Practical No.1

Aim: Create a simple window form application for calculator.

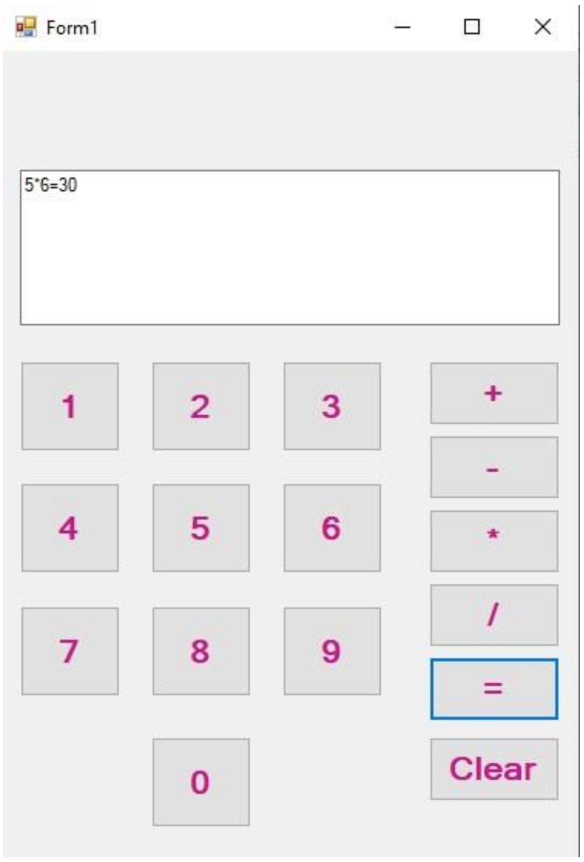
Code:

Form1.cs

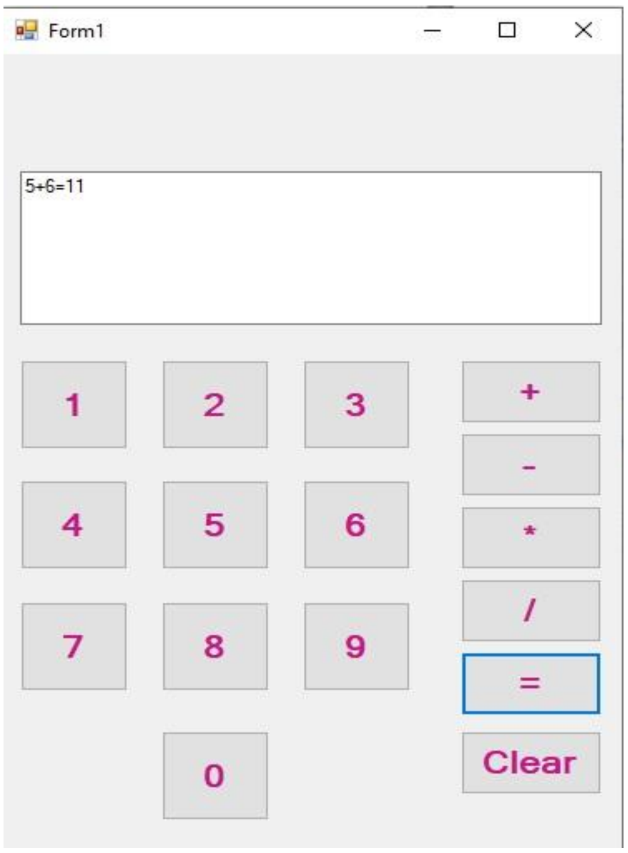
```
using           System;           using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
 System.Drawing; using
 System.Linq; using
 System.Text; using
 System.Windows.Forms;

namespace Calculator
{ public partial class Form1 : Form
    { char op;
        int t1, t2;
        public Form1()
        {
            InitializeComponent();
        }


        private void Form1_Load(object sender, EventArgs e) {

        }

        private void button9_Click(object sender, EventArgs e)
        { textBox1.Text += "9";
        }
        private void button1_Click(object sender, EventArgs e)
        { textBox1.Text += "1";
        }
        private void button2_Click(object sender, EventArgs e)
        { textBox1.Text += "2";
        }
        private void button3_Click(object sender, EventArgs e)
        { textBox1.Text += "3";
        }
        private void button4_Click(object sender, EventArgs e)
        { textBox1.Text += "4";
        }
        private void button5_Click(object sender, EventArgs e)
        { textBox1.Text += "5";
        }
        private void button6_Click(object sender, EventArgs e)
        { textBox1.Text += "6";
        }
        private void button7_Click(object sender, EventArgs e)
        { textBox1.Text += "7";
        }
        private void button8_Click(object sender, EventArgs e)
        { textBox1.Text += "8";
        }
        private void button10_Click(object sender, EventArgs e)
        { textBox1.Text += "0";
        }


        private void button16_Click(object sender, EventArgs e)
        { textBox1.Text = " ";
        }


        private void button11_Click(object sender, EventArgs e)
        { t1 = Convert.ToInt32(textBox1.Text);
           textBox1.Text = " "; op = '+';
        }


        private void button15_Click(object sender, EventArgs e)
        { t1 = Convert.ToInt32(textBox1.Text);
           textBox1.Text = " "; op = '-';
        }


        private void button14_Click(object sender, EventArgs e)
        { t1 = Convert.ToInt32(textBox1.Text);
           textBox1.Text = " "; op = '*';
        }


        private void button13_Click(object sender, EventArgs e)
        { t1 = Convert.ToInt32(textBox1.Text);
           textBox1.Text = " "; op = '/';
        }
        private void button12_Click(object sender, EventArgs e)
        {
           t2 = Convert.ToInt32(textBox1.Text);
          switch (op) {   case
'+':
                textBox1.Text = Convert.ToString(t1)+"+"+Convert.ToString(t2)+"="+Convert.ToString(t1 +
t2);
```

```
                    break;
                case '-':
                    textBox1.Text = Convert.ToString(t1) + "-" + Convert.ToString(t2) + "=" +
Convert.ToString(t1 - t2);
                    break;
                case '*':
                    textBox1.Text = Convert.ToString(t1) + "*" + Convert.ToString(t2) + "=" +
Convert.ToString(t1 * t2);
                break; case '/':
                    textBox1.Text = Convert.ToString(t1) + "/" + Convert.ToString(t2) + "=" +
Convert.ToString(t1 / t2);
                    break;
            }
        }
    }
}
```

