Name: Sarvesh Sawant Date:12/12/2023
Subject: ASP.NET Class: SYIT

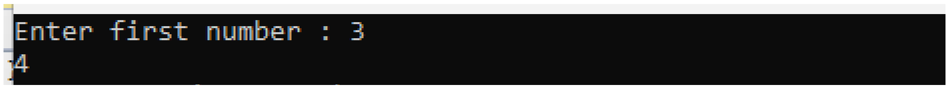
Practical 2: Working with object oriented C# and ASP.NET

a.) Create a simple application to perform the foll operations

1. Functions Overloading:
   Write a C# program which has 2 methods “add” which can either add 2 numbers or concatenate two strings which are taken as input from the user and display the output.
   Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Practical2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter first number : ");
            int a = Convert.ToInt32(Console.ReadLine());
            int b = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter 2 strings : ");
            String n1 = Console.ReadLine();
            String n2 = Console.ReadLine();

            Overloading a1 = new Overloading();
            a1.add(a, b);
            a1.add(n1, n2);
            Console.ReadLine();
        }
    }

    class Overloading
    {


        public void add(int a, int b)
        {
            Console.WriteLine("additon of 2 number" + a + "and" + b + "is:" + (a + b));

        }

        public void add(string a, string b)
        {
            Console.WriteLine("concentration of 2 number " + a + "and" + b + "is:" + (a +
b));

        }
    }
    }
```

Output:

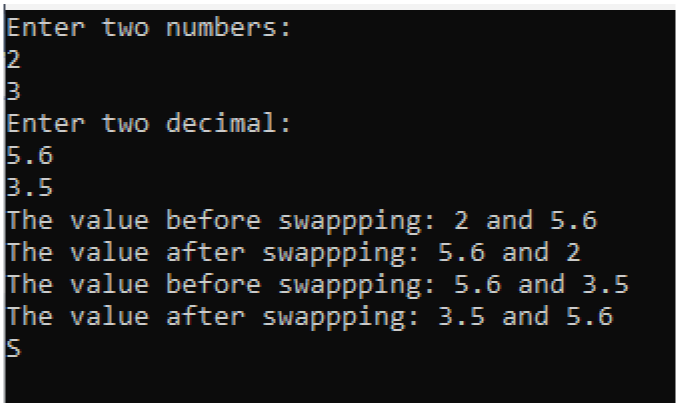


```
Enter first number : 3
4
```

2. Constructor overloading
   Write a C# program which has 2 constructors which can swap 2 integers or 2 floating numbers which are taken as input from the user and display the output.
   Code:

```
using System;

namespace Practical2
{
    class swap
    {
        swap(int a, int b)
        {
            Console.WriteLine("The value before swappping: " + a + " and " + b);
            int t;
            t = a;
            a = b;
            b = t;
            Console.WriteLine("The value after swappping: " + a + " and " + b);
        }
        swap(double f1, double f2)
        {
            Console.WriteLine("The value before swappping: " + f1 + " and " + f2);
            double t;
            t = f1;
            f1 = f2;
            f2 = t;
            Console.WriteLine("The value after swappping: " + f1 + " and " + f2);
        }
        static void Main(string[] args)
        {
            Console.WriteLine("Enter two numbers:");
            int a1 = Convert.ToInt32(Console.ReadLine());
            int b1 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter two decimal:");
            double a2 = Convert.ToDouble(Console.ReadLine());
            double b2 = Convert.ToDouble(Console.ReadLine());

            swap s1 = new swap(a1, a2);
            swap s2 = new swap(a2, b2);
            Console.ReadLine();
        }
    }
}
```

Output:



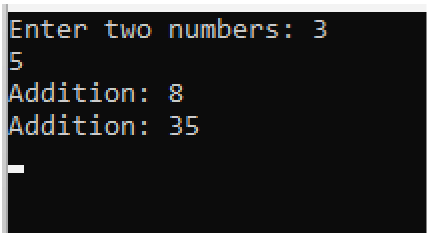
```
Enter two numbers:
2
3
Enter two decimal:
5.6
3.5
The value before swappping: 2 and 5.6
The value after swappping: 5.6 and 2
The value before swappping: 5.6 and 3.5
The value after swappping: 3.5 and 5.6
S
```

3. Interfaces

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Practical2
{
    interface Calculator
    {
        void add(int a, int b);
    }

    public class DemoNumber : Calculator
    {
        public void add(int a, int b)
        {
            Console.WriteLine("Addition: " + (a + b));
        }
    }

    public class DemoString : Calculator
    {
        public void add(int a, int b)
        {
            Console.WriteLine("Addition: " + a + b);
        }
    }

    class calculotor
    {
        static void Main(string[] args)
        {

            Console.Write("Enter two numbers: ");
            string s1 = Console.ReadLine();
            string s2 = Console.ReadLine();
            int n1 = Convert.ToInt32(s1);
            int n2 = Convert.ToInt32(s2);

            DemoNumber demo = new DemoNumber();
            demo.add(n1, n2);

            DemoString demoS = new DemoString();
            demoS.add(n1, n2);
            Console.Read();

        }
    }
}
```

Output:



```
Enter two numbers: 3
5
Addition: 8
Addition: 35
```

4. Inheritance (all types)

b.) Create a simple applications to demonstrate use of foll concepts

1.Using Delegates and events

2. Exceptions handling