

.Net Practical (207) – Assignment-3

/*

1) Create a class Rectangle. The class has attributes length and width, each of which defaults to 1. It has member functions that calculate the perimeter and the area of the rectangle. It has set and get routines for both length and width. The set routine should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0. Develop an appropriate application that makes use of this class.

*/

Question-1 (Rectangle Class)

Enter Length of Rectangle :

Enter Width of Rectangle :

Perimeter of Rectangle :

Area of Rectangle :

Windows taskbar: Type here to search, 30°C, 09:15 PM, 07-06-2022

----- 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 Question_1
{
    public partial class frm1 : Form
    {
        public frm1()
        {
            InitializeComponent();
        }
        Rectangle r;
        private void txtCalculate_Click(object sender, EventArgs e)
        {
            if (txtLength.Text.Length > 0 && txtWidth.Text.Length > 0)
            {
                txtLength.Enabled = false;
                txtWidth.Enabled = false;
                btnCalculate.Enabled = false;

                try
                {
                    double l = Convert.ToDouble(txtLength.Text);
                    double w = Convert.ToDouble(txtWidth.Text);
                    r = new Rectangle(l, w);

                    if(r.Length == 1)
                    {
                        MessageBox.Show("Length is set to 1");
                    }
                    if(r.Width == 1)
                    {
                        MessageBox.Show("Width is set to 1");
                    }

                    txtPerimeter.Text = r.perimeter().ToString();
                    txtArea.Text = r.area().ToString();

                }
                catch
                {
                    MessageBox.Show("Error in Conversion of Length and Width..");
                }
            }
            else
            {
                MessageBox.Show("Please Enter Values of Length and Width...");
            }
        }
    }
}
```

```
private void btnReset_Click(object sender, EventArgs e)
{
    txtLength.Enabled = true;
    txtWidth.Enabled = true;
    btnCalculate.Enabled = true;

    txtLength.Text = "";
    txtWidth.Text = "";
    txtPerimeter.Text = "";
    txtArea.Text = "";
    r = null;
}
}
```

----- Rectangle.cs -----

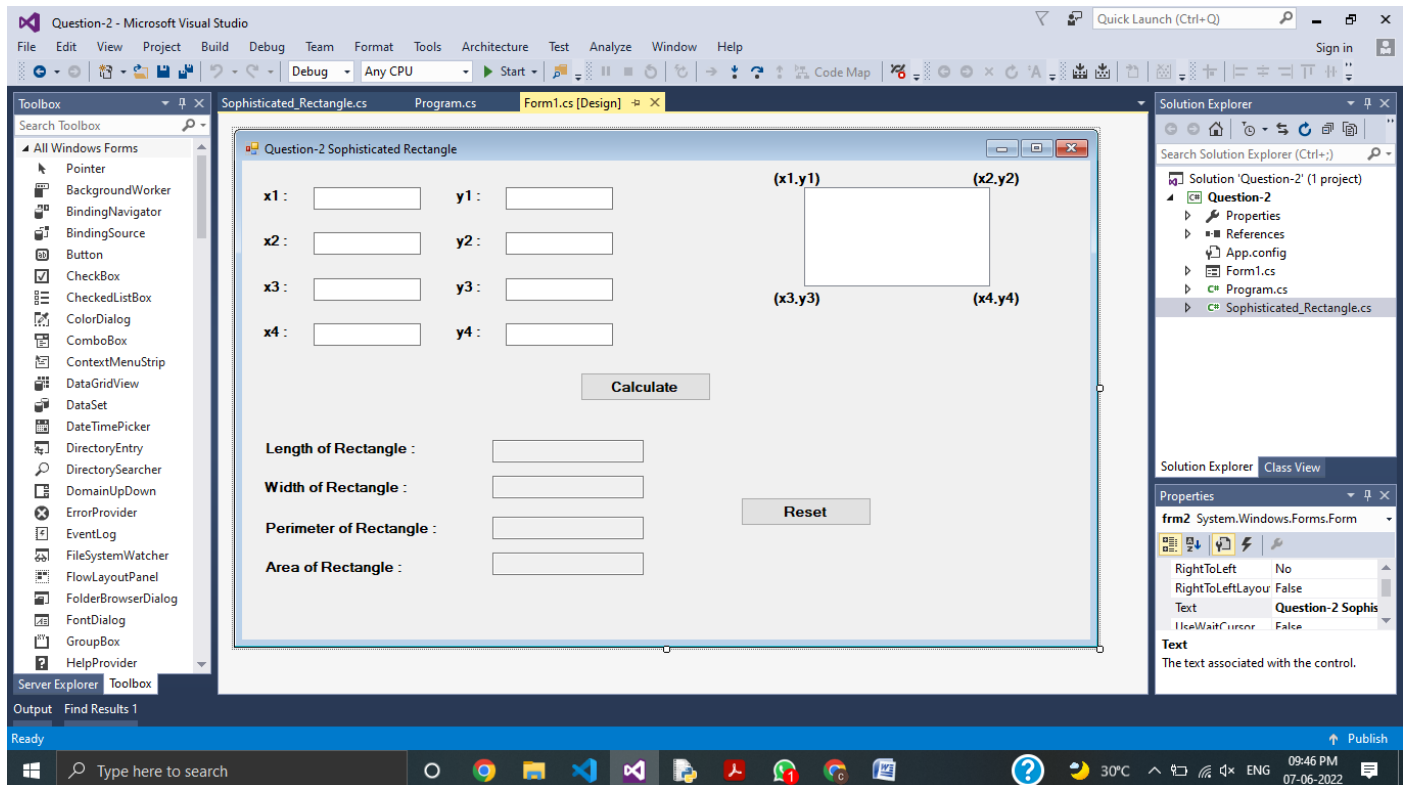
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Question_1
{
    class Rectangle
    {
        private double length;
        private double width;
        public Rectangle() { }
        public Rectangle(double l=1.0, double w =1.0)
        {
            Length = l;
            Width = w;
        }
        public double perimeter()
        {
            return 2*(this.Length + this.Width);
        }
        public double area()
        {
            return (this.Length * this.Width);
        }
        public double Length
        {
            set
            {
                length = (value > 0.0 && value < 20.0) ? value : 1;
            }
            get
            {
                return length;
            }
        }
        public double Width
        {
            set
            {
                width = (value > 0.0 && value < 20.0) ? value : 1;
            }
            get
            {
                return width;
            }
        }
    }
}
```

/*

2) Create a more sophisticated Rectangle class than the one you created in Problem 1. This class stores only the Cartesian coordinates of the four corners of the rectangle. The constructor calls a setR function that accepts four sets of coordinates and verifies that each of these is in the first quadrant with no single x or y coordinate larger than 20.0. The setR function also verifies that the supplied coordinates do, in fact, specify a rectangle. Member functions calculate the length, width, perimeter and area. The length is the larger of the two dimensions. Include a predicate function square that determines if the rectangle is a square. Develop an appropriate application that makes use of this class.

