AWT UNIT 2 @ UNIT 3 SUBMISSION
Roll No : 47

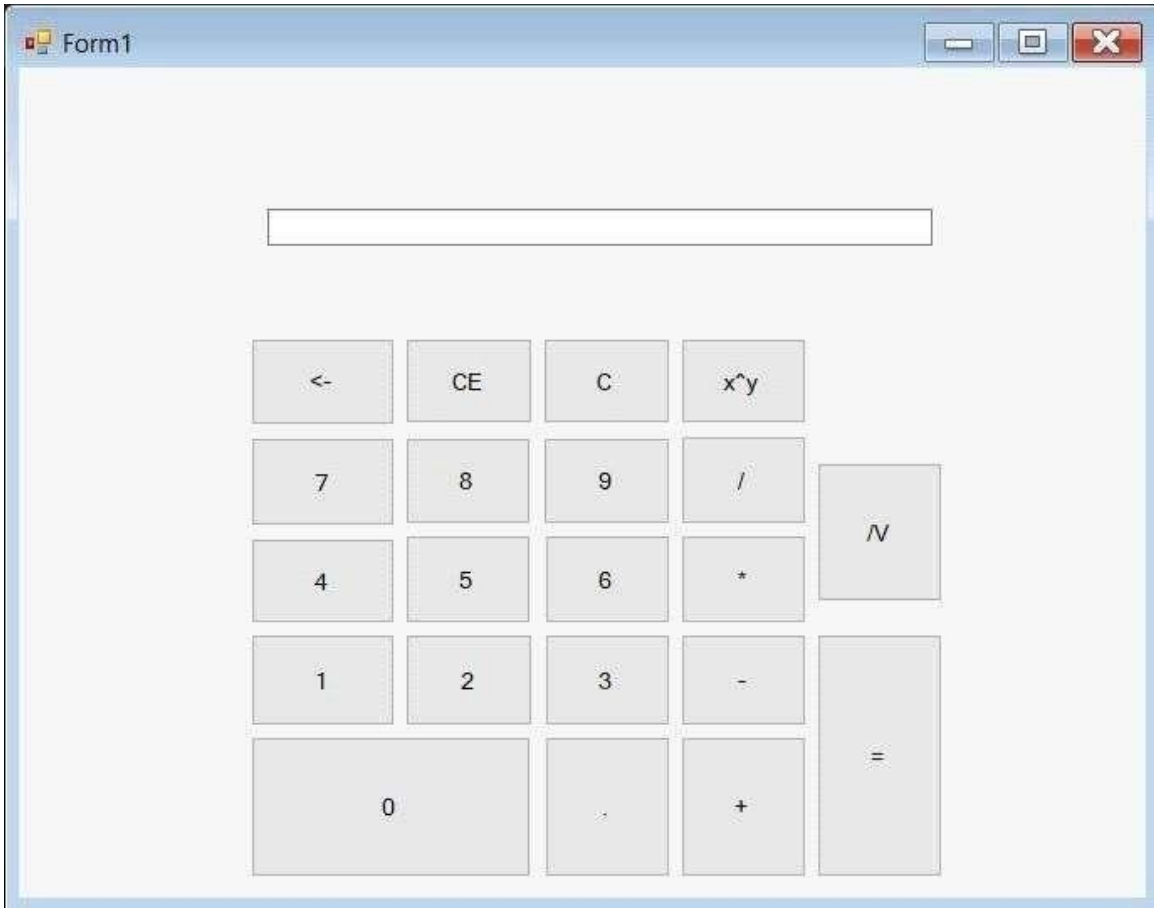
Batch – A3

Unit 1

Basics Of C#

Practical 1: Create a functional calculator using C#

Design:



cal.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```

```
 using System.Drawing;
 using System.Linq;
 using System.Text;
 using System.Threading.Tasks;
 using System.Windows.Forms;

 namespace Calculator
 {

public partial class cal : Form
 {
 // variable to hold operands
 string calfun;
 double v1, v2;

 public cal()
 {
 InitializeComponent();
 }

 private void btn0_Click(object sender, EventArgs e)
 {
 txtinput.Text += btn0.Text;
 }

 private void btn1_Click(object sender, EventArgs e)
 {
 txtinput.Text += btn1.Text;
 }

 private void btn2_Click(object sender, EventArgs e)
 {
 txtinput.Text += btn2.Text;
 }

 private void btn3_Click(object sender, EventArgs e)
 {
 txtinput.Text += btn3.Text;
 }

 private void btn4_Click(object sender, EventArgs e)
 {
```

```
txtinput.Text += btn4.Text;
}

private void btn5_Click(object sender, EventArgs e)
{
txtinput.Text += btn5.Text;
}
```

```
private void btn6_Click(object sender, EventArgs e)
{
txtinput.Text += btn6.Text;
}

private void btn7_Click(object sender, EventArgs e)
{
txtinput.Text += btn7.Text;
}

private void btn8_Click(object sender, EventArgs e)
{
txtinput.Text += btn8.Text;
}

private void btn9_Click(object sender, EventArgs e)
{
txtinput.Text += btn9.Text;
}

private void btnaddition_Click(object sender, EventArgs e)
{
v1 = Convert.ToDouble(txtinput.Text); calfun = "add" ; txtinput.Text = "";
}

private void btnsub_Click(object sender, EventArgs e)
{
v1=Convert.ToDouble(txtinput.Text); calfun = "mins"; txtinput.Text = " ";
}

private void btnmulti_Click(object sender, EventArgs e)
{
v1 = Convert.ToDouble(txtinput.Text); calfun = "multi"; txtinput.Text ="";
}

private void btndivi_Click(object sender, EventArgs e)
{
v1 = Convert.ToDouble(txtinput.Text); calfun = "div"; txtinput.Text ="";
}
```

```
private void btnsquareroot_Click(object sender, EventArgs e)
{
v1 = Convert.ToDouble(txtinput.Text);
txtinput.Text = Math.Sqrt(v1).ToString();// converted into string
}

private void btnequal_Click(object sender, EventArgs e)
{
v2 = Convert.ToDouble(txtinput.Text);
switch (calfun)
{
case "add": v1 = v1 + v2;
               break;
case "minus": v1 = v1 - v2;
               break;
case "multi": v1 = v1 * v2;
               break;
case "div": v1 = v1 / v2;
               break;
case "PowerOf": v1 = System.Math.Pow(v1, v2);
               break;
}//switch closed
txtinput.Text = v1.ToString();
}

private void btnbackspace_Click(object sender, EventArgs e)
{
if (txtinput.Text != "")
{
int l = txtinput.Text.Length;
txtinput.Text = txtinput.Text.Remove(l - 1);//
}
}

private void btndecimal_Click(object sender, EventArgs e)
{
txtinput.Text += btndecimal.Text;
}
```

```
private void btnclearentry_Click(object sender, EventArgs e)
{
txtinput.Text = string.Empty;
}

private void btnclearall_Click(object sender, EventArgs e)
{
v1 = 0; v2 = 0;
txtinput.Text = "";
}

private void btnpowerof_Click(object sender, EventArgs e)
{
v1 = Convert.ToDouble(txtinput.Text); calfun = "PowerOf"; txtinput.Text = "";
}
}}
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using  System.Threading.Tasks;
Using System.Windows.Forms;
namespace Calculator
{
static class Program
{
/// <summary>
/// The main entry point for the application.
/// </summary>
[STAThread] static
void Main()
{
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new cal());
}
}
}
```