Output: -



```
5*6=30
```



```
5+6=11
```

NAME: Patel Arun Ramjanak ROLL NO:21

Practical No.2

Aim: Design account class to handle different operations such as open, deposit, withdraw, check balance. Design GUI to demonstrate the use of this class.

Code:

Form1.cs

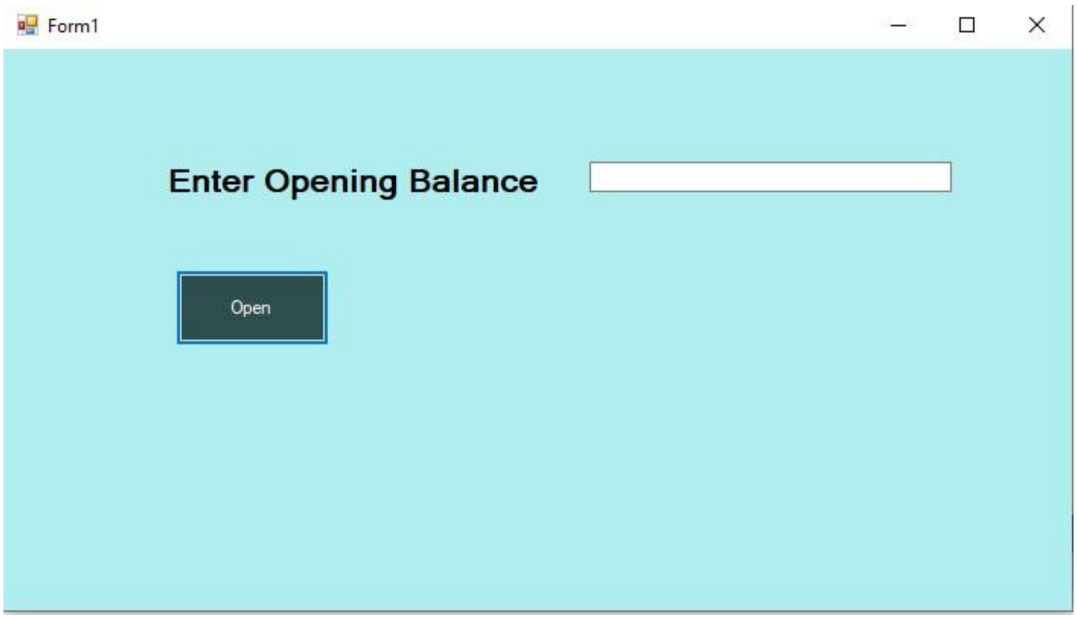
```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Bank
{ public partial class Form1 : Form
   { bank bankobj = new bank();
      public Form1()
      {
         InitializeComponent();
      }


      private void label1_Click(object sender, EventArgs e) {

      }

      private void button1_Click(object sender, EventArgs e)
      { deposit.Visible = true;
         checkbal.Visible = true;
         withdraw.Visible =
         true;
         int t1 = Convert.ToInt32(textBox1.Text);
         int b1=bankobj.open(t1);
         MessageBox.Show("Opening balance: "+b1);
         textBox1.Clear();

      }
      private void button4_Click(object sender, EventArgs e)
      { int t2 = Convert.ToInt32(textBox1.Text);
         int dep =bankobj.deposit(t2);
         MessageBox.Show("After Deposited: " + dep);
         textBox1.Clear();
      }
```