*/



----- 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 Question_2
{
    public partial class frm2 : Form
    {
        public frm2()
        {
            InitializeComponent();
        }
        Sophisticated_Rectangle r;
        private void btnCalculate_Click(object sender, EventArgs e)
        {
            if(txtx1.Text.Length > 0 && txtx1.Text.Length > 0 && txtx2.Text.Length > 0
&& txtx2.Text.Length > 0 && txtx3.Text.Length > 0 && txtx3.Text.Length > 0 &&
txtx4.Text.Length > 0 && txtx4.Text.Length > 0)
            {
                txtx1.Enabled = false;
                txtx1.Enabled = false;
                txtx2.Enabled = false;
                txtx2.Enabled = false;
                txtx3.Enabled = false;
                txtx3.Enabled = false;
                txtx4.Enabled = false;
                txtx4.Enabled = false;

                double x1, x2, x3, x4, y1, y2, y3, y4;
                try
                {

                    x1 = Convert.ToDouble(txtx1.Text);
                    x2 = Convert.ToDouble(txtx2.Text);
                    x3 = Convert.ToDouble(txtx3.Text);
                    x4 = Convert.ToDouble(txtx4.Text);
                    y1 = Convert.ToDouble(txtx1.Text);
                    y2 = Convert.ToDouble(txtx2.Text);
                    y3 = Convert.ToDouble(txtx3.Text);
                    y4 = Convert.ToDouble(txtx4.Text);
                    r = new Sophisticated_Rectangle(x1, x2, x3, x4, y1, y2, y3, y4);

                    if(r.getLength() <= 0 || r.getWidth() <= 0 )
                    {
                        MessageBox.Show(r.str);
                    }
                    else{
                        MessageBox.Show(r.str);

                        txtLength.Text = r.getLength().ToString();
                        txtWidth.Text = r.getWidth().ToString();
                        txtPerimeter.Text = r.perimeter().ToString();
                    }
                }
            }
        }
    }
}
```

```

        txtArea.Text = r.area().ToString();
    }

    }
    catch
    {
        MessageBox.Show("Error in Conversion of Length and Width..");
    }
}
else
{
    MessageBox.Show("Please Enter All Required Fields..");
}

}

private void btnReset_Click(object sender, EventArgs e)
{
    txtx1.Enabled = true;
    txty1.Enabled = true;
    txtx2.Enabled = true;
    txty2.Enabled = true;
    txtx3.Enabled = true;
    txty3.Enabled = true;
    txtx4.Enabled = true;
    txty4.Enabled = true;
    btnCalculate.Enabled = true;

    txtx1.Text = "";
    txty1.Text = "";
    txtx2.Text = "";
    txty2.Text = "";
    txtx3.Text = "";
    txty3.Text = "";
    txtx4.Text = "";
    txty4.Text = "";

    txtLength.Text = "";
    txtWidth.Text = "";
    txtPerimeter.Text = "";
    txtArea.Text = "";
    r = null;
}
}
}

```

-----Sophisticated_Rectangle.cs-----

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

namespace Question_2
{
    class Sophisticated_Rectangle
    {
        private double x1,y1,x2,y2,x3,y3,x4,y4;
        public string str = "";
        public bool flag = true;

        public Sophisticated_Rectangle() { }
        public Sophisticated_Rectangle(double x1,double x2,double x3,double x4, double
y1, double y2, double y3, double y4)
        {
            this.setR(x1, x2, x3, x4, y1, y2, y3, y4);
            this.flag = false;
        }
        private void setR(double x1, double x2, double x3, double x4, double y1,
double y2, double y3, double y4)
        {
            if (x1 <= 0.0 || x1 >= 20.0 || y1 <= 0.0 || y1 >= 20.0 || x2 <= 0.0 || x2
>= 20.0 || y2 <= 0.0 || y2 >= 20.0 || x3 <= 0.0 || x3 >= 20.0 || y3 <= 0.0 || y3 >=
20.0 || x4 <= 0.0 || x4 >= 20.0 || y4 <= 0.0 || y4 >= 20.0)
            {
                str += "Please Enter valid coordinates\n";
                this.flag = false;
            }
            else
            {
                if(isRectangle(x1, x2, x3, x4, y1, y2, y3, y4))
                {
                    this.flag = true;
                    str += "Perfect Rectangle..\n";
                    this.x1 = x1;
                    this.x2 = x2;
                    this.x3 = x3;
                    this.x4 = x4;
                    this.y1 = y1;
                    this.y2 = y2;
                    this.y3 = y3;
                    this.y4 = y4;
                    if (this.isSquare())
                    {
                        str += "Perfect Square...\n";
                    }
                    else
                    {
                        str += "Not Perfect Square...\n";
                    }
                }
            }
            else
            {
                this.flag = false;
            }
        }
    }
}
```



```

        str += "Not Perfect Rectangle...\n";
    }
}

private bool isRectangle(double x1, double x2, double x3, double x4, double
y1, double y2, double y3, double y4)
{
    double cx, cy;
    double dd1, dd2, dd3, dd4;
    cx = (x1 + x2 + x3 + x4) / 4.0;
    cy = (y1 + y2 + y3 + y4) / 4.0;

    dd1 = sqr(cx - x1) + sqr(cy - y1);
    dd2 = sqr(cx - x2) + sqr(cy - y2);
    dd3 = sqr(cx - x3) + sqr(cy - y3);
    dd4 = sqr(cx - x4) + sqr(cy - y4);

    if (dd1 == dd2 && dd1 == dd3 && dd1 == dd4)
    {
        this.flag = true;
        return true;
    }
    else
    {
        this.flag = false;
        return false;
    }
}

private bool isSquare()
{
    if (this.getLength() == this.getWidth())
    {
        return true;
    }
    else
    {
        return false;
    }
}

public double getLength()
{
    return max((this.x2 - this.x1), (this.y2 - this.y4));
}

public double getWidth()
{
    return min((this.x2 - this.x1), (this.y2 - this.y4));
}

private double max(double a, double b)
{
    return (a > b) ? a : b;
}

private double min(double a, double b)
{
    return (a < b) ? a : b;
}

private double sqr(double a)
{
    return a * a;
}

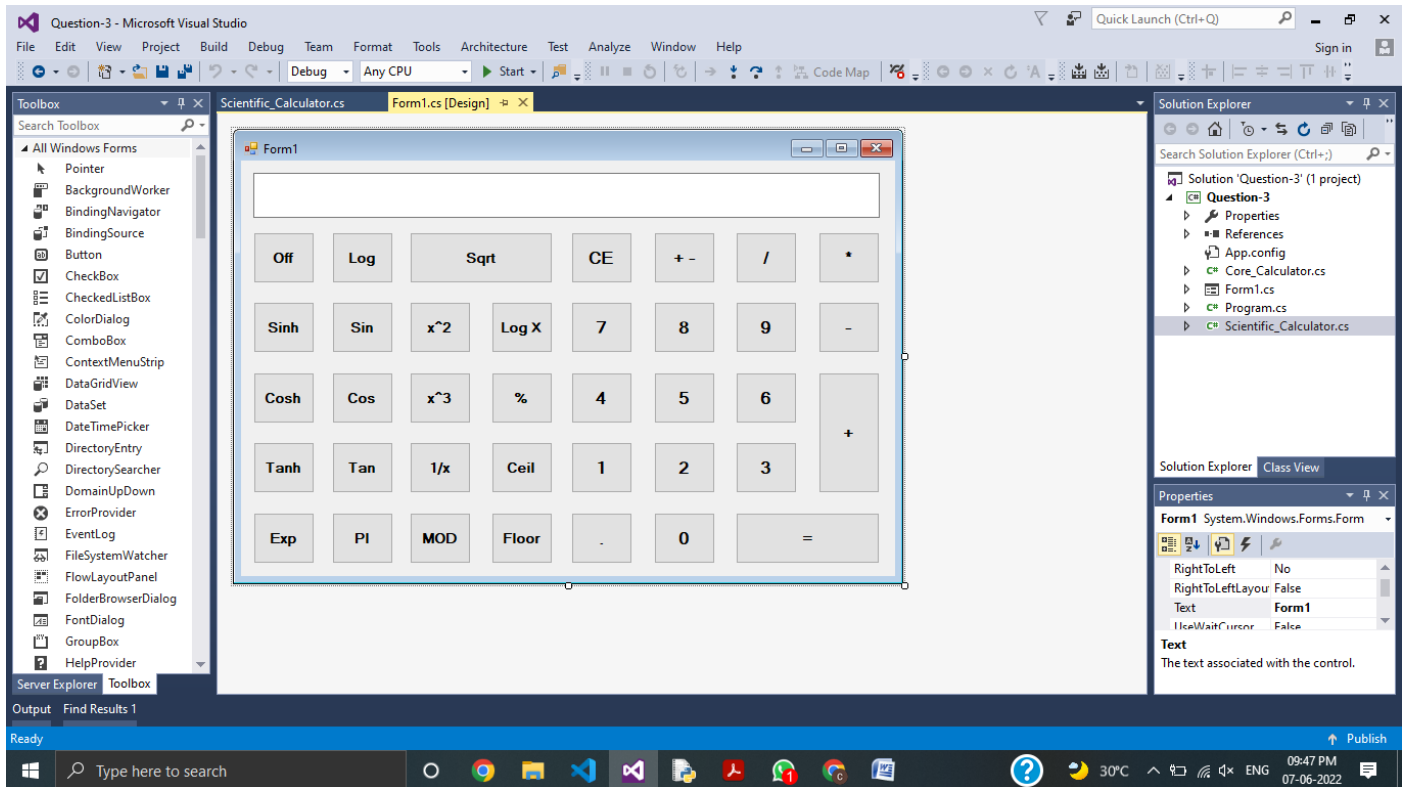
```

```
}  
public double perimeter()  
{  
    return (2 * (this.getLength() + this.getWidth()));  
}  
public double area()  
{  
    return (this.getLength() * this.getWidth());  
}  
}  
}
```

/*

3) Develop a basic calculator by creating an instance of the class `core_calculator`. Inherit the `core_calculator` class into `scientific_calculator` class and use it to develop an application for a scientific calculator.