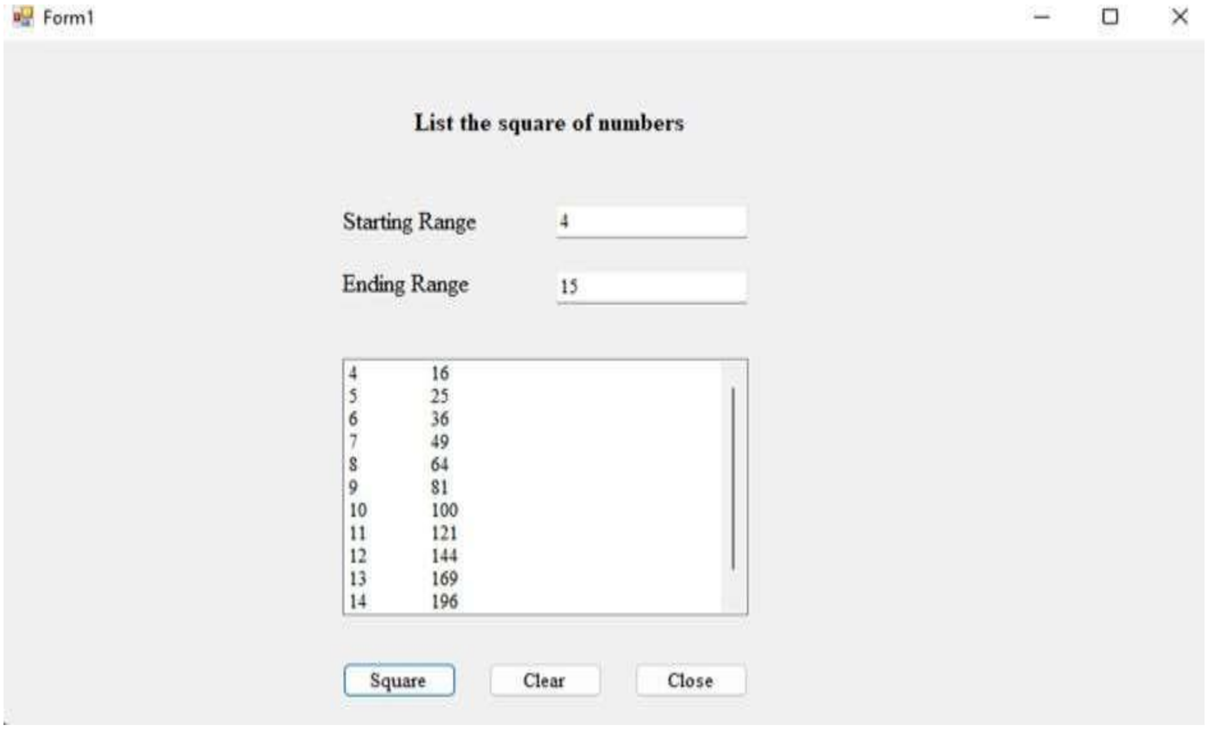
Practical 2: Write a program in C# to list the squares of given range of numbers in a list box.

Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SquareOfANumber
{
        public partial class Form1 : Form
        {
        public Form1()
        {
        InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        long start = Convert.ToInt32(textBox1.Text);
        long end = Convert.ToInt32(textBox2.Text);
        while (start <= end)
        {
                long sq = start * start;
        // string item = String.Format("{0,5}{1,30}", start, sq);
                string item = start.ToString() + "\t" + sq.ToString();
                listBox1.Items.Add(item);
                start++;
        }
        }

        private void button2_Click(object sender, EventArgs e)
        {
        textBox1.Clear();
        textBox2.Clear();
        listBox1.Items.Clear();
        }

        private void button3_Click(object sender, EventArgs e)
        {
        Close();
        }
        }
}
```

Output:

After clicking on “Square” button ->



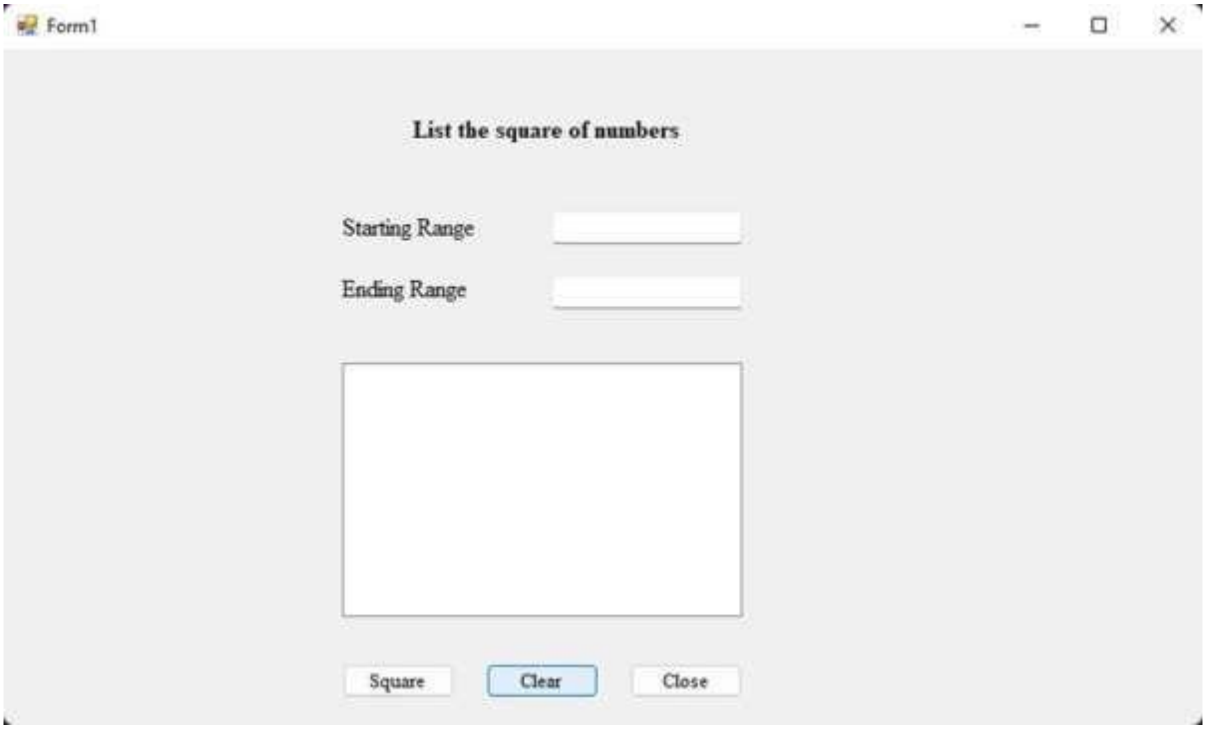
List the square of numbers

Starting Range 4

Ending Range 15