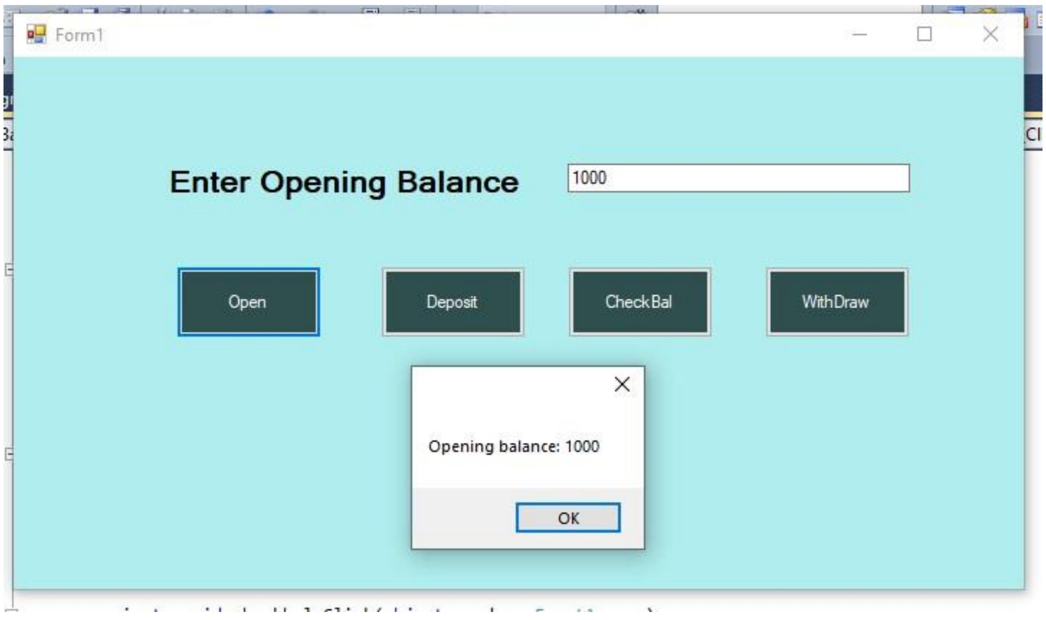
```
        private void Form1_Load(object sender, EventArgs e) {
            deposit.Visible = false;
            checkbal.Visible = false;
            withdraw.Visible = false;
        }


        private void checkbal_Click(object sender, EventArgs e) {


            int chk = bankobj.check();
            MessageBox.Show("Current balance: " + chk);
            textBox1.Clear();
        }
        private void withdraw_Click(object sender, EventArgs e)
        { int t3 = Convert.ToInt32(textBox1.Text);


            int wd = bankobj.withdraw(t3);
            MessageBox.Show("After withdraw: " + wd);
            textBox1.Clear();
        }
    }
}
```

Bank.cs

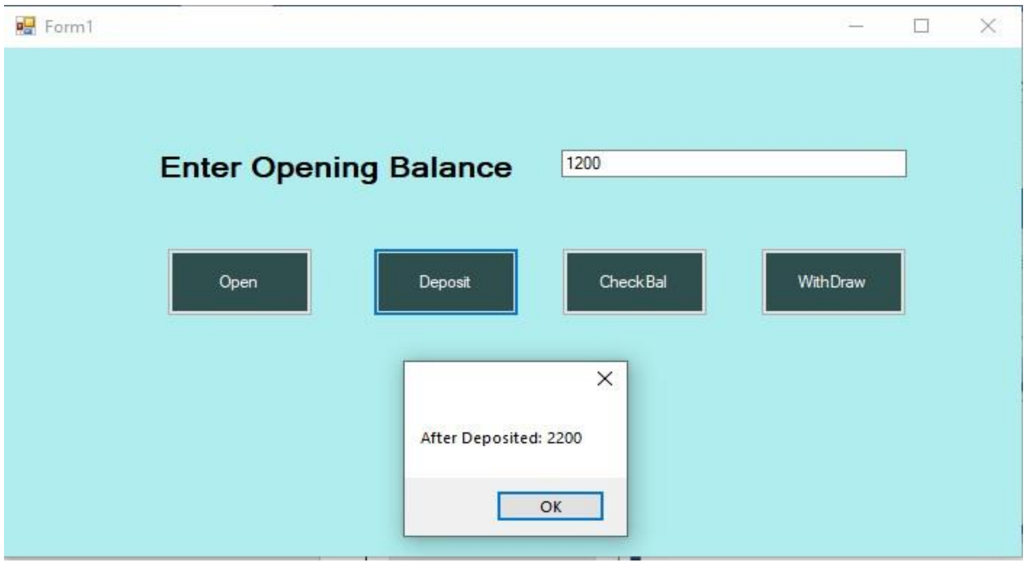
```
using System; using
System.Collections.Generic;
using System.Linq; using
System.Text;
namespace Bank
{ class bank
    { public int bal;
        public bank()
        { bal = 0;
        }


        public int open(int openbal)
        { bal +=openbal;
            return bal;
        }


        public int deposit(int depbal)
        { bal += depbal;
            return bal;
        }
        public int withdraw(int drawbal)
        { bal -= drawbal;
            return bal;
        }


        public int check() { int
            check = bal; return
            check;
        }


    }
}
```