*/



----- 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 Question_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        Scientific_Calculator sc;
        double num;
        string op;
        double lastnum = 0;
        bool Ans = true;
        private void Form1_Load(object sender, EventArgs e)
        {
            sc = new Scientific_Calculator();
        }

        private void button10_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void btnSinh_Click(object sender, EventArgs e)
        {
            if(txtAns.Text.Length > 0)
            {
                num = Convert.ToDouble(txtAns.Text);
                txtAns.Text = sc.Sinh_btn(num).ToString();
            }
        }

        private void btnLog_Click(object sender, EventArgs e)
        {
            if (txtAns.Text.Length > 0)
            {
                num = Convert.ToDouble(txtAns.Text);
                txtAns.Text = sc.Log_btn(num).ToString();
            }
        }

        private void btnSqrt_Click(object sender, EventArgs e)
        {
            if (txtAns.Text.Length > 0)
            {
                num = Convert.ToDouble(txtAns.Text);
                txtAns.Text = sc.Sqrt_btn(num).ToString();
            }
        }
    }
}
```

```

    }
}

private void btnSin_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Sin_btn(num).ToString();
    }
}

private void btnSquare_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Square_btn(num).ToString();
    }
}

private void btnLogX_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Logx_btn(num).ToString();
    }
}

private void btnCosh_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Cosh_btn(num).ToString();
    }
}

private void btnCos_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Cos_btn(num).ToString();
    }
}

private void btnCube_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Cube_btn(num).ToString();
    }
}

private void btnPer_Click(object sender, EventArgs e)
{

```

```

        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Per_btn(num).ToString();
        }
    }

    private void btnTanh_Click(object sender, EventArgs e)
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Tanh_btn(num).ToString();
        }
    }

    private void btnTan_Click(object sender, EventArgs e)
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Tan_btn(num).ToString();
        }
    }

    private void btnInverse_Click(object sender, EventArgs e)
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Inverse_btn(num).ToString();
        }
    }

    private void btnCeil_Click(object sender, EventArgs e)
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Ceil_btn(num).ToString();
        }
    }

    private void btnExp_Click(object sender, EventArgs e)
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Exp_btn(num).ToString();
        }
    }

    private void btnPI_Click(object sender, EventArgs e)
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.PI_btn(num).ToString();
        }
    }

```

```

}

private void btnMOD_Click(object sender, EventArgs e)
{
    if (lastnum == 0)
    {
        if (txtAns.Text.Length > 0)
        {
            lastnum = Convert.ToDouble(txtAns.Text);
            op = "*";
            txtAns.Text = "";
        }
    }
    else
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Mod_btn(lastnum, num).ToString();
            lastnum = Convert.ToDouble(txtAns.Text);
        }
    }
    Ans = true;
}

private void btnFloor_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Floor_btn(num).ToString();
    }
}

private void btnCE_Click(object sender, EventArgs e)
{
    txtAns.Text = "";
    num = 0;
    lastnum = 0;
}

private void button30_Click(object sender, EventArgs e)
{
}

private void btnSign_Click(object sender, EventArgs e)
{
    if (txtAns.Text.Length > 0)
    {
        num = Convert.ToDouble(txtAns.Text);
        txtAns.Text = sc.Sign_btn(num).ToString();
    }
}

private void btnMultiply_Click(object sender, EventArgs e)
{
    if (lastnum == 0)
    {

```

```

        if (txtAns.Text.Length > 0)
        {
            lastnum = Convert.ToDouble(txtAns.Text);
            op = "*";
            txtAns.Text = "";
        }
    }
    else
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Multiply_btn(lastnum, num).ToString();
            lastnum = Convert.ToDouble(txtAns.Text);
        }
    }
    Ans = true;
}

private void btnDivide_Click(object sender, EventArgs e)
{
    if (lastnum == 0)
    {
        if (txtAns.Text.Length > 0)
        {
            lastnum = Convert.ToDouble(txtAns.Text);
            op = "/";
            txtAns.Text = "";
        }
    }
    else
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Divide_btn(lastnum, num).ToString();
            lastnum = Convert.ToDouble(txtAns.Text);
        }
    }
    Ans = true;
}

private void btnSubtract_Click(object sender, EventArgs e)
{
    if (lastnum == 0)
    {
        if (txtAns.Text.Length > 0)
        {
            lastnum = Convert.ToDouble(txtAns.Text);
            op = "-";
            txtAns.Text = "";
        }
    }
    else
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Subtract_btn(lastnum, num).ToString();

```



```

        lastnum = Convert.ToDouble(txtAns.Text);
    }
}
Ans = true;
}

private void btnAdd_Click(object sender, EventArgs e)
{
    if(lastnum == 0)
    {
        if (txtAns.Text.Length > 0)
        {
            lastnum = Convert.ToDouble(txtAns.Text);
            op = "+";
            txtAns.Text = "";
        }
    }
    else
    {
        if (txtAns.Text.Length > 0)
        {
            num = Convert.ToDouble(txtAns.Text);
            txtAns.Text = sc.Add_btn(lastnum, num).ToString();
            lastnum = Convert.ToDouble(txtAns.Text);
        }
    }
    Ans = true;
}

private void btnAns_Click(object sender, EventArgs e)
{
    if (!Ans)
    {
        switch (op)
        {
            case "+":
            {
                try
                {
                    num = Convert.ToDouble(txtAns.Text);
                }
                catch
                {
                    num = 0;
                }
                txtAns.Text = sc.Add_btn(lastnum, num).ToString();
                lastnum = Convert.ToDouble(txtAns.Text);
            }
            break;
            case "-":
            {
                num = Convert.ToDouble(txtAns.Text);
                txtAns.Text = sc.Subtract_btn(lastnum, num).ToString();
                lastnum = Convert.ToDouble(txtAns.Text);
            }
            break;
            case "*":
            {
                num = Convert.ToDouble(txtAns.Text);

```

```

        txtAns.Text = sc.Multiply_btn(lastnum, num).ToString();
        lastnum = Convert.ToDouble(txtAns.Text);
    }
    break;
case "/":
{
    num = Convert.ToDouble(txtAns.Text);
    txtAns.Text = sc.Divide_btn(lastnum, num).ToString();
    lastnum = Convert.ToDouble(txtAns.Text);
}
break;
case "%":
{
    num = Convert.ToDouble(txtAns.Text);
    txtAns.Text = sc.Mod_btn(lastnum, num).ToString();
    lastnum = Convert.ToDouble(txtAns.Text);
}
break;
    }
}
else
{
    //Ans = true;
    lastnum = Convert.ToDouble(txtAns.Text);
}
Ans = true;
}

private void btn_Click(object sender, EventArgs e)
{
    Button b = (Button)sender;
    if(Ans)
    {
        txtAns.Text = b.Text;
    }
    else
    {
        txtAns.Text += b.Text;
    }

