav\source\repos> Enter a Number :
4
4 is Even Number
```

5.Create Interface Based Program To Calculate Area of Circle Square And Triangle

The screenshot shows the Microsoft Visual Studio interface. The top part is the code editor with the file `Program.cs` open. The code defines an interface `ICalculate` with methods for triangle, circle, and square area calculations. It also contains a class `AreaCalculator` that implements this interface and a `Main` method to test it. The bottom part is the output window showing the console log of the program's execution.

```
GitHub Copilot Walkthrough      Program.cs  Class1      Main(string[] args)
Gradeing1
1  using System;
2  using System.Security.Authentication;
3  using System.Security.Cryptography.X509Certificates;
4
5  public class Class1
6  {
7      interface ICalculate
8      {
9          void TriangleArea();
10         void CircleArea();
11         void SquareArea();
12     }
13     class AreaCalculator : ICalculate
14     {
15         public void TriangleArea()
16         {
17             double b;
18             double h;
19             double Area;
20             Console.WriteLine("Enter the triangle Base :");
21             b = double.Parse(Console.ReadLine());
22             Console.WriteLine("Enter the triangle Hieght : ");
23             h = double.Parse(Console.ReadLine());
24             Area = 0.5 * b * h;
25             Console.WriteLine("Area of Triangle is : "+Area);
26         }
27     }
28     public void CircleArea()
29     {
30         float r;
31         float Area;
32         Console.WriteLine("Enter the Circle Radius");
33         r = float.Parse(Console.ReadLine());
34         Area = 3.14f * r * r;
35         Console.WriteLine("Area of Circle is : "+Area);
36     }
37     public void SquareArea()
38     {
39         double s;
40         Console.WriteLine("Enter the Square Side :");
41         s = int.Parse(Console.ReadLine());
42         double Area = s * s;
43         Console.WriteLine("Area of Square is : "+Area);
44     }
45 }
46 static void Main(string[] args)
47 {
48     AreaCalculator a = new AreaCalculator();
49     a.TriangleArea();
50     a.CircleArea();
51     a.SquareArea();
52 }
```

```
Microsoft Visual Studio Debug X + -
Enter the triangle Base :
5
Enter the triangle Hieght :
5
Area of Triangle is : 12.5
Enter the Circle Radius
6
Area of Circle is : 113.04
Enter the Square Side :
8
Area of Sqaure is : 64

C:\Users\yadav\source\repos\Gradeing1\Gradeing1\bin\Debug\net8.0\Gradeing1.exe (process 9840) exited with code 0 (0x0)
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

6.Dependency Injection Task

```
using System.Collections.Generic;
namespace DependencyInjectionDesignPattern
{
    public class EmployeeDAL
    {
        public List<Employee> SelectAllEmployees()
        {
            List<Employee> ListEmployees = new List<Employee>
            {
                //Get the Employees from the Database
                //for now we are hard coded the employees
                new Employee() { ID = 1, Name = "Pranaya", Department =
                    "IT" },
                new Employee() { ID = 2, Name = "Kumar", Department =
                    "HR" },
                new Employee() { ID = 3, Name = "Rout", Department =
                    "Payroll" }
            };
            return ListEmployees;
        }
    }
}
```

Create A Class And Method Using Dependency Injection

To Show All Employees Shown In Above Class

```
using System;
using System.Collections.Generic;
namespace Gradeing1
{
    public class Program
    {
        public interface IEmployee
        {
            List<Employee> SelectAllEmp();
        }
        public class Employee
        {
            public int ID { get; set; }
            public string Name { get; set; }
            public string Department { get; set; }
        }
        public class EmpDAL : IEmployee
        {
            public List<Employee> SelectAllEmp()
            {
                return new List<Employee>
                {
                    new Employee() { ID = 1, Name = "Pranaya", Department = "IT" },
                    new Employee() { ID = 2, Name = "Kumar", Department = "HR" },
                    new Employee() { ID = 3, Name = "Rout", Department = "Payroll" }
                };
            }
        }
        public class EmpBL
        {
            private EmpDAL empDAL;
            public EmpBL(IEmployee empDAL)
            {
                this.empDAL = empDAL;
            }
            public void DisplayEmp()
            {
                List<Employee> employees = empDAL.SelectAllEmp();
                foreach (var emp in employees)
                {
                    Console.WriteLine($"{emp.ID}: {emp.Name}, {emp.Department}");
                }
            }
        }
        public static void Main(string[] args)
        {
            EmpBL empBL = new EmpBL();
            empBL.DisplayEmp();
        }
    }
}
```

```
ID: 1, Name: Pranaya, Department: IT
ID: 2, Name: Kumar, Department: HR
ID: 3, Name: Rout, Department: Payroll

C:\Users\yadav\source\repos\Gradeing1\Gradeing1\bin\Debug\net8.0\Gradeing1.exe (process 5184) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```