Output:

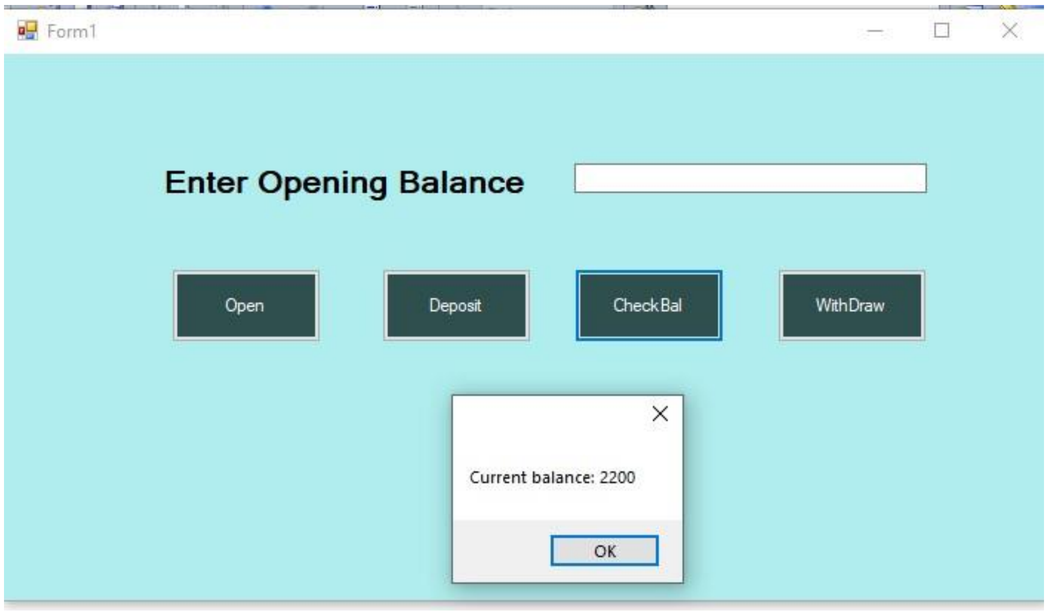




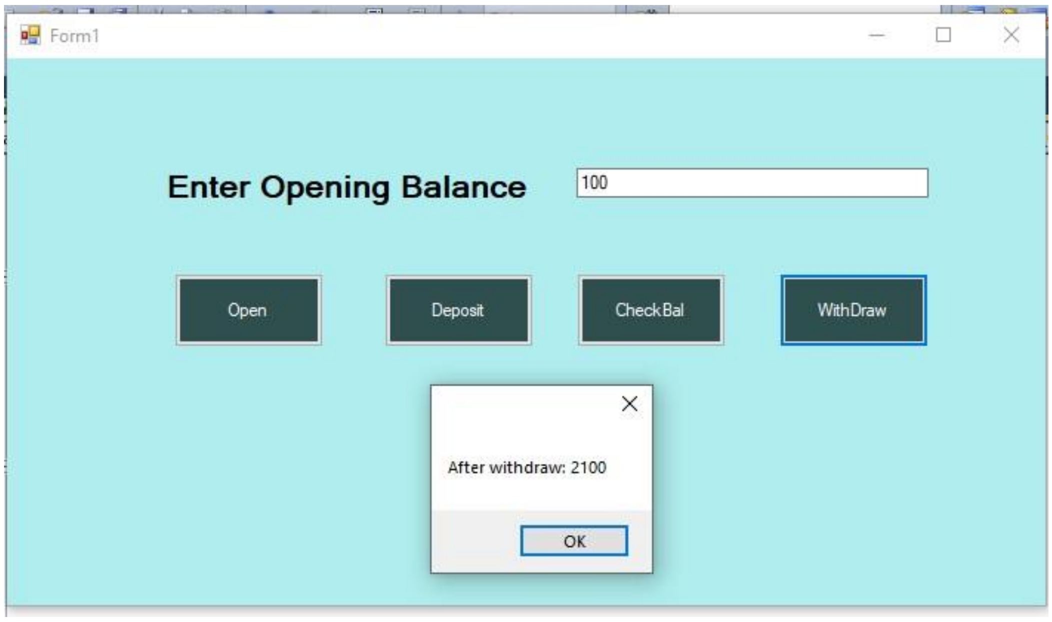
Opening balance: 1000



After Deposited: 2200



Current balance: 2200



After withdraw: 2100

NAME: Patel Arun Ramjanak **ROLL NO:21**

Practical No.3

Aim: Create windows form application to take array element form user & display that element on label. Also sort the array element & display the sorted list.

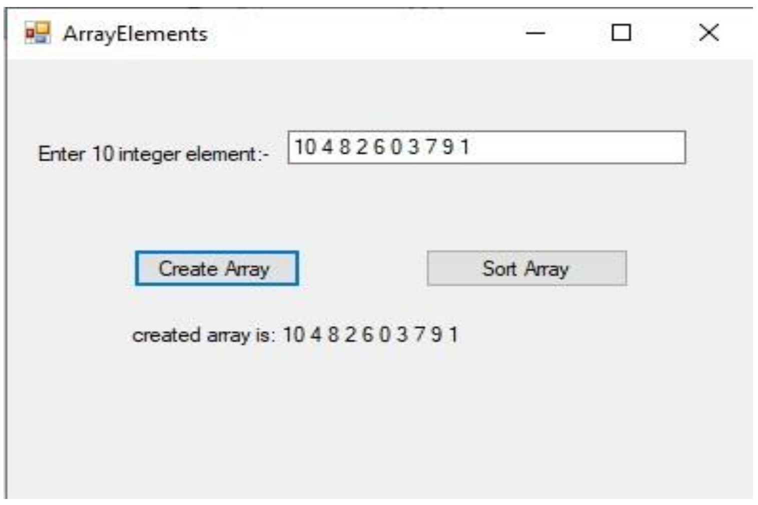
Code:

Form1.cs

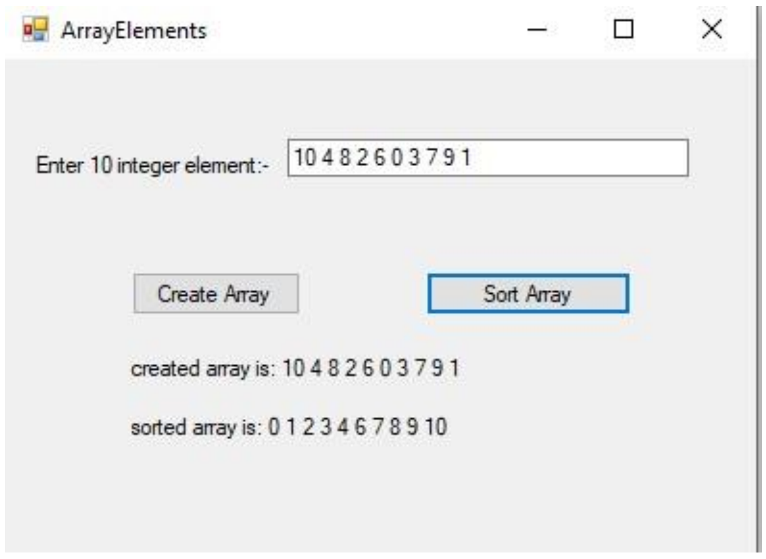
```
using System;

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

namespace Array
{ public partial class Form1 : Form
   { int[] num;
       public
       Form1()
       {
           InitializeComponent();
       }


       private void button1_Click(object sender, EventArgs e)
       { num = new int[10]; string
           s = textBox1.Text;
           string[] a = s.Split(' ');
           int c = 0;
           foreach (string i in a)
           { num[c] = int.Parse(i);
               c++;


           } label2.Text = "created array is:"; for
           (int j = 0; j < 10; j++) label2.Text += " "
           + num[j].ToString();
       }
       private void button2_Click(object sender, EventArgs e)
       { for(int i=0;i<num.Length;i++)
           for (int j = 0; j < i; j++)
               { if (num[j] > num[i])
                   { int temp = num[j];
                     num[j]=num[i];
                     num[i]=temp;
```

```
            }} label3.Text = "sorted
        array is:";
            for (int j = 0; j < 10; j++)
                label3.Text += " " + num[j].ToString();
        }



        private void label3_Click(object sender, EventArgs e) {


        }
    }
}
```

Output:



```
10 4 8 2 6 0 3 7 9 1
created array is: 10 4 8 2 6 0 3 7 9 1
```



```
10 4 8 2 6 0 3 7 9 1
created array is: 10 4 8 2 6 0 3 7 9 1
sorted array is: 0 1 2 3 4 6 7 8 9 10
```

Practical No. 4

Aim: Create windows form application to demonstrate array of object to display employee details like employee name and age.