    Ans = false;
}

private void btnDot_Click(object sender, EventArgs e)
{
    if (!txtAns.Text.Contains("."))
    {
        btn_Click(sender, e);
    }
}

private void btn0_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

```

```
private void btn1_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn2_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn3_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn4_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn5_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn6_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn7_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn8_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}

private void btn9_Click(object sender, EventArgs e)
{
    btn_Click(sender, e);
}
}
}
```

----- Core_Calculator.cs -----

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

namespace Question_3
{
    class Core_Calculator
    {
        public double Add_btn(double n1,double n2)
        {
            return n1 + n2;
        }
        public double Subtract_btn(double n1, double n2)
        {
            return n1 - n2;
        }
        public double Multiply_btn(double n1, double n2)
        {
            return n1 * n2;
        }
        public double Divide_btn(double n1, double n2)
        {
            return n1 / n2;
        }
    }
}
```

-----Scientific_Calculator.cs-----

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

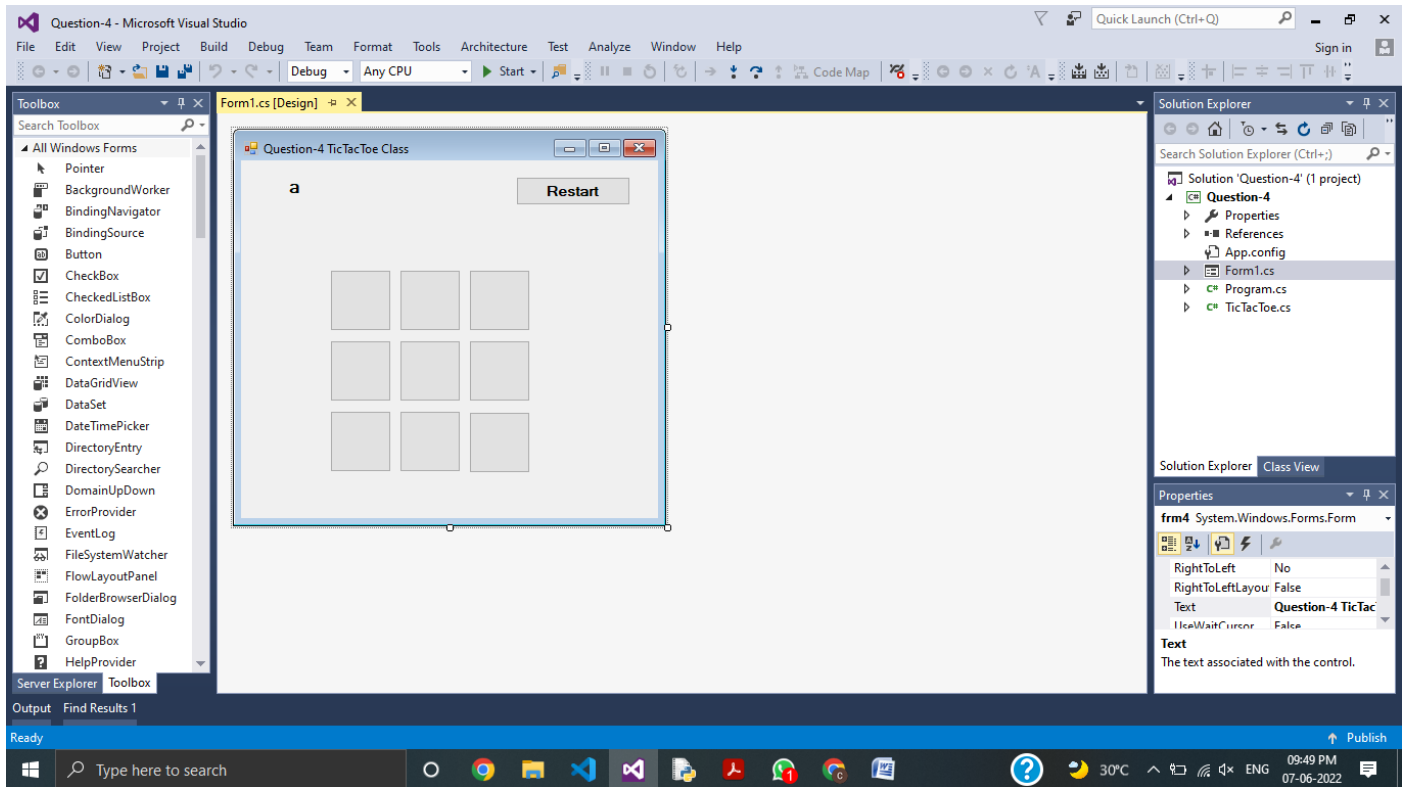
namespace Question_3
{
    class Scientific_Calculator:Core_Calculator
    {
        public double Sign_btn(double n1)
        {
            return n1 * -1;
        }
        public double Sqrt_btn(double n1)
        {
            return Math.Sqrt(n1);
        }
        public double Per_btn(double n1)
        {
            return n1 / 100;
        }
        public double Ceil_btn(double n1)
        {
            return Math.Ceiling(n1);
        }
        public double Floor_btn(double n1)
        {
            return Math.Floor(n1);
        }
        public double Square_btn(double n1)
        {
            return Math.Pow(n1,2);
        }
        public double Cube_btn(double n1)
        {
            return Math.Pow(n1,3);
        }
        public double Inverse_btn(double n1)
        {
            return 1/n1;
        }
        public double Mod_btn(double n1,double n2)
        {
            return n1 % n2;
        }
        public double Log_btn(double n1)
        {
            return Math.Log10(n1);
        }
        public double Sin_btn(double n1)
        {
            return Math.Sin(n1);
        }
        public double Cos_btn(double n1)
        {
            return Math.Cos(n1);
        }
    }
}
```

```
public double Tan_btn(double n1)
{
    return Math.Tan(n1);
}
public double PI_btn(double n1)
{
    return Math.PI;
}
public double Exp_btn(double n1)
{
    return Math.Exp(n1);
}
public double Sinh_btn(double n1)
{
    return Math.Sinh(n1);
}
public double Cosh_btn(double n1)
{
    return Math.Cosh(n1);
}
public double Tanh_btn(double n1)
{
    return Math.Tanh(n1);
}
public double Logx_btn(double n1)
{
    return Math.Log(n1);
}
}
}
```

/*

4) Develop a class tic tac toe and develop an application using it to simulate a tic tac toe game.

*/



----- 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 Question_4
{
    public partial class frm4 : Form
    {
        public frm4()
        {
            InitializeComponent();
            btnRestart.Visible = false;
            txtWinner.Text = "";
        }
        TicTacToe ttt = new TicTacToe();
        Button[] btn = new Button[9];

        private void gameStatus(string ans)
        {
            if(!(txtWinner.Text.Equals("")))
            {
                MessageBox.Show(ans);

                btn0.Enabled = false;
                btn1.Enabled = false;
                btn2.Enabled = false;
                btn3.Enabled = false;
                btn4.Enabled = false;
                btn5.Enabled = false;
                btn6.Enabled = false;
                btn7.Enabled = false;
                btn8.Enabled = false;

                btnRestart.Visible = true;
            }
        }

        private void btn0_Click(object sender, EventArgs e)
        {
            Button b = (Button)sender;
            b.Text = ttt.btn_Click(0);
            b.Enabled = false;
            btn0 = b;
            txtWinner.Text = ttt.check_win();
            gameStatus(txtWinner.Text);
        }

        private void btn1_Click(object sender, EventArgs e)
        {
            Button b = (Button)sender;
            b.Text = ttt.btn_Click(1);
            b.Enabled = false;
        }
    }
}
```



```

        btn1 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn2_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(2);
        b.Enabled = false;
        btn2 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn3_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(3);
        b.Enabled = false;
        btn3 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn4_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(4);
        b.Enabled = false;
        btn4 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn5_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(5);
        b.Enabled = false;
        btn5 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn6_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(6);
        b.Enabled = false;
        btn6 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn7_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(7);
    }