```
4    16
5    25
6    36
7    49
8    64
9    81
10   100
11   121
12   144
13   169
14   196
```

After clicking on “Clear” button ->


Form1
List the square of numbers
Starting Range
Ending Range
Square
Clear
Close

Practical 3: Create a Windows based application to design a screensaver.

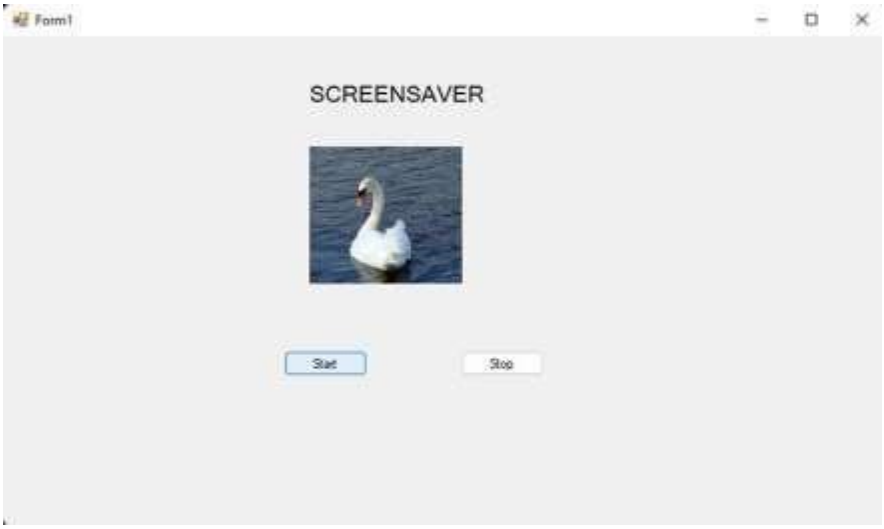
Code :

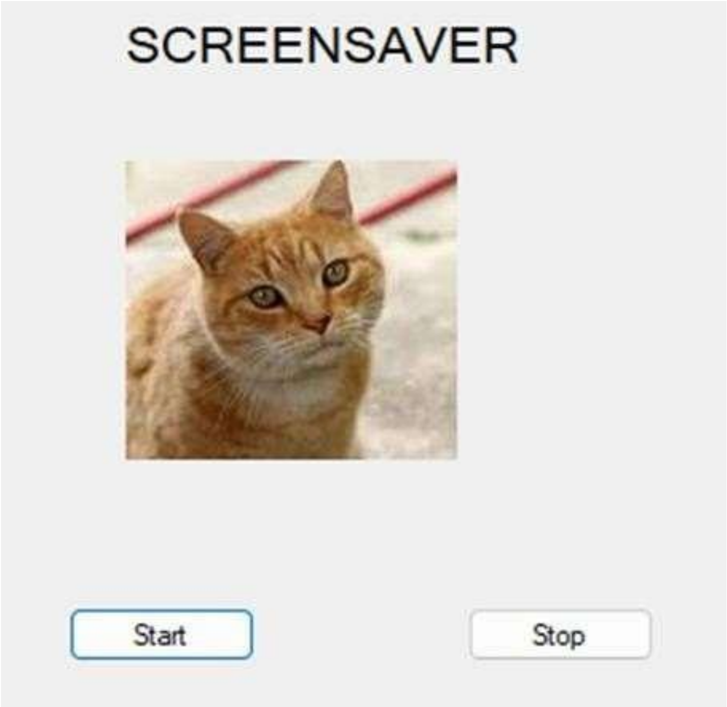
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ScreenSaver
{
        public partial class Form1 : Form
        {
        string[] s = new string[7];
        int i = 0;

        public Form1()
        {
        InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        timer1.Enabled = true;

        s[0] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic1.jpg";
        s[1] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic2.jpg";
        s[2] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic3.jpg";
        s[3] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic4.jpg";
        s[4] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic5.jpg";
        s[5] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic6.jpg";
        s[6] = @"C:\Users\Admin\Desktop\PratikshaManjrekar\AWT Lab\Images\pic7.jpg";
      }

        private void button2_Click(object sender, EventArgs e)
        {
```

```
    timer1.Enabled = false;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
    pictureBox1.Image = System.Drawing.Image.FromFile(s[i]);
    i++;
    if (i == 6)
        i = 0;
    }
    }
}
```

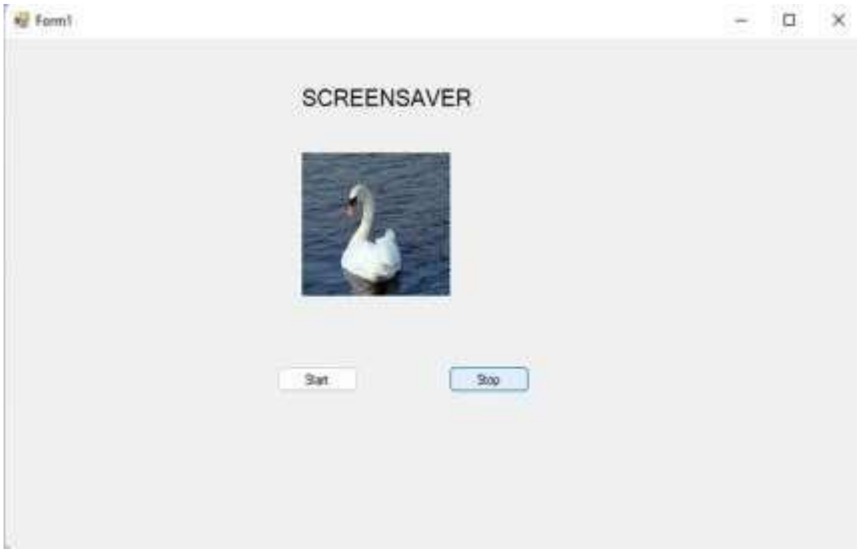
Output :

After clicking on Start button, it will start scrolling the pictures





After clicking on “Stop” button scrolling will stop



Practical 4: Create a class to check whether user input year is leap year on not.

