

Example (1)

Method Overloading

1. Write a C# windows Form Application code to create a Class Named **Rectangle** that contains the following :

✓ **Fields :**

- Length , Width of type double

✓ **Methods:**

- Default and parametrized constructor .
- Set and get Methods .
- Area and diameters .
- Type to determine is it a square or rectangle (private to show Encapsulation)
- Return Details method.

Solution:

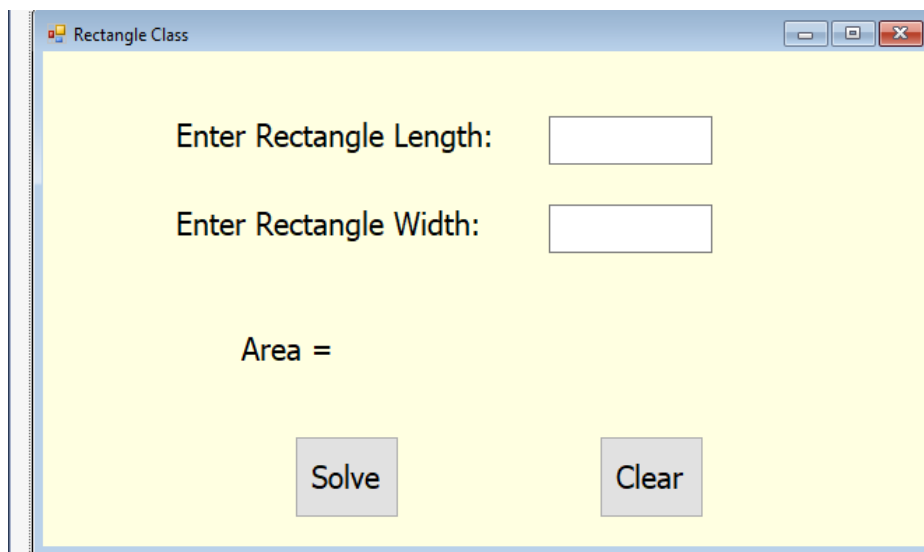
- ✓ **First add a class named Rectangle :**

In Rectangle.CS write the following

```
class Rectangle {
    private double Length;
    private double Width;
    // default constructor
    public Rectangle() {
        Length = 0;
        Width = 0;
    }
    // paramaterized constructor
    public Rectangle(int L, int W) {
        Length = L;
        Width = W;
    }
    public Rectangle( double L, double W){
        Length = L;
        Width = W;
    }
    public Rectangle(double L){
        Length = L;
        Width = L;
    }
    public Rectangle(int L) {
        Length = L;
        Width = L;
    }
}
```

```
}  
public Rectangle(double L, int W) {  
    Length = L;  
    Width = W;  
}  
public Rectangle( int L, double W){  
    Length = L;  
    Width = W;  
}  
// set Rectangle length method  
public void set_length(double L) {  
    Length = L;  
}  
// set Rectangle length method  
public void set_width(double W){  
    Width = W;  
}  
// set Rectangle length method  
public double get_length() {  
    return Length ;  
}  
public double get_width() {  
    return Width;  
}  
public double area()  
{  
    return Length * Width; }  
} // end of class Rectangle
```

in The form Add the following Tools



The screenshot shows a Java Swing window titled "Rectangle Class". The window has a yellow background and contains the following elements:

- Two text labels: "Enter Rectangle Length:" and "Enter Rectangle Width:", each followed by a text input field.
- A text label: "Area ="
- Two buttons: "Solve" and "Clear", positioned at the bottom of the window.

- In the Form1.Cs Add the following Code:

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

        private void Form1_Load(object sender, EventArgs e)
        {
            textBox1.Text = Convert.ToString(R1.get_length());
            textBox2.Text = Convert.ToString(R1.get_width());
            label4.Text = Convert.ToString(R1.area());
        }

        Rectangle R1 = new Rectangle();