```

```

        b.Enabled = false;
        btn7 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btn8_Click(object sender, EventArgs e)
    {
        Button b = (Button)sender;
        b.Text = ttt.btn_Click(8);
        b.Enabled = false;
        btn8 = b;
        txtWinner.Text = ttt.check_win();
        gameStatus(txtWinner.Text);
    }

    private void btnRestart_Click(object sender, EventArgs e)
    {
        ttt = null;
        ttt = new TicTacToe();

        btn0.Text = "";
        btn1.Text = "";
        btn2.Text = "";
        btn3.Text = "";
        btn4.Text = "";
        btn5.Text = "";
        btn6.Text = "";
        btn7.Text = "";
        btn8.Text = "";

        btn0.Enabled = true;
        btn1.Enabled = true;
        btn2.Enabled = true;
        btn3.Enabled = true;
        btn4.Enabled = true;
        btn5.Enabled = true;
        btn6.Enabled = true;
        btn7.Enabled = true;
        btn8.Enabled = true;

        btnRestart.Visible = false;

        txtWinner.Text = "";
    }

    /*
    public void create_button(int n)
    {
        int top = 100;
        int left = 100;
        for (int i = 0, k = 0; i < n && k < n * n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                Button b = new Button();
                b.Text = k.ToString();
                b.Width = 50;
            }
        }
    }
    */

```

```
        b.Height = 50;
        b.Top = top;
        b.Left = left;
        b.Visible = true;
        b.Enabled = true;

        btn[k] = b;

        left += 60;
    }
    top += 60;
    left = 100;
}
for (int i = 1; i <= n * n; i++)
{
    this.Controls.Add(btn[i]);
}
}
*/
}
}
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Question_4
{
    class TicTacToe
    {
        public string[] arr;
        bool turn;
        bool game;
        string win = "";
        string xo;
        public TicTacToe()
        {
            arr = new string[9];
            for(int i = 0; i < 9; i++)
            {
                arr[i] = " ";
            }
            game = true;
            turn = true;
            xo = "X";
        }
        public string btn_Click(int n)
        {
            if (turn)
            {
                arr[n] = "X";
            }
            else
            {
                arr[n] = "O";
            }
            turn = !turn;
            return arr[n];
        }
        public string check_win()
        {
            //horizontal
            if(!(arr[0].Equals(" ")) && arr[0].Equals(arr[1]) &&
arr[1].Equals(arr[2]))
            {
                win = "Winner : "+arr[0];
            }
            else if (!(arr[3].Equals(" ")) && arr[3].Equals(arr[4]) &&
arr[4].Equals(arr[5]))
            {
                win = "Winner : " + arr[3];
            }
            else if (!(arr[6].Equals(" ")) && arr[6].Equals(arr[7]) &&
arr[7].Equals(arr[8]))
            {
                win = "Winner : " + arr[6];
            }
        }
    }
}

```

```

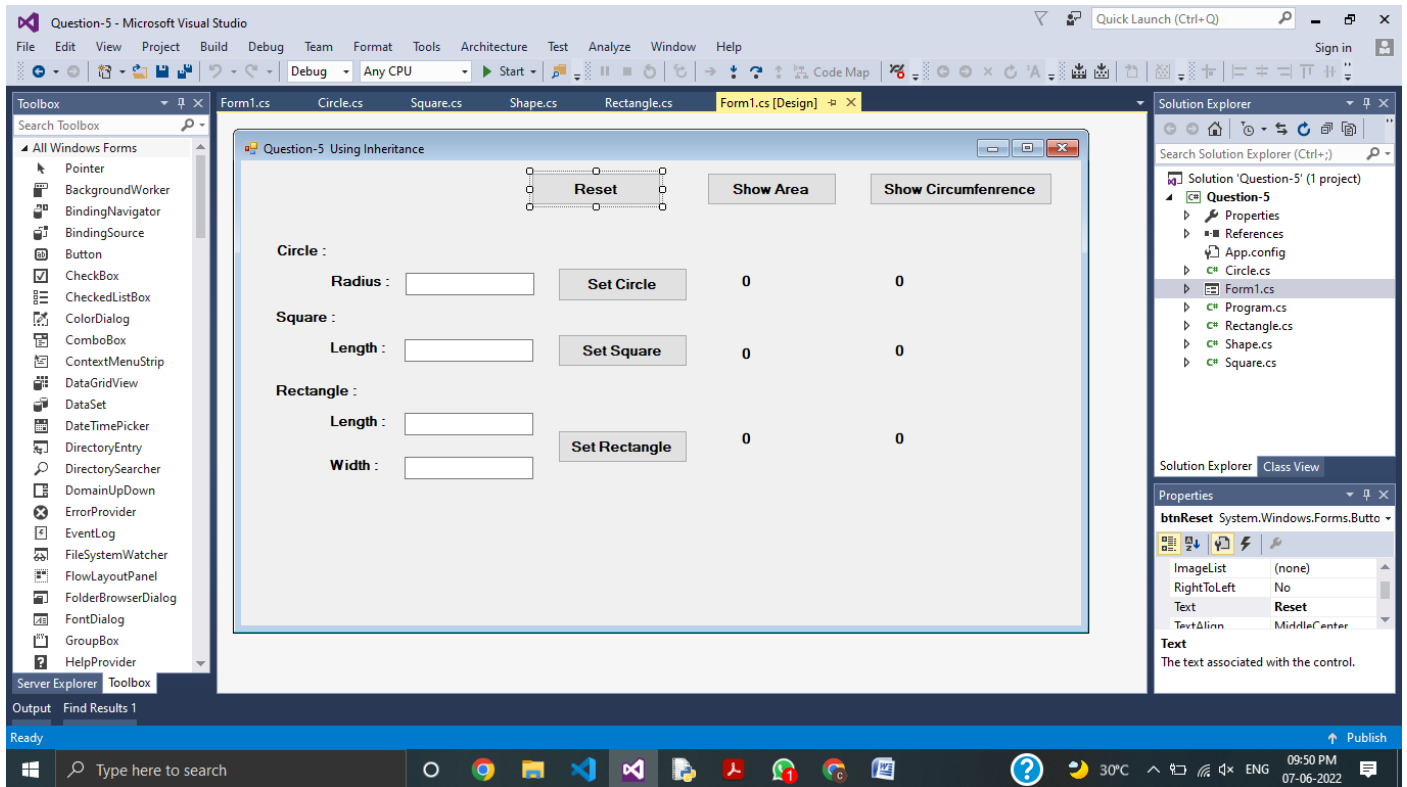
        //vertical
        else if (!(arr[0].Equals(" ")) && arr[0].Equals(arr[3]) &&
arr[3].Equals(arr[6]))
        {
            win = "Winner : " + arr[0];
        }
        else if (!(arr[1].Equals(" ")) && arr[1].Equals(arr[4]) &&
arr[4].Equals(arr[7]))
        {
            win = "Winner : " + arr[1];
        }
        else if (!(arr[2].Equals(" ")) && arr[2].Equals(arr[5]) &&
arr[5].Equals(arr[8]))
        {
            win = "Winner : " + arr[2];
        }
        //diagonal
        else if (!(arr[0].Equals(" ")) && arr[0].Equals(arr[4]) &&
arr[4].Equals(arr[8]))
        {
            win = "Winner : " + arr[0];
        }
        else if (!(arr[2].Equals(" ")) && arr[2].Equals(arr[4]) &&
arr[4].Equals(arr[6]))
        {
            win = "Winner : " + arr[2];
        }
        else if (!(arr[0].Equals(" ")) && !(arr[1].Equals(" ")) &&
!(arr[2].Equals(" ")) && !(arr[3].Equals(" ")) && !(arr[4].Equals(" ")) &&
!(arr[5].Equals(" ")) && !(arr[6].Equals(" ")) && !(arr[7].Equals(" ")) &&
!(arr[8].Equals(" ")))
        {
            win = "Game Over";
        }
        return win;
    }
}
}

```

/*

5) Create classes using real world entities which use all the keywords available in C# with regards to inheritance.