Code :

Class.cs (Right click on project -> New -> Class)

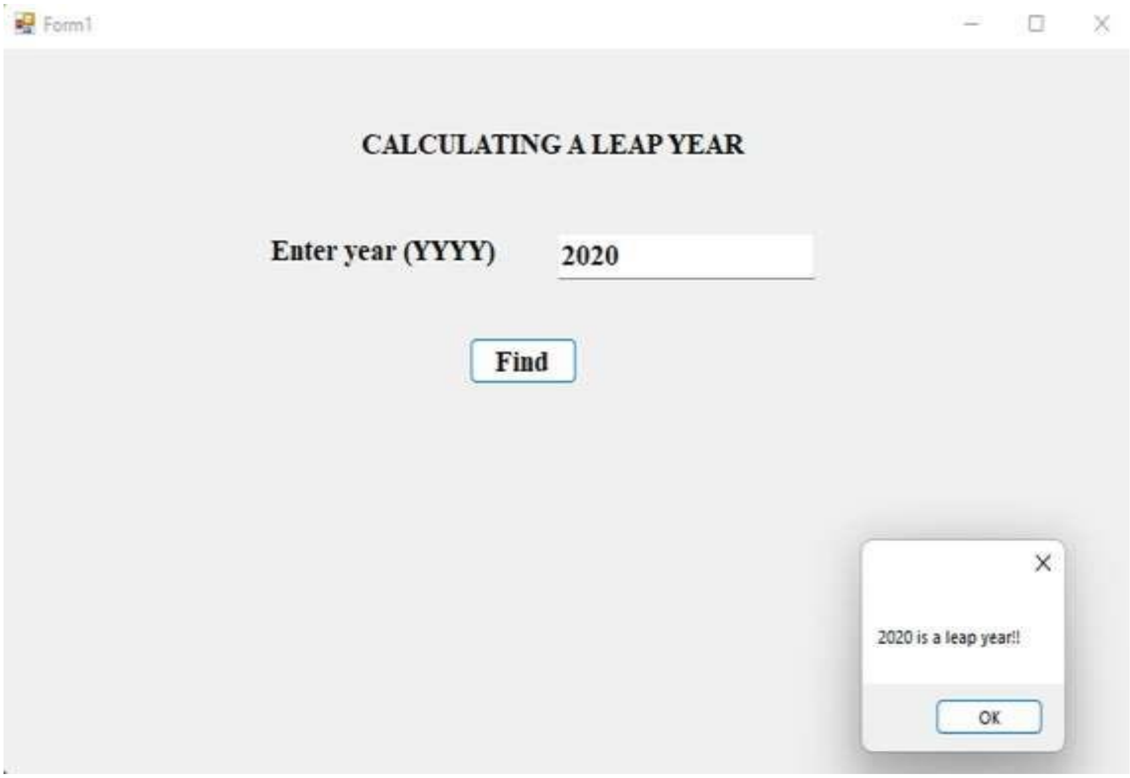
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LeapyearOrNot
{
        internal class leapYear
        {
        public int isleapyear(long year)
        {
        if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
                return 1;
        else
                return 0;
        }
        }
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LeapyearOrNot
{
        public partial class Form1 : Form
        {
        leapYear ly=new leapYear();
```

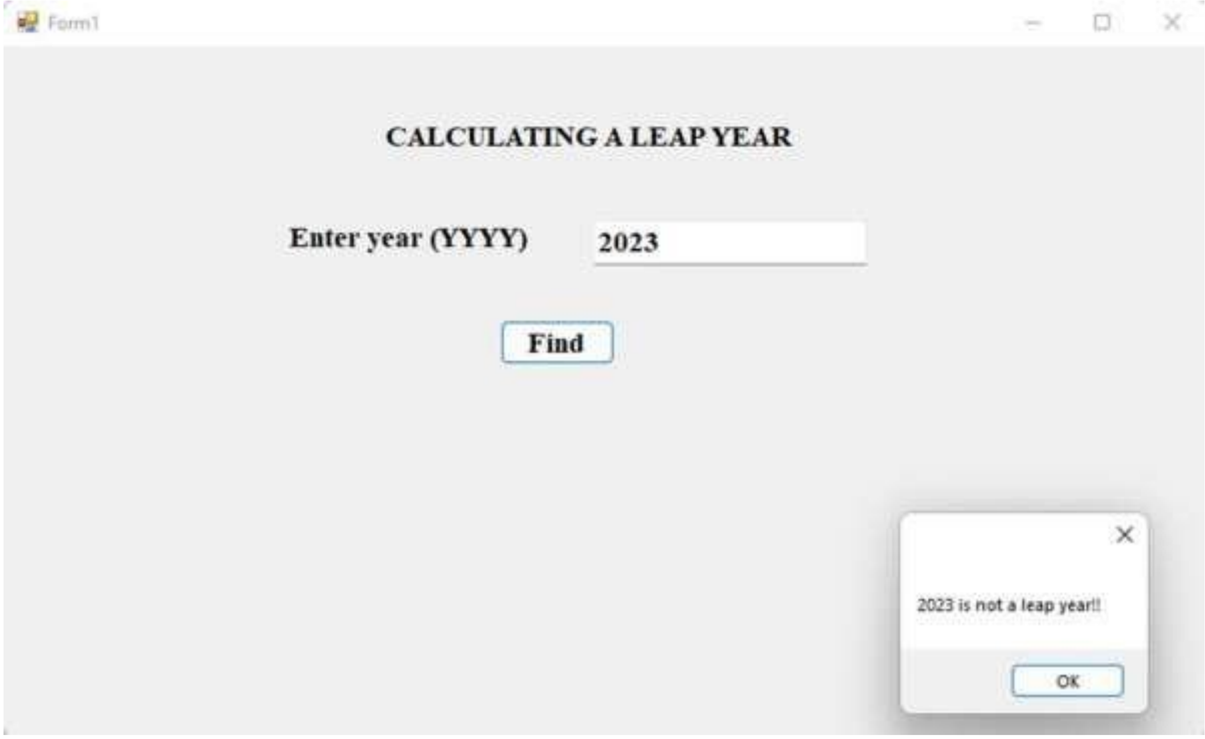
```
    public Form1()
    {
    InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
      long year = Convert.ToInt32(textBox1.Text);
    int flag = ly.isleapyear(year);
    if (flag == 1)
        MessageBox.Show(textBox1.Text + " is a leap year!!");
    else
        MessageBox.Show(textBox1.Text + " is not a leap year!!");
    }
    }
}
```

Output:



2020

2020 is a leap year!!



CALCULATING A LEAP YEAR

Enter year (YYYY) 2023

2023 is not a leap year!!

Practical 5: Write a Program to create a class to calculate simple interest and get and set method

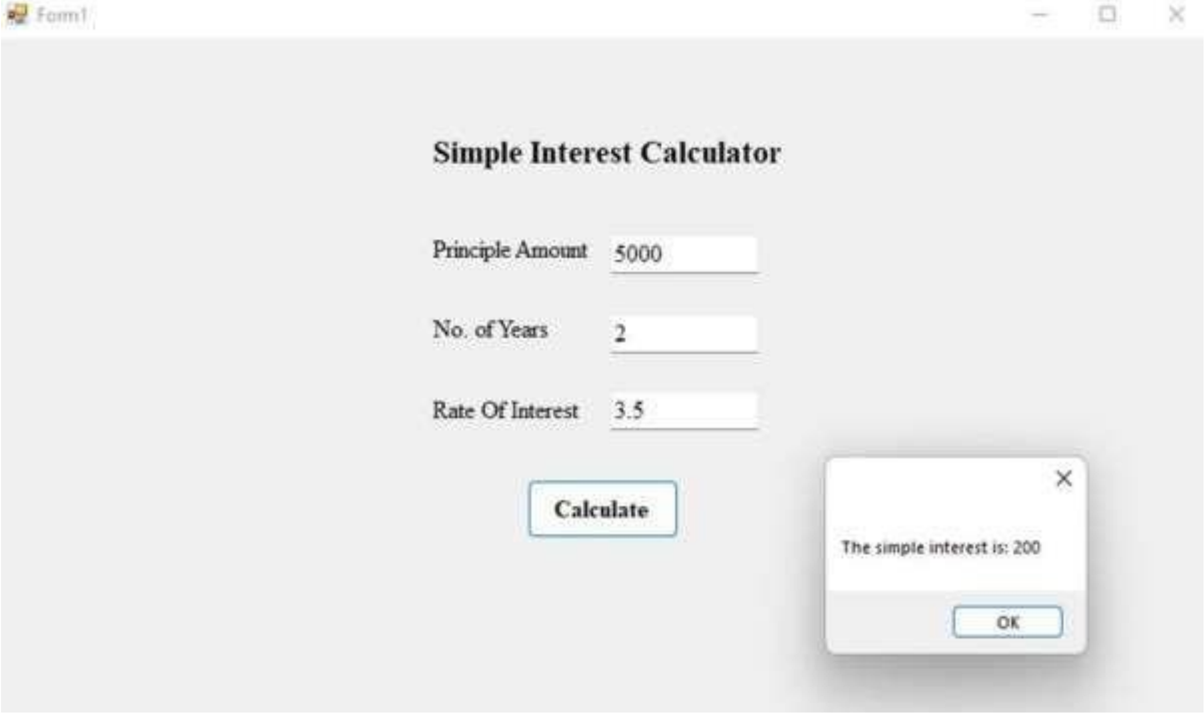
Code:

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SimpleInterestCalculator
{
        public partial class Form1 : Form
        {
        SimpleInterest si = new SimpleInterest();

        public Form1()
        {
        InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
        si.Pamt = Convert.ToDouble(textBox1.Text);
        si.Rate = Convert.ToDouble(textBox3.Text);
        si.Year = Convert.ToInt32(textBox2.Text);
        MessageBox.Show("The simple interest is: " + si.getSI().ToString());
        }
        }
}
```