Code:

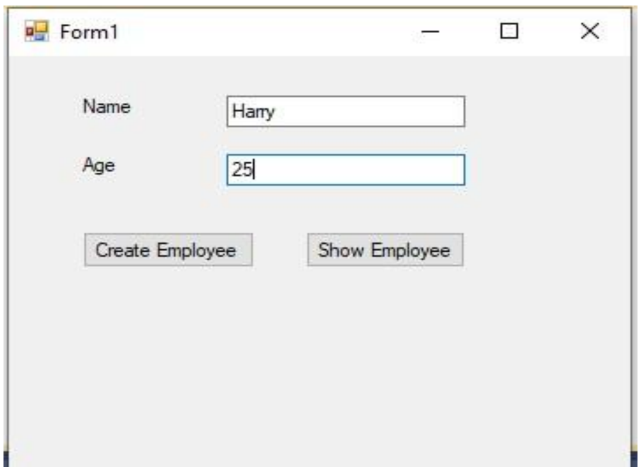
Form1.cs

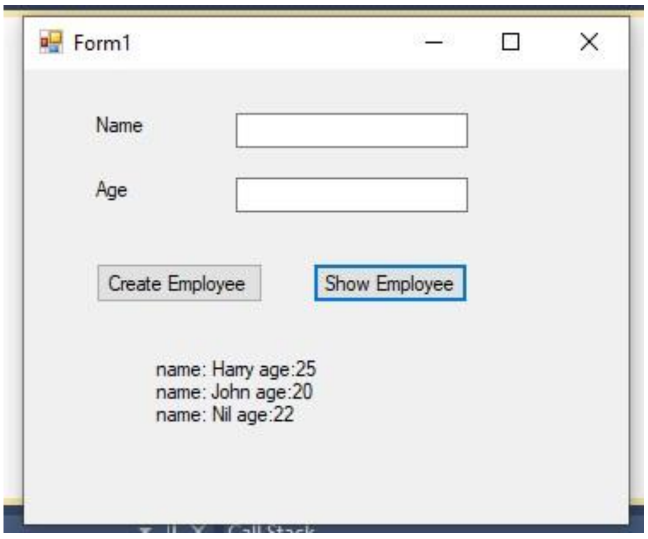
```
using System;
 usingSystem.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Employee
{ public partial class Form1 : Form
   { public employee[] em = new employee[3];
       string name; int age; int count = 0; public
       Form1()
       {
          InitializeComponent();
       }


       private void Form1_Load(object sender, EventArgs e) {

       }

       private void button1_Click(object sender, EventArgs e)
       {   name   =   textBox1.Text;   age   =
          Convert.ToInt32(textBox2.Text);
          em[count] = new employee(name, age);
          count++;              textBox1.Text="";
          textBox2.Text="";
       }


       private void button2_Click(object sender, EventArgs e)
       { for (int i = 0; i < em.Length; i++)
          { label3.Text += "name:" +" "+ em[i].name +" "+ "age:" + em[i].age + "\n";
          }
       }
   }
}
```

Employee.cs:

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

namespace Employee
{ public class employee
   {


       public string name; public int
       age; public employee(string n,
       int a)
   { name = n;
       age = a;
   }
   }
}
```

Output:





```
name: Harry age:25
name: John age:20
name: Nil age:22
```

Practical No. 5

Aim: Design a GUI to demonstrate single inheritance.

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.Windows.Forms;

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


      private void button1_Click(object sender, EventArgs e)
      { int t5 = Convert.ToInt32(textBox5.Text);
         int t6 = Convert.ToInt32(textBox6.Text);
         addInfo ai = new addInfo(textBox1.Text,textBox2.Text,textBox3.Text,textBox4.Text,t5,t6);
      label10.Text = ai.processInfo() + ai.processadd(); }
   }
}
```