*/



----- 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 Question_5
{
    public partial class frm5 : Form
    {
        public frm5()
        {
            InitializeComponent();
            btnReset.Visible = false;
        }
        double r = 0, sl = 0, l = 0, w = 0;
        Circle c;
        Square s;
        Rectangle re;

        private void btnCircle_Click(object sender, EventArgs e)
        {
            if (txtRadius.Text.Length > 0)
            {
                try
                {
                    r = Convert.ToDouble(txtRadius.Text);
                    if(r > 0)
                    {
                        c = new Circle(r);
                        txtRadius.Enabled = false;
                    }
                    else
                    {
                        MessageBox.Show("Please enter Value of Radius Greater than 0....");
                    }
                }
                catch
                {
                    MessageBox.Show("Please ENTER Valid Value for Radius...");
                }
            }
            else
            {
                MessageBox.Show("Please ENTER Value of Radius...");
            }
        }

        private void btnSquare_Click(object sender, EventArgs e)
        {
            if(txtSLength.Text.Length > 0)
```

```

{
    try
    {
        s1 = Convert.ToDouble(txtSLength.Text);
        if(s1 > 0)
        {
            s = new Square(s1);
            txtSLength.Enabled = false;
        }
        else
        {
            MessageBox.Show("Please enter Value of Length Greater than
0....");
        }
    }
    catch
    {
        MessageBox.Show("Please ENTER valid Value of Length...");
    }
}
else
{
    MessageBox.Show("Please enter value of Length...");
}
}

private void btnRectangle_Click(object sender, EventArgs e)
{
    if (txtLength.Text.Length > 0 && txtWidth.Text.Length > 0)
    {
        try
        {
            l = Convert.ToDouble(txtLength.Text);
            w = Convert.ToDouble(txtWidth.Text);
            if(l > 0 && w > 0)
            {
                re = new Rectangle(l, w);
                txtLength.Enabled = false;
                txtWidth.Enabled = false;
            }
            else
            {
                MessageBox.Show("Please enter Value of Length and Width
Greater than 0....");
            }
        }
        catch
        {
            MessageBox.Show("Please ENTER valid Value of Length and
Width...");
        }
    }
    else
    {
        MessageBox.Show("Please enter value of Length and Width...");
    }
}

private void btnArea_Click(object sender, EventArgs e)

```



```

{
    if (r > 0 && sl > 0 && l > 0 && w > 0)
    {
        lblCircleArea.Text = Math.Round(c.Area(),2).ToString();
        lblSquareArea.Text = Math.Round(s.Area(),2).ToString();
        lblRectangleArea.Text = Math.Round(re.Area(),2).ToString();

        btnReset.Visible = true;
    }
    else
    {
        MessageBox.Show("Please Enter Details and Set them then click to find Area...");
    }
}

private void btnCircumference_Click(object sender, EventArgs e)
{
    if (r > 0 && sl > 0 && l > 0 && w > 0)
    {
        lblCircleCircumference.Text =
Math.Round(c.Circumference(),2).ToString();
        lblSquareCircumference.Text =
Math.Round(s.Circumference(),2).ToString();
        lblRectangleCircumference.Text =
Math.Round(re.Circumference(),2).ToString();

        btnReset.Visible = true;
    }
    else
    {
        MessageBox.Show("Please Enter Details and Set them then click to find Circumference...");
    }
}

private void btnReset_Click(object sender, EventArgs e)
{
    lblCircleArea.Text = "0";
    lblSquareArea.Text = "0";
    lblRectangleArea.Text = "0";

    lblCircleCircumference.Text = "0";
    lblSquareCircumference.Text = "0";
    lblRectangleCircumference.Text = "0";

    txtRadius.Enabled = true;
    txtSLength.Enabled = true;
    txtLength.Enabled = true;
    txtWidth.Enabled = true;

    txtRadius.Text = "";
    txtSLength.Text = "";
    txtLength.Text = "";
    txtWidth.Text = "";

    btnReset.Visible = false;
}
}

```


-----Shape.cs-----

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

namespace Question_5
{
    public abstract class Shape
    {
        public virtual double Area()
        {
            return 0;
        }
        public virtual double Circumference()
        {
            return 0;
        }
    }
}
```

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

namespace Question_5
{
    class Circle : Shape
    {
        private double radius;

        public Circle(double r=1)
        {
            Radius = r;
        }

        public double Radius
        {
            set
            {
                if(value > 0)
                {
                    radius = value;
                }
                else
                {
                    radius = 1;
                }
            }
            get
            {
                return radius;
            }
        }
        public override double Area()
        {
            return (Math.PI * Math.Pow(Radius, 2));
        }
        public override double Circumference()
        {
            return (Math.PI * 2 * Radius);
        }
    }
}
```

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

namespace Question_5
{
    class Square : Shape
    {
        private double length;

        public Square(double l=1)
        {
            Length = 1;
        }

        public double Length
        {
            set
            {
                if (value > 0)
                {
                    length = value;
                }
                else
                {
                    length = 1;
                }
            }
            get
            {
                return length;
            }
        }
        public override double Area()
        {
            return (Math.Pow(Length, 2));
        }
        public override double Circumference()
        {
            return (4 * Length);
        }
    }
}
```

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

namespace Question_5
{
    class Rectangle : Shape
    {
        private double length;
        private double width;

        public Rectangle()
        {
            Length = 1;
            Width = 1;
        }
        public Rectangle(double l, double w)
        {
            Length = l;
            Width = w;
        }

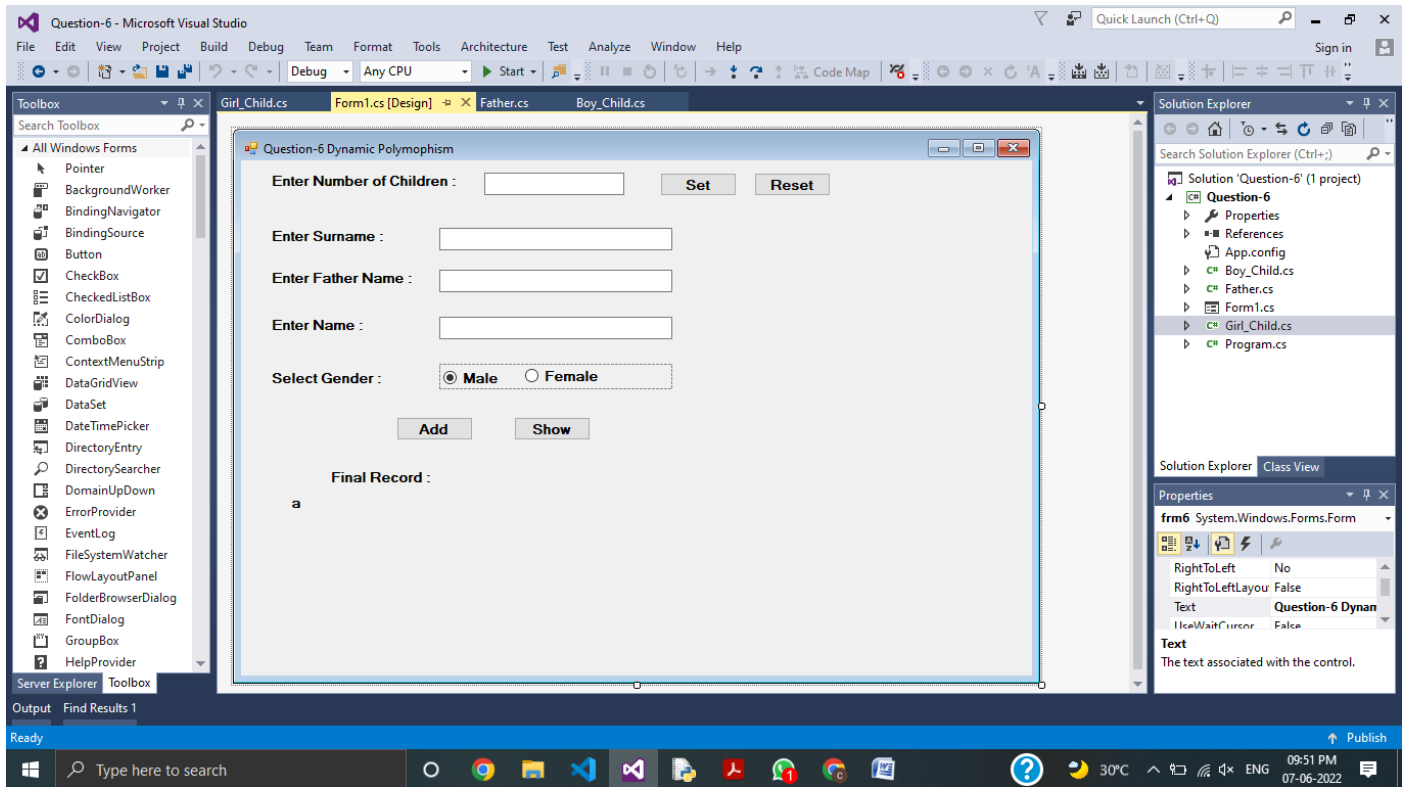
        public double Length
        {
            set
            {
                if (value > 0)
                {
                    length = value;
                }
                else
                {
                    length = 1;
                }
            }
            get
            {
                return length;
            }
        }
        public double Width
        {
            set
            {
                if (value > 0)
                {
                    width = value;
                }
                else
                {
                    width = 1;
                }
            }
            get
            {
                return width;
            }
        }
    }
}
```