class.cs

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

namespace SimpleInterestCalculator
{
    internal class SimpleInterest
    {
    double pamt, rate;
    int year;

    public double Pamt
    {
    get
    {
        return pamt;
    }
    set
    {
        pamt = value;
    }}

    public double Rate
    {
    get
    {
        return rate;
    }
    set
    {
        rate = value;
    }
    }

    public int Year
    {
    get
    {
        return year;
    }
    set
    {
        year = value;
    }
    }

    public double getSI()
    {
    return ((pamt* rate* year)/100);}}}
```



Simple Interest Calculator

Principle Amount 5000

No. of Years 2

Rate Of Interest 3.5

The simple interest is: 200

Output:

Practical 6: Write a Program using C# to create a class to implement Single inheritance

<u>Code:</u>

baseEmployee.cs (Class)

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

namespace awt1
{
    public class baseEmployee
    {
        string name;
        double basic;
        public baseEmployee(string Name, double Basic)
        {
            this.name = Name;
            this.basic = Basic;
        }
    }
}
```

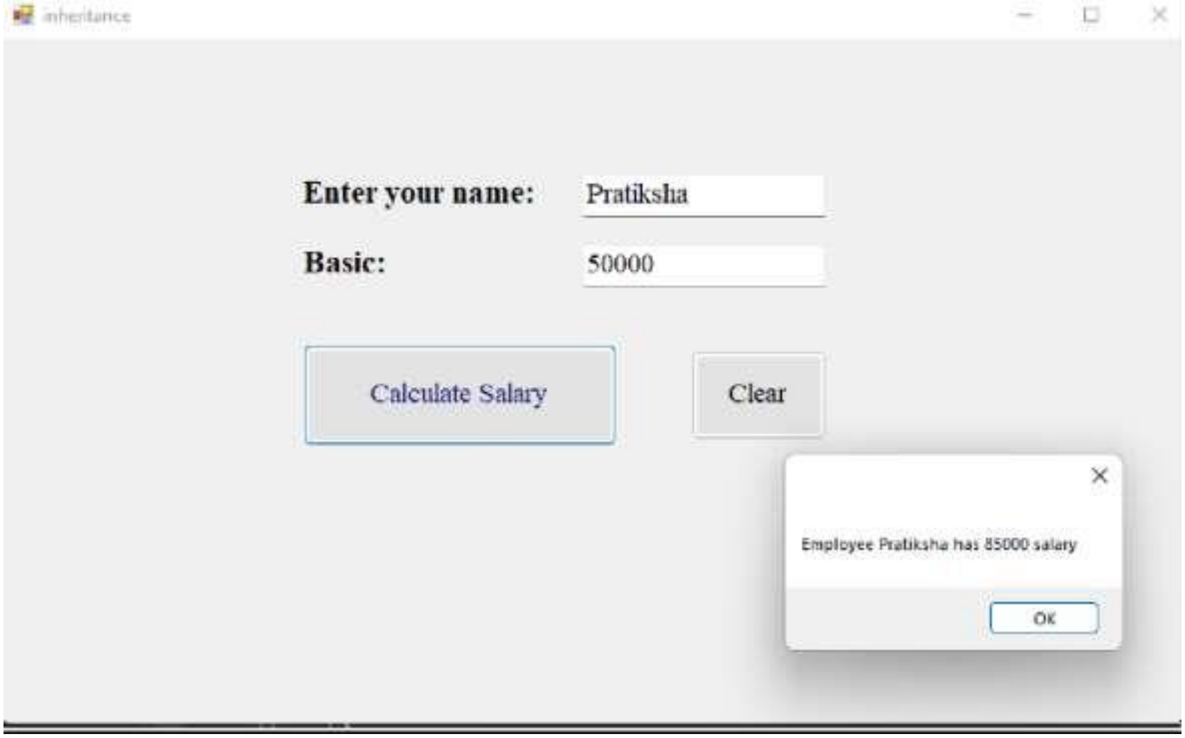
childEmployee.cs

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

namespace awt1
{
    public class childEmployee:baseEmployee
    {
        string name1;
        double basic1;

        public childEmployee(string n, double s):base(n, s)
        {
            name1 = n;
            basic1 = s;
        }

        public double getSalary()
        {
            double salary = basic1 + .30 * basic1 + .40 * basic1;
            return salary;
        }

        public string getName()
        {
            return name1;
        }
    }
}
```

singleInheritance.cs (Design)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace awt1
{
    public partial class singleInheritance : Form
    {

        public singleInheritance()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            childEmployee c = new childEmployee(textBox1.Text, Convert.ToDouble(textBox2.Text));
            MessageBox.Show("Employee " + c.getName() + " has "+ c.getSalary() +" salary");
        }
    }
}
```