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
        }

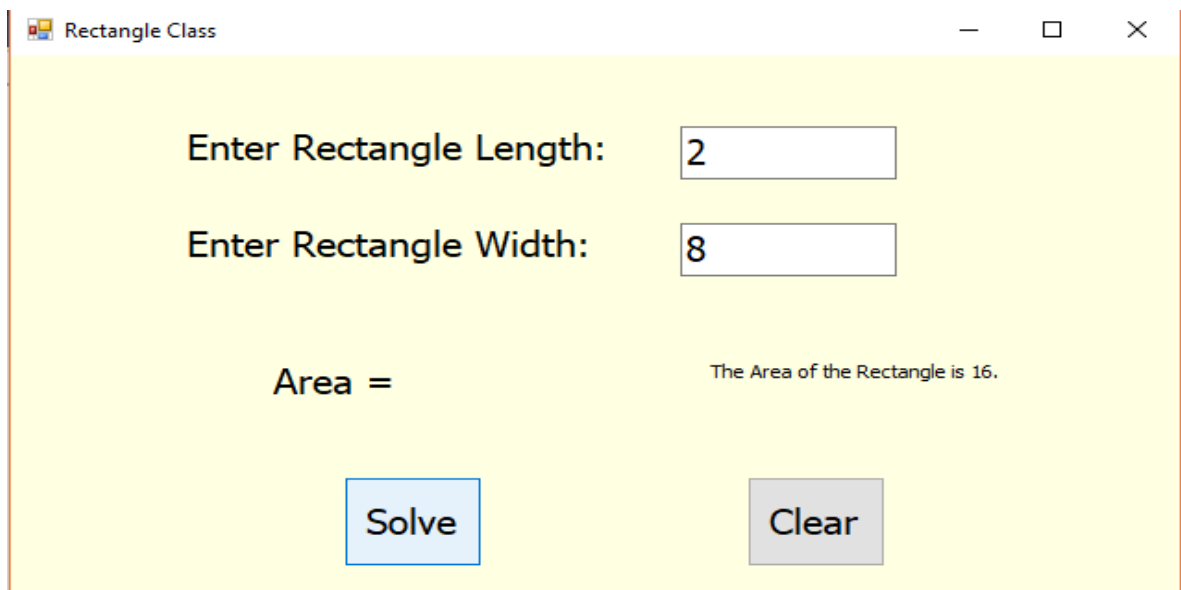
        private void textBox2_TextChanged(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int length = 0, breadth = 0;
            if (!int.TryParse(textBox1.Text, out length))
            {
                MessageBox.Show("I need just a number in the
texttextbox.");
                textBox1.Text = "";
                textBox2.Text = "";
                textBox1.Focus();
            }
            else if (!int.TryParse(textBox2.Text, out breadth))
            {
                MessageBox.Show("I need just a number in the
texttextbox.");
                textBox1.Text = "";
                textBox2.Text = "";
                textBox1.Focus();
            }
        }
    }
}
```

```
R1.set_width(Convert.ToDouble(textBox2.Text));
                R1.set_length(Convert.ToDouble(textBox1.Text));
label4.Text = "The Area of the Rectangle is " + R1.area() + ".";
}
    }
}
```

Run the Project:

The output Will Be :



Rectangle Class

Enter Rectangle Length: 2

Enter Rectangle Width: 8

Area = The Area of the Rectangle is 16.

Solve Clear

Example (2)

2. Write a C# windows Form Application code to create a Class Named *Shape* that contains the following :

✓ **Fields :**

- Length , Width

✓ **Methods:**

- Default and parametrized constructor .
- Set and get Methods .
- Area and diameters .
- Type to determine is it a square or rectangle (private to show Encapsulation)

- Return Details method.

Solution:

```

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace Shape_Example
7. {
8.     class Shape
9.     {
10.        private int length;
11.        private int width;
12.
13.
14.        public Shape()
15.        {
16.            length = 0;
17.            width = 0;
18.
19.        }
20.        public Shape(int length, int width)
21.        {
22.            this.length = length ;
23.            this.width = width ;
24.
25.        }
26.
27.
28.        private string check()
    
```

```
29.     {
30.     if (length == width)
31.         return "Square";
32.     else
33.         return "Rectangle";
34.     }
35.
36.     public int area()
37.     {
38.         return length * width;
39.     }
40.
41.     public int diameter()
42.     {
43.         return 2 * (length + width);
44.     }
45.
46.     public int getLength()
47.     {
48.         return length;
49.     }
50.
51.     public void setlength(int l)
52.     {
53.         length =l;
54.     }
55.
56.     public int getWidth()
57.     {
58.         return width;
59.     }
60.
61.     public void setWidth(int w)
62.     {
63.         width = w;
64.     }
65.
66.
67.
68.     public string returnDetails ()
69.     {
70.         string s = check(); string r = "";
71.         if (s == "Rectangle")
72.         {
73.             r += "Rectangle Length = " + getLength() + "\n" + "Rectangle Width = " + getWidth() + "\n";
74.             r += "Rectangle Area = " + area() + "\n" + "Rectangle Diameter = " + diameter();
75.         }
76.         else
77.         {
78.             r += "Square Length = " + getLength() + "\n" + "Square Width = " + getWidth();
79.             r += "Square Area = " + area() + Environment.NewLine + "Square Diameter = " + diameter();
```