```
    }  
}  
public override double Area()  
{  
    return (Length * Width);  
}  
public override double Circumference()  
{  
    return (2 * (Length+Width));  
}  
}  
}
```

/*

6) Demonstrate Dynamic Polymorphism by taking an example involving real world entities.

*/



----- 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 Question_6
{
    public partial class frm6 : Form
    {
        public frm6()
        {
            InitializeComponent();
            txtN.Text = "";
            txtSurname.Text = "";
            txtFName.Text = "";
            txtName.Text = "";
            rdbMale.Checked = true;
            rdbFemale.Checked = false;
            lblFinal.Text = "";

            txtN.Enabled = true;
            txtSurname.Enabled = false;
            txtFName.Enabled = false;
            txtName.Enabled = false;
            rdbFemale.Enabled = false;
            rdbMale.Enabled = false;

            btnReset.Visible = false;
            btnSet.Visible = true;
            btnShow.Visible = false;
            btnAdd.Visible = false;
        }
        Father[] f;
        int i = 0;
        int n;
        private void btnSet_Click(object sender, EventArgs e)
        {
            if(txtN.Text.Length > 0)
            {
                try
                {
                    n = Convert.ToInt32(txtN.Text);
                    if(n>0 && n < 10)
                    {
                        f = new Father[n];

                        txtN.Enabled = false;
                        txtSurname.Enabled = true;
                        txtFName.Enabled = true;
                        txtName.Enabled = true;
                        rdbMale.Enabled = true;
                        rdbFemale.Enabled = true;
                    }
                }
            }
        }
    }
}
```

```

        txtSurname.Text = "";
        txtFName.Text = "";
        txtName.Text = "";

        btnSet.Visible = false;
        btnReset.Visible = true;
        btnAdd.Visible = true;
        btnShow.Visible = true;
    }
    else
    {
        MessageBox.Show("Please ENter value of N in range of 1 to
10...");
    }
}
catch
{
    MessageBox.Show("Please ENter valid Number for N....");
    return;
}
}
else
{
    MessageBox.Show("Please ENter Value of N....");
    return;
}
}

private void btnReset_Click(object sender, EventArgs e)
{
    txtN.Text = "";
    txtSurname.Text = "";
    txtFName.Text = "";
    txtName.Text = "";
    rdbMale.Checked = true;
    rdbFemale.Checked = false;
    lblFinal.Text = "";

    rdbMale.Enabled = true;
    rdbFemale.Enabled = true;
    txtN.Enabled = true;
    txtSurname.Enabled = true;
    txtFName.Enabled = true;
    txtName.Enabled = true;

    btnReset.Visible = false;
    btnSet.Visible = true;
    btnShow.Visible = false;
    btnAdd.Visible = false;
}

private void btnAdd_Click(object sender, EventArgs e)
{
    string surname = txtSurname.Text;
    string fname = txtFName.Text;
    string name = txtName.Text;

    if (i < n)
    {

```

```

if (surname.Length > 0 && fname.Length > 0 && name.Length > 0)
{
    /*
    txtSurname.Enabled = false;
    txtFName.Enabled = false;
    txtName.Enabled = false;
    rdbMale.Enabled = false;
    rdbFemale.Enabled = false;

    btnReset.Visible = true;
    btnSet.Visible = false;
    */

    if (rdbMale.Checked == true)
    {
        Boy_Child b = new Boy_Child(surname, name, fname);
        // lblAns.Text = b.display();

        f[i] = b;
        MessageBox.Show(f[i].display());
        i++;

    }
    else
    {
        Girl_Child g = new Girl_Child(surname, name, fname);
        // lblAns.Text = g.display();

        f[i] = g;
        MessageBox.Show(f[i].display());
        i++;

    }
}
else
{
    MessageBox.Show("Please Fill All TextBoxes...");
}
}
else
{
    btnAdd.Enabled = false;
    btnAdd.Visible = false;
    btnShow.Visible = true;
    btnShow.Enabled = true;
    MessageBox.Show("Array is Full....");
}
}

private void btnShow_Click(object sender, EventArgs e)
{
    String str = "";
    for(int temp = 0; temp < i; temp++)
    {
        str += f[temp].display() + "\r\n";
    }
    /*
    foreach(Father temp in f)

```

```
        {
            str += temp.display() + "\r\n";
        }
        */
        lblFinal.Visible = true;
        lblFinal.Text = str;
    }
}
```

----- Father.cs -----

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

namespace Question_6
{
    class Father
    {
        protected string surname, fname;
        public Father(string surname, string fname)
        {
            this.surname = surname;
            this.fname = fname;
        }
        public virtual string display()
        {
            return fname;
        }
    }
}
```

----- Boy_Child.cs -----

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

namespace Question_6
{
    class Boy_Child:Father
    {
        private string name;
        public Boy_Child(string surname,string name,string fname) :
base(surname,fname)
        {
            this.name = name;
        }
        public override string display()
        {
            return "Mr. " + name + " " + fname + " " + surname;
        }
    }
}
```