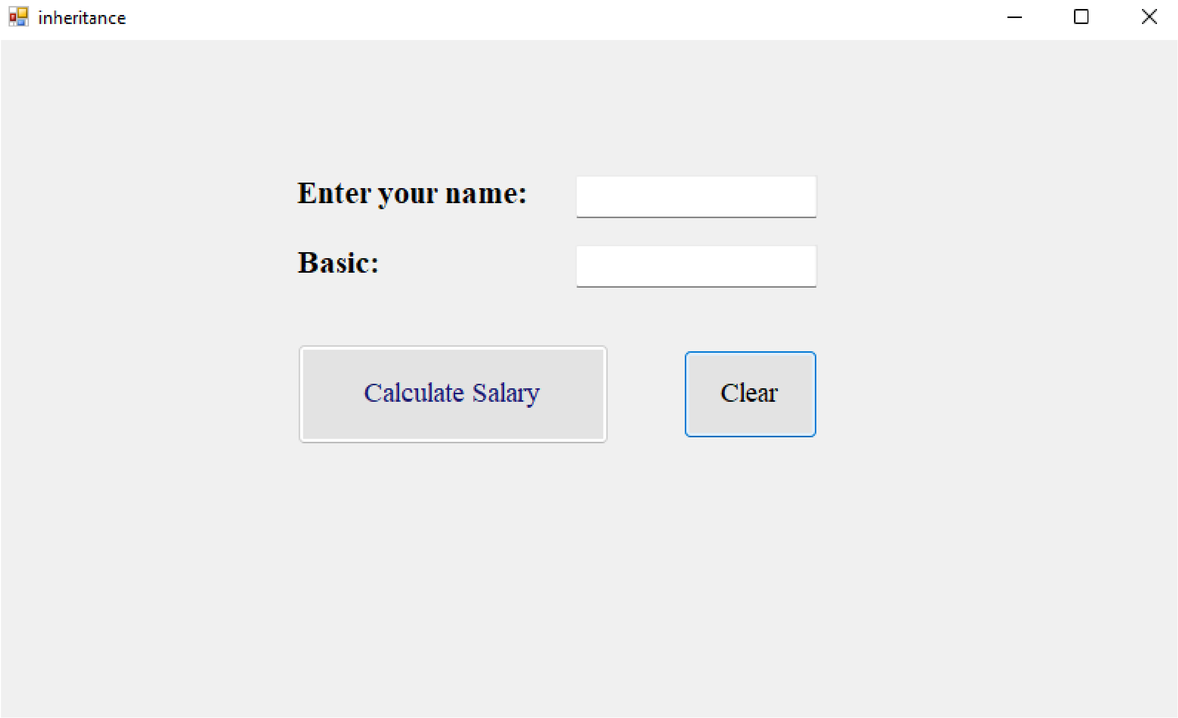
Output:



Enter your name: Pratiksha

Basic: 50000

Employee Pratiksha has 85000 salary


inheritance
Enter your name:
Basic:
Calculate Salary
Clear

Practical 7: Write a C# program to implement Hierarchical Inheritance

<u>Code:</u>

shape.cs (Class)

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

namespace awt1
{
    internal class shape
    {
        protected double height, width;

    }
}
```

rectangle.cs

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

namespace awt1
{
    internal class rectangle:shape
    {
        public void setHeight(double h)
        {
            height = h;
        }

        public void setWidth(double w)
        {
            width = w;
        }
        public double getArea()
        {
            return (height * width);
        }
    }
}
```

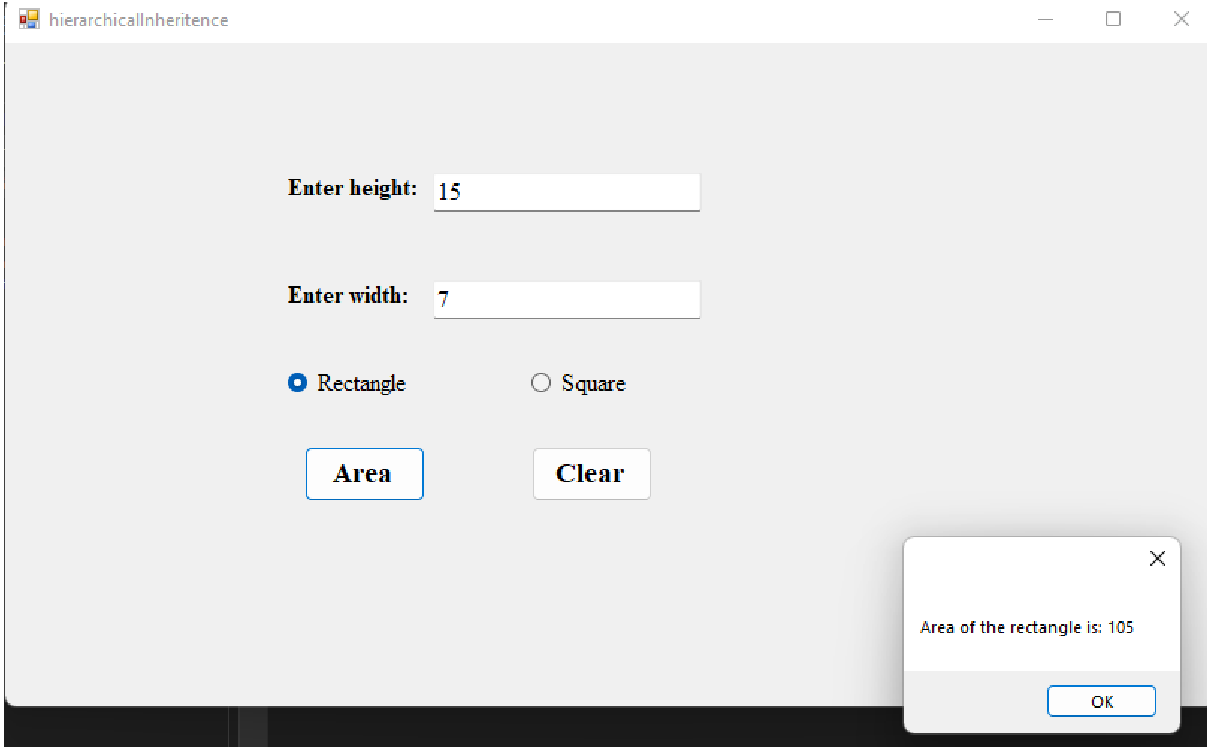
square.cs

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

namespace awt1
{
    internal class square:shape
    {
        public void setHeight(double w)
        {
            height = w;
        }

        public double getArea()
        {
            return (height * height);
        }
    }
}
```

hierarchicalInheritance.cs (Design)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace awt1
{
    public partial class hierarchicalInheritance : Form
    {
        public hierarchicalInheritance()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (rdRectangle.Checked)
            {
                rectangle r = new rectangle();
                r.setHeight(Convert.ToDouble(textBox1.Text));
                r.setWidth(Convert.ToDouble(textBox2.Text));
                MessageBox.Show("Area of the rectangle is: " + r.getArea());
            }
            else
            {
                square s = new square();
                s.setHeight(Convert.ToDouble(textBox1.Text));
                MessageBox.Show("Area of the square is: " + s.getArea());
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
        }
    }
}
```