```
80.     }
81.     return r;
82.     }
83. }
84. }
```

```
1. private void button1_Click(object sender, EventArgs e)
2.     {
3.         if (textBox1.Text != "" & textBox2.Text != "")
4.             {
5.                 int l = int.Parse(textBox1.Text);
6.                 int w = int.Parse(textBox2.Text);
7.                 Shape s = new Shape(l, w);
8.                 textBox3.Text = s.return Details ();
9.             }
10.        else
11.            MessageBox.Show("Empty Text Boxes");
12.    }
```

Example (3)

3. Write a C# windows Form Application code to create a Class Named *Student* that contains the following :

✓ **Fields :**

- Student Name , Degrass of three subjects (deg1,deg2,deg3).

✓ **Methods:**

- Default and parametrized constructor(overloading) .
- Set and get Methods (Encapsulation) .
- Average method to calculate student average (private) .
- stState method to determine the result of sudent only if all degrees pass 50 it will be pass and calculate the average.

Solution :

```

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5.
6. namespace Student_Example
7. {
8.     class Student
9.     {
10.         private string stName;
11.         private int stdeg1;
12.         private int stdeg2;
13.         private int stdeg3;
14.
15.         public Student()
16.         {

```



```
17.         stName = "Dina Riadh";
18.         stdeg1 = 80;
19.         stdeg2 = 70;
20.         stdeg3 = 65;
21.     }
22.
23.     public Student(string stName, int stdeg1, int stdeg2, int stdeg3)
24.     {
25.         this.stName = stName;
26.         this.stdeg1 = stdeg1;
27.         this.stdeg2 = stdeg2;
28.         this.stdeg3 = stdeg3;
29.
30.     }
31.
32.     public string getStName()
33.     {
34.         return stName;
35.     }
36.
37.     public int getStDeg1()
38.     {
39.         return stdeg1;
40.     }
41.
42.     public int getStDeg2()
43.     {
44.         return stdeg2;
45.     }
46.
47.     public int getStDeg3()
48.     {
49.         return stdeg3;
50.     }
51.
52.     public void setStName(string n)
53.     {
54.         stName = n;
55.     }
56.
57.     public void setStDeg1(int d1)
58.     {
59.         stdeg1 = d1;
60.     }
61.
62.     public void setStDeg2(int d2)
63.     {
64.         stdeg3 = d2;
65.     }
```

```

66.
67.     public void setStDeg3(int d3)
68.     {
69.         stdeg3 = d3;
70.     }
71.
72.     private double average()
73.     {
74.         double av = (getStDeg1() + getStDeg2() + getStDeg3()) / 3;
75.         return av;
76.     }
77.
78.     public string stState()
79.     {
80.         double av = 0; string state = "";
81.         if (getStDeg1() >= 50 && getStDeg2() >= 50 && getStDeg3() >= 50)
82.         {
83.
84.             av = average();
85.             state = "PASS " + " Student Average = " + av.ToString();
86.         }
87.
88.         else
89.             state = "Fail";
90.         return state;
91.     }
92. }
93. }

```

```

1. private void button1_Click(object sender, EventArgs e)
2. {
3.     if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") || (textBox4.Text == ""))
4.         st1 = new Student();
5.     else
6.     {
7.         string n=textBox1 .Text ;
8.         int d1=int.Parse (textBox2 .Text );
9.         int d2=int.Parse (textBox3 .Text );
10.        int d3=int.Parse (textBox4 .Text );
11.
12.        st1=new Student (n,d1,d2,d3);
13.    }
14.
15.    textBox5.Text += "Student Name : " + st1.getStName() + Environment .NewLine;
16.    textBox5.Text += "Student Result : " + st1.stState();
17.
18.
19. }

```

Exercises

1. Create a class named **BankAccount** with fields that hold an *account number*, the *owner's name*, and the *account balance(decimal)*. Include a constructor that initializes each field to appropriate default values. *Deposit* And *Withdraw* Methods .
2. Create a class named **Person** that holds the following fields: two String objects for the *person's first* and *last name* and a *LocalDate* object for the person's birthdate. Create two **Person** objects.
3. Create a class named **Complex** to represent complex numbers. this form. e.g

$$2.3 + 4.6i$$

The members of this class are as follows:

- ✓ Private data members:
 - **real** and **imag** of type double - to represent the complex number (real + imag i)
- ✓ Public member functions:
 - Default constructor that will initialize real and imag to zero
 - **getReal**: Accessor method that returns the value of data member real
 - **getImag**: Accessor method that returns the value of data member imag
 - **setReal**: method that sets the value of real
 - **setImag**: method that sets the value of imag
- ✓ Declare two Complex objects: c1 and c2.
- ✓ Prompt the user and use the input method to input complex values for the two objects c1 and c2. 3
- ✓ Display c1 and c2 using the output method. • Use **setReal()** to change the value of real for object c2 to 22.2.
- ✓ Use **getReal()** to obtain the value of real for object c2; display the value on the terminal screen.
- ✓ Display the updated c2 value