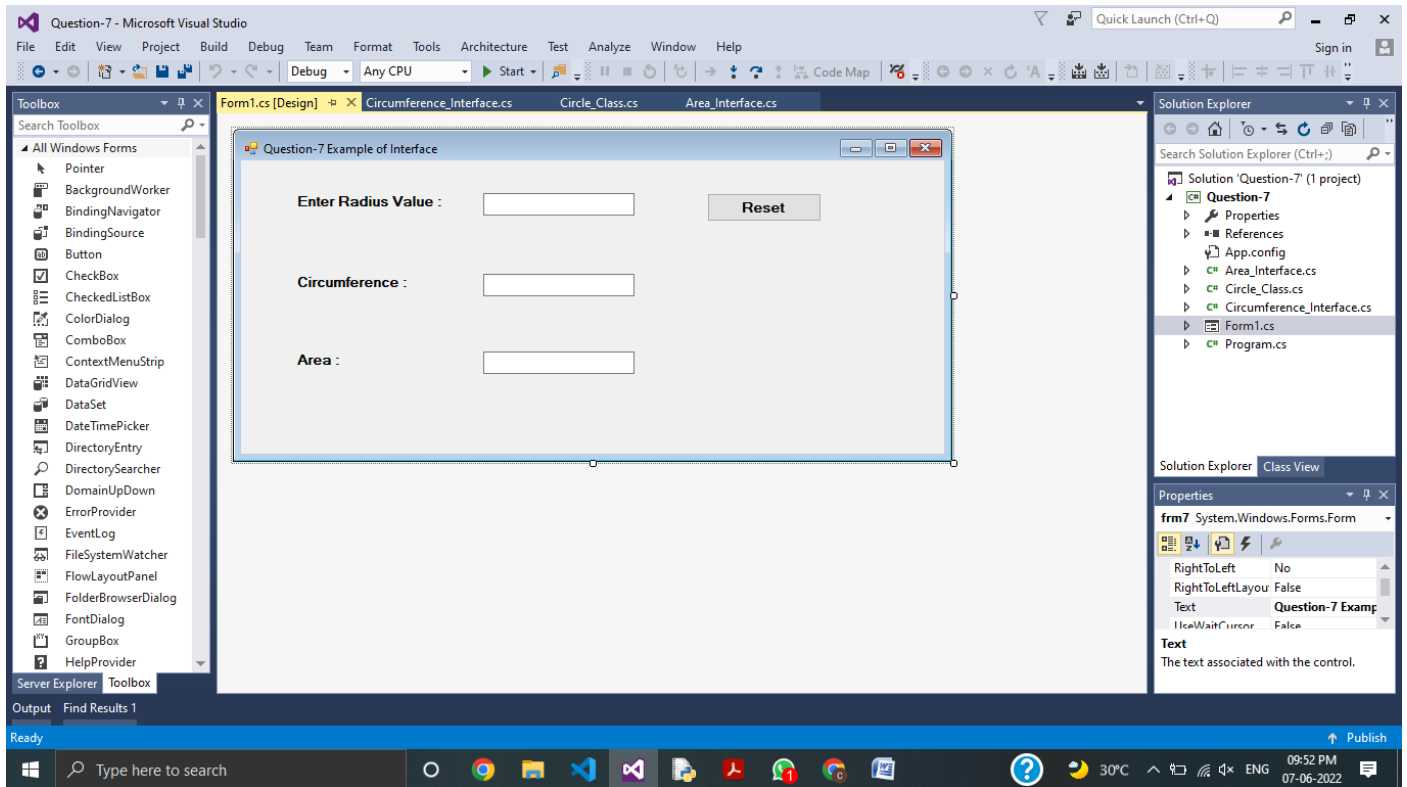
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Question_6
{
    class Girl_Child : Father
    {
        private string name;
        public Girl_Child(string surname, string name, string fname) :
base(surname,fname)
        {
            this.name = name;
        }
        public override string display()
        {
            return "Mis. " + name + " "+ fname + " "+ surname;
        }
    }
}
```

/*

7) Demonstrate the usage of interface taking an example which removes anomalies of multiple inheritance.

*/



----- 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 Question_7
{
    public partial class frm7 : Form
    {
        public frm7()
        {
            InitializeComponent();
            txtCircumference.Enabled = false;
            txtArea.Enabled = false;
            btnReset.Visible = false;
            btnSet.Visible = true;
        }
        Circle_Class c;
        private void btnSet_Click(object sender, EventArgs e)
        {
            if(txtRadius.Text.Length > 0)
            {
                try
                {
                    txtRadius.Enabled = false;
                    btnSet.Visible = false;
                    btnReset.Visible = true;

                    double r = Convert.ToDouble(txtRadius.Text);
                    c = new Circle_Class(r);
                    txtCircumference.Text =
Math.Round(c.Circumference(),2).ToString();
                    txtArea.Text = Math.Round(c.Area(),2).ToString();
                }
                catch
                {
                    MessageBox.Show("Please ENTER Valid Value of Radius...");
                }
            }
            else
            {
                MessageBox.Show("Please ENTER Value of Radius...");
            }
        }

        private void btnReset_Click(object sender, EventArgs e)
        {
            txtRadius.Text = "";
            txtCircumference.Text = "";
            txtArea.Text = "";
        }
    }
}
```

```
        txtRadius.Enabled = true;
        btnSet.Visible = true;
        btnReset.Visible = false;
    }
}
```

----- Circumference_Interface.cs -----

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

namespace Question_7
{
    public interface Circumference_Interface
    {
        double Circumference();
    }
}
```

----- Area_Interface.cs -----

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

namespace Question_7
{
    public interface Area_Interface
    {
        double Area();
    }
}
```

----- Circle_Class.cs -----

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

namespace Question_7
{
    class Circle_Class : Circumference_Interface, Area_Interface
    {
        private double radius;
        public Circle_Class(double r)
        {
            this.radius = r;
        }
        public double Area()
        {
            return (Math.PI * Math.Pow(radius, 2));
        }
        public double Circumference()
        {
            return (2 * Math.PI * radius);
        }
    }
}
```

/*

8) Demonstrate the usage of Delegates.

*/

Question-8 Delegate Example

Enter Value of N :

Input : 6 89 92 73 50 66 21 42 11 2 52 62 66 33 42

Ascending Sort : 2 6 11 21 33 42 42 50 52 62 66 66 73 89 92

Descending Sort : 92 89 73 66 66 62 52 50 42 42 33 21 11 6 2

Windows taskbar: Type here to search, 30°C, 09:54 PM, 07-06-2022

----- 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 Question_8
{
    public partial class frm8 : Form
    {
        public frm8()
        {
            InitializeComponent();
            btnReset.Visible = false;
        }
        private int[] arr;
        private void btnSet_Click(object sender, EventArgs e)
        {
            if(txtN.Text.Length > 0)
            {
                try
                {
                    int n = Convert.ToInt32(txtN.Text);
                    if (n > 0 && n<=20)
                    {
                        arr = new int[n];

                        lblInput.Text = "";
                        lblASort.Text = "";
                        lblDSort.Text = "";

                        Random rnd = new Random();
                        for(int i = 0; i < n; i++)
                        {
                            arr[i] = rnd.Next(100);
                            lblInput.Text += arr[i] + " ";
                        }
                        ASort();
                        DSort();

                        txtN.Enabled = false;
                        btnSet.Visible = false;
                        btnReset.Visible = true;
                    }
                    else
                    {
                        MessageBox.Show("Please enter valid positive Number from 1 to
20");
                    }
                }
                catch
                {
                    MessageBox.Show("Please ENter Valid Value for N.....");
                    return;
                }
            }
        }
    }
}
```

```

        }

    }

}

private bool SortAscending(int ele1,int ele2)
{
    return ele1 > ele2;
}
private void ASort()
{
    DelegateSort.SortArray(arr, new DelegateSort.Comparator(SortAscending));
    DisplayAResult();
}
private bool SortDescending(int ele1,int ele2)
{
    return ele1 < ele2;
}
private void DSort()
{
    DelegateSort.SortArray(arr, new DelegateSort.Comparator(SortDescending));
    DisplayDResult();
}
private void DisplayAResult()
{
    lblASort.Text = "";
    foreach(int i in arr)
    {
        lblASort.Text += i + " ";
    }
}
private void DisplayDResult()
{
    lblDSort.Text = "";
    foreach (int i in arr)
    {
        lblDSort.Text += i + " ";
    }
}

private void btnReset_Click(object sender, EventArgs e)
{
    txtN.Enabled = true;
    btnSet.Visible = true;
    btnReset.Visible = false;

    lblInput.Text = "";
    lblASort.Text = "";
    lblDSort.Text = "";
    txtN.Text = "";
}
}
}

```


----- DelegateSort.cs -----

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

namespace Question_8
{
    class DelegateSort
    {
        public delegate bool Comparator(int element1, int element2);
        public static void SortArray(int[] array, Comparator Compare)
        {
            for(int i = 0; i < array.Length; i++)
            {
                for(int j = 0; j < array.Length-1; j++)
                {
                    if (Compare(array[j], array[j + 1]))
                    {
                        swap(ref array[j], ref array[j + 1]);
                    }
                }
            }
        }
        private static void swap(ref int fele, ref int sele)
        {
            int temp = fele;
            fele = sele;
            sele = temp;
        }
    }
}
```