Output:



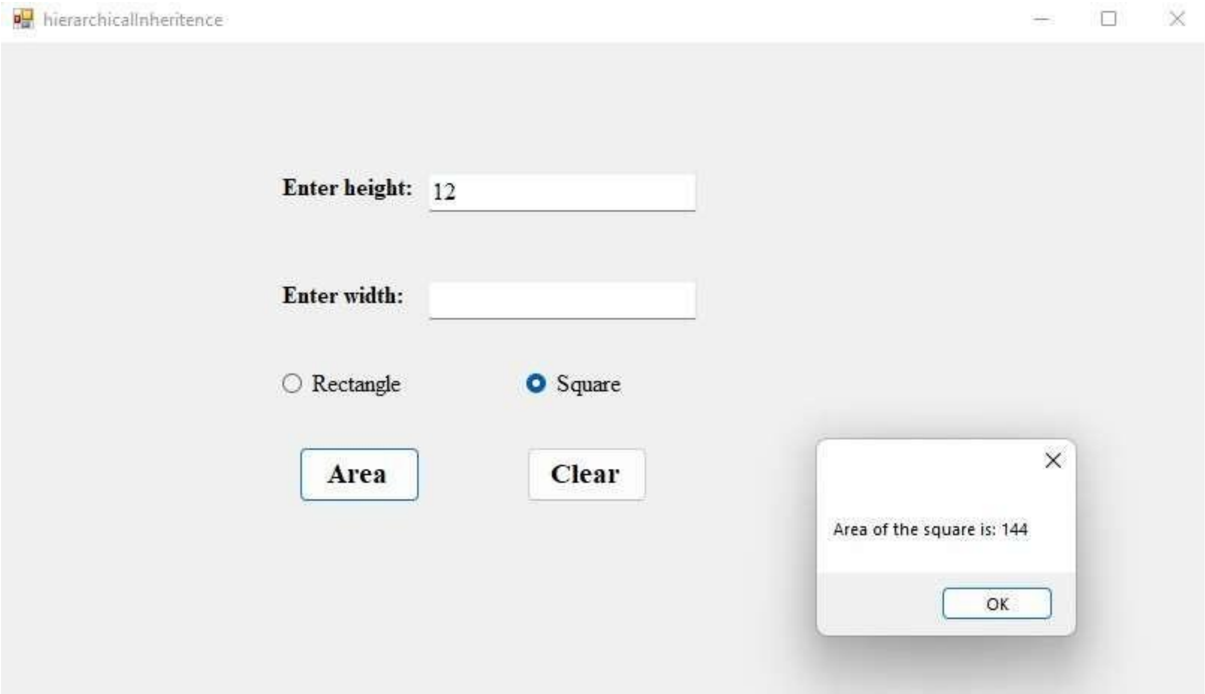
hierarchicalInheritence

Enter height: 15

Enter width: 7

● Rectangle ○ Square

Area of the rectangle is: 105



hierarchicalInheritence

Enter height: 12

Enter width:

○ Rectangle ● Square

Area of the square is: 144

Practical 8: WAP in C# to implement Abstract class. (Employee Wage Calculator)

Code:

employeeSalary.cs (Class)

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

namespace EmployeeWageCalculator
{
    internal abstract class employeeSalary
    {
        string ename, dname;
        public string Ename
        {
            get { return ename; }
            set { ename = value; }
        }

        public string Dname
        {
            get { return dname; }
            set { dname = value; }
        }

        public abstract double getSalary();
    }
}
```

manager.cs (Class)

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

namespace EmployeeWageCalculator
```

```
{
    internal class manager:employeeSalary
    {
        double basic;

        public double Basic
        {
            get { return basic; }
            set { basic = value; }
        }

        public override double getSalary()
        {
            return (basic+(basic * .30) + (basic * .40) - (basic * .50));
        }
    }
}
```

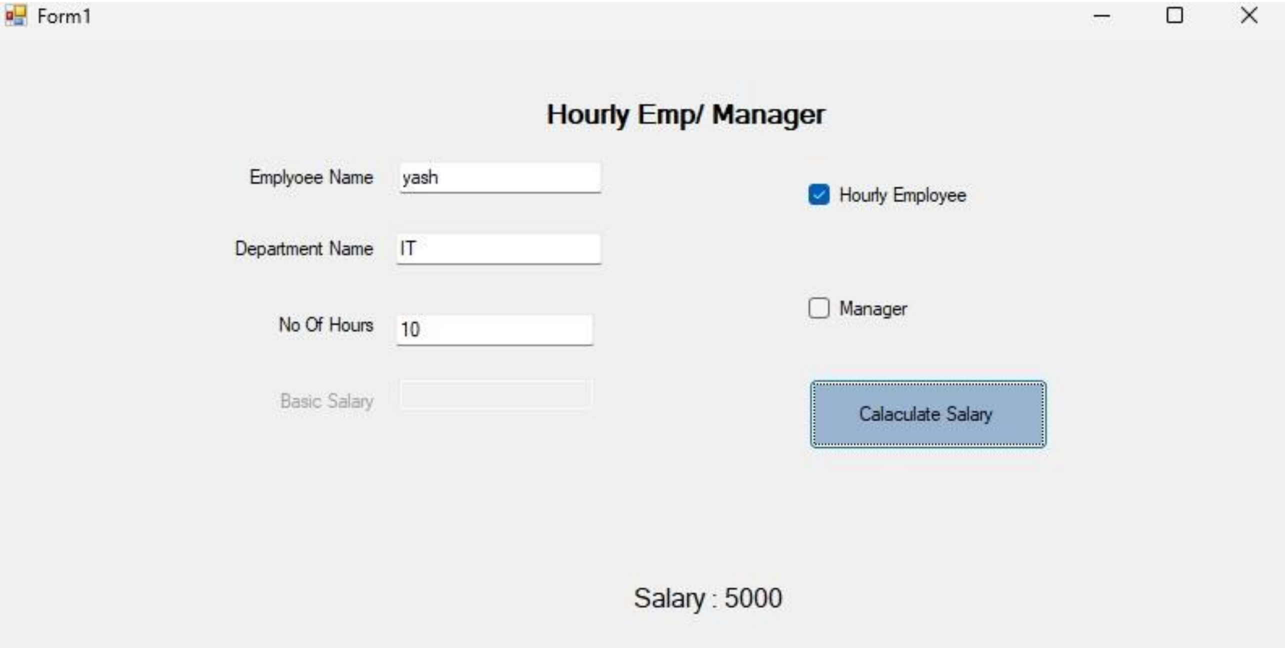
hourlyEmployee.cs (Class)