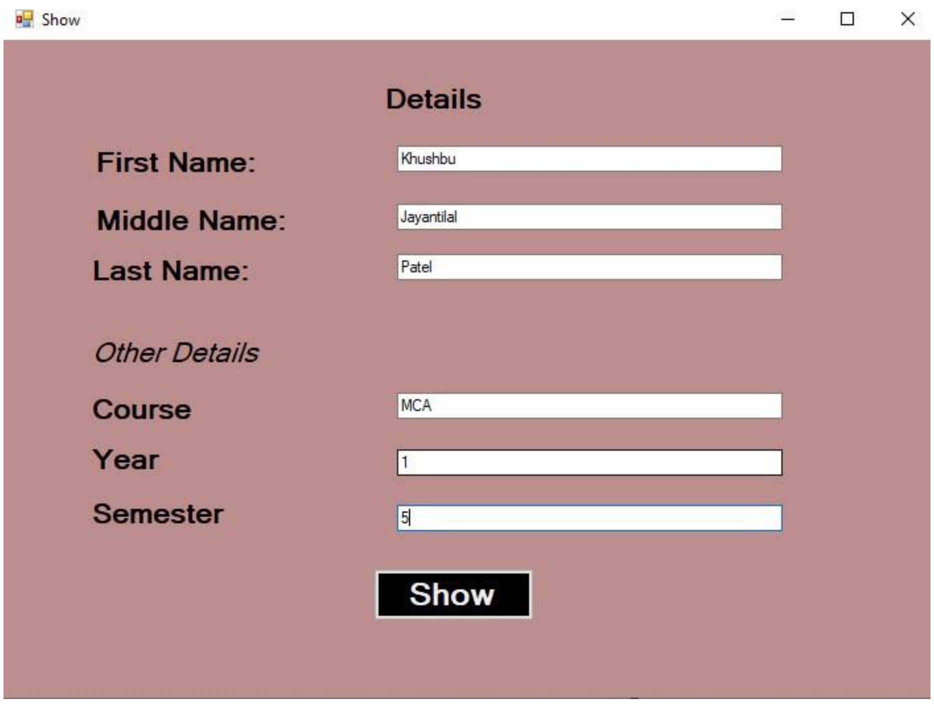
addInfo.cs:

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

namespace SingleInheritance
{

   class addInfo:basicInfo
   { public string course;
        public int sem, year;
        public addInfo() {
       }
       public addInfo(string afn, string amn, string aln, string cn, int s, int y) {
            fn = afn;
            mn = amn;
            ln   =   aln;
            course = cn;
            sem   =   s;
            year = y;
        }
        public string processadd() { return "\n" + "Course Name: " + course + "Semester:
        " + sem + "Year: " + year; }
    }
}
```

basicInfo.cs:

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

namespace SingleInheritance
{ class basicInfo
    { public string fn, ln, mn;
        public basicInfo()
        { }
        public basicInfo(string f, string m, string l) {
            fn = f; ln = l;
            mn = m;
        }
        public string processInfo()
        { return "Welcome " + fn + " "+mn+" "+ln;
        }
    }
}
```

Output:



Show

Details

First Name: Khushbu

Middle Name: Jayantilal

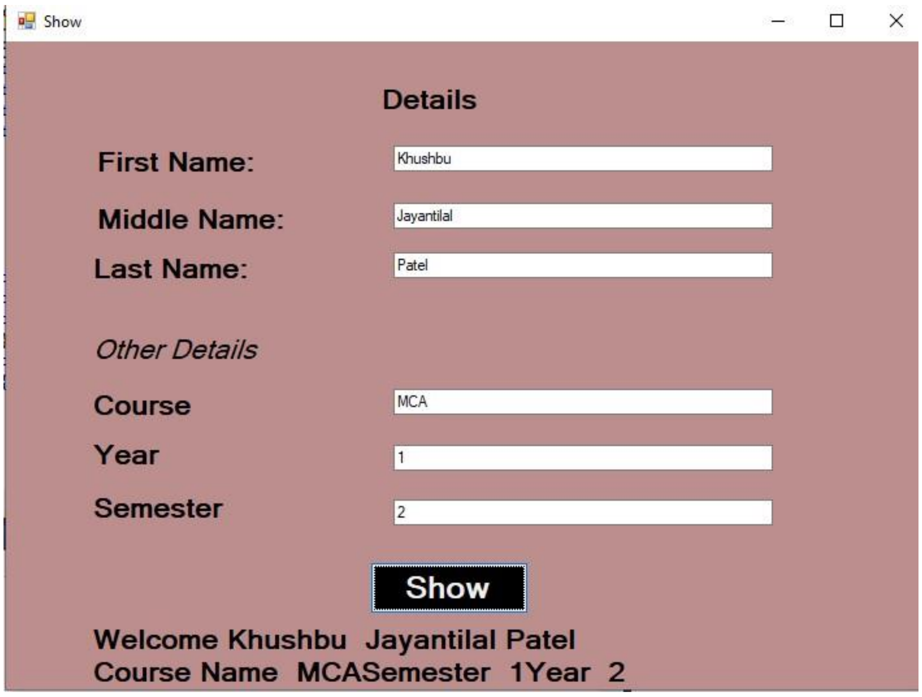
Last Name: Patel

Other Details

Course MCA

Year 1

Semester 5



Show

Details

First Name: Khushbu

Middle Name: Jayantilal

Last Name: Patel

Other Details

Course MCA

Year 1

Semester 2

```
Welcome Khushbu  Jayantilal Patel
Course Name  MCASemester  1Year  2
```

Practical No. 6

Aim: Design a GUI to demonstrate runtime polymorphism using abstract class.