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

namespace EmployeeWageCalculator
{
    internal class hourlyEmployee:employeeSalary
    {
        int  hrs;
        public int Hrs
        {
            get { return hrs; }
            set {  hrs = value; }
        }

        public override double getSalary()
        {
            return (hrs*5000) ;
        }
    }
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EmployeeWageCalculator
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            if(textBox1.Text!="" && textBox2.Text!="")
            {
                if (checkBox1.Checked)
                {
                    if (textBox4.Text != "")
                    {
                        manager m = new manager();
                        m.Ename = textBox1.Text;
                        m.Dname = textBox2.Text;
                        m.Basic = Convert.ToDouble(textBox4.Text);
                        MessageBox.Show("Employee " + m.Ename + " working in department " + m.Dname + "
has salary " + m.getSalary());
                    }
                    else
                    {
                        MessageBox.Show("The basic must be provided!!!");
                        textBox4.Focus();
                    }
                }
                if(checkBox2.Checked)
                {
                    if(textBox3.Text!="")
                    {
                        hourlyEmployee h = new hourlyEmployee();
                        h.Ename = textBox1.Text;
                        h.Dname = textBox2.Text;
                        h.Hrs = Convert.ToInt32(textBox3.Text);
                        MessageBox.Show("Employee " + h.Ename + " working in department " + h.Dname + "
has salary " + h.getSalary());
                    }
                    else
                    {
                        MessageBox.Show("The no. of hours must be provided!!!");
                    }
                }

                if (checkBox1.Checked == false && checkBox2.Checked ==false)
                {
                    MessageBox.Show("Need to select a type of employee!!!");
                }
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
            checkBox1.Checked = false;
            checkBox2.Checked = false;
        }
      }
    }
```

Output:



Hourly Emp/ Manager

Emplyoee Name yash

Department Name IT

No Of Hours 10

Basic Salary

☑ Hourly Employee

□ Manager

Salary : 5000

Practical 9: WAP in C# to implement Interface class. (Arithmetic String Operations)

Code:

IAddition.cs (Class)

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

namespace ArithmeticStringOperations
{
    internal interface IAddition
    {
        int addition(int i1, int i2);
    }
}
```

IMultiplication.cs (Class)

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

namespace ArithmeticStringOperations
{
    internal interface IMultiplication
    {
        double multiplication(double d1, double d2);
    }
}
```

IConcat.cs (Class)

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

```
namespace ArithmeticStringOperations
{
    internal interface IConcat
    {
        string concatination(string s1, string s2);
    }
}
```

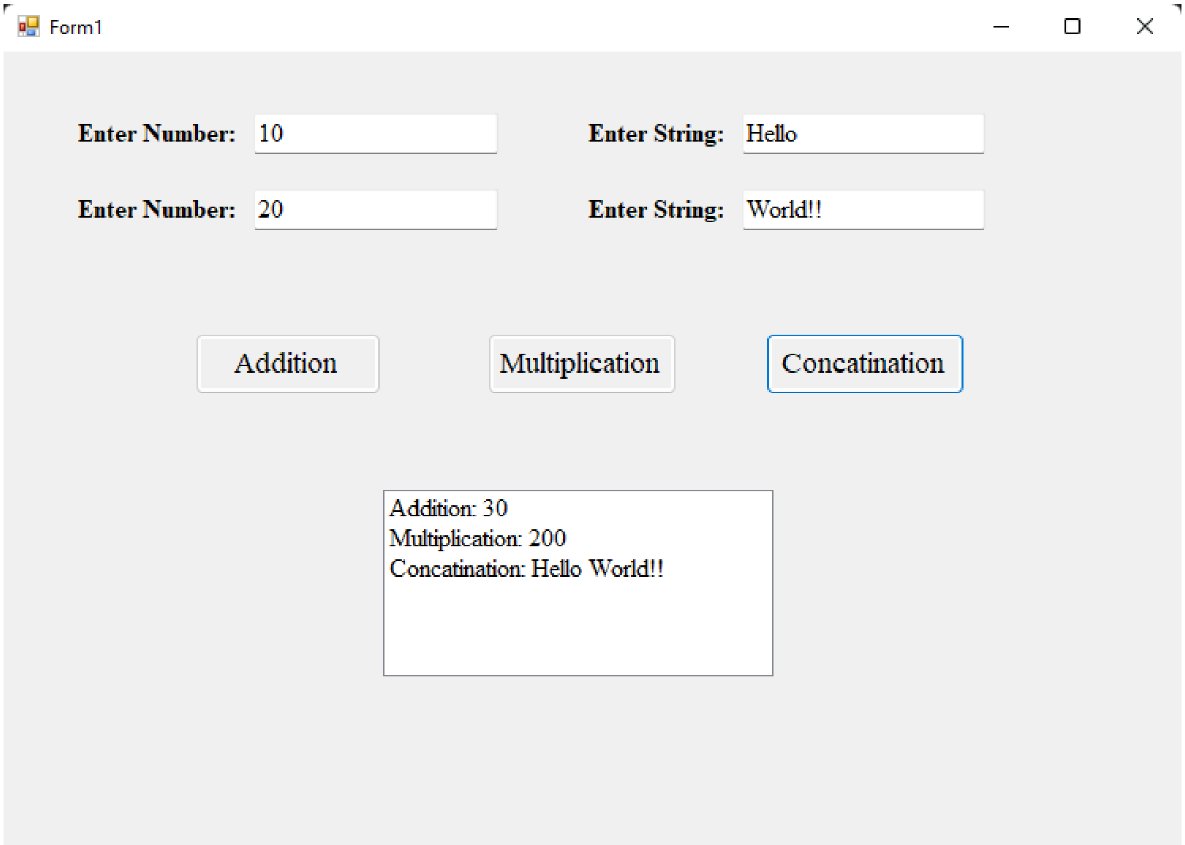
Operation.cs (Class)

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

namespace ArithmeticStringOperations
{
    internal class Operation:IAddition, IMultiplication, IConcat
    {
        public int addition(int i1, int i2)
        {
            throw new NotImplementedException();
        }

        public string concatination(string s1, string s2)
        {
            throw new NotImplementedException();
        }

        public double multiplication(double d1, double d2)
        {
            throw new NotImplementedException();
        }

        public int Add(int n1, int n2)
        {
            return n1 + n2;
        }
        public string Concatination(string s1, string s2)
        {
            return s1 + s2;
        }

        public double Multiply(double n1, double n2)
        {
            return n1 * n2;
        }
    }
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ArithmeticStringOperations
{
    public partial class Form1 : Form
    {
        Operation o=new Operation();
        public Form1()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            double res = o.Multiply(Convert.ToDouble(textBox1.Text), Convert.ToDouble(textBox4.Text));
            listBox1.Items.Add("Multiplication: " + res);
        }
        private void button1_Click(object sender, EventArgs e)
        {
            int res=o.Add(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox4.Text));
            listBox1.Items.Add("Addition: " + res);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            string res = o.concatination(textBox2.Text,textBox3.Text);
            listBox1.Items.Add("Concatination: " + res);
        }
    }
}
```

Output:



Enter Number: 10

Enter Number: 20

Enter String: Hello

Enter String: World!!

```
Addition: 30
Multiplication: 200
Concatination: Hello World!!
```