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.Windows.Forms;

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


      private void button1_Click(object sender, EventArgs e)
      {
         Rectangle r= new Rectangle(); int t1 =
         Convert.ToInt32(textBox1.Text); int t2
         = Convert.ToInt32(textBox2.Text);
         r.set(t1, t2);
         int result = Convert.ToInt32(r.Area());
         MessageBox.Show("Area of Rectangle: "+result); }
   }
}
```

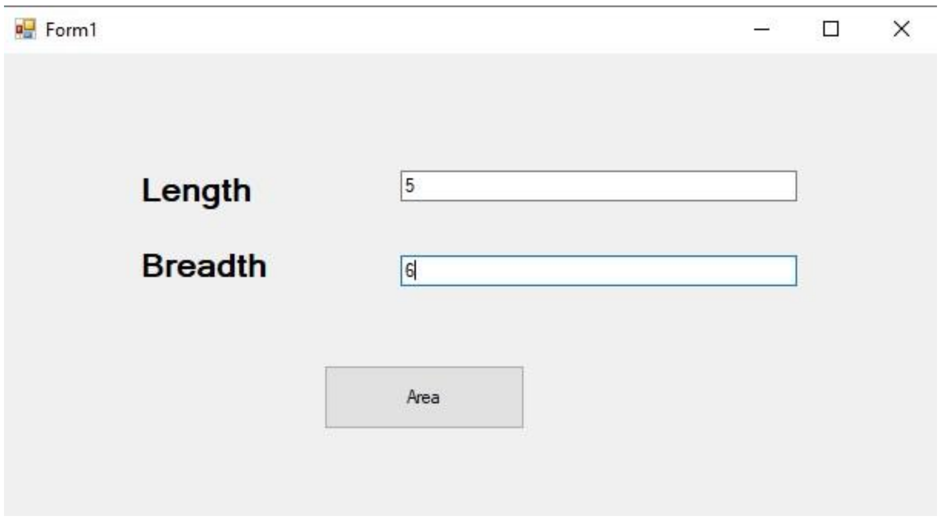
Rectangle.cs:

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

namespace Abstract
{ class Rectangle:Shape
   { public override int Area()
      { return length * breadth;
      }
   }
}
```

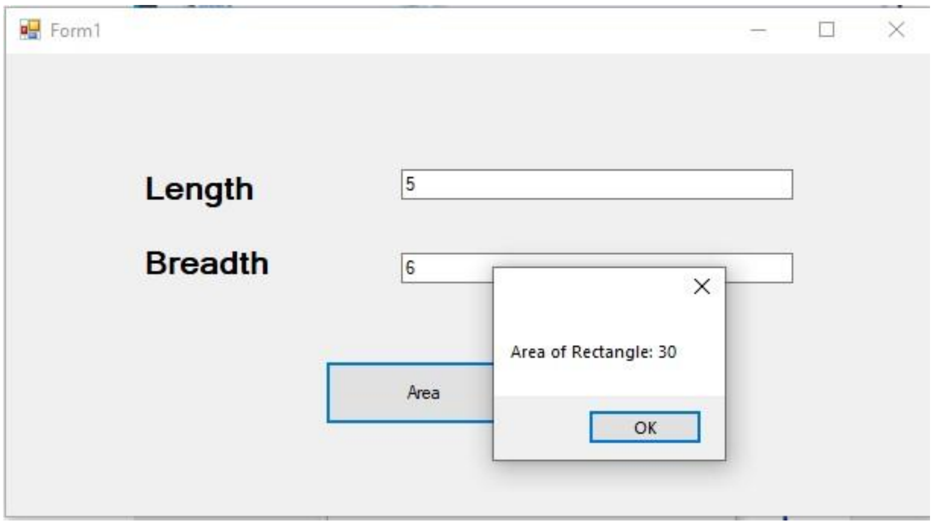
Shape.cs:

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

namespace Abstract
{ abstract class Shape
   { public int length, breadth; public
       void set(int a = 0, int b = 0) {
           length = a;
           breadth = b;
       }


       public abstract int Area();
   }
}
```

Output:



```
Length    5
Breadth   6
```



Area of Rectangle: 30

Practical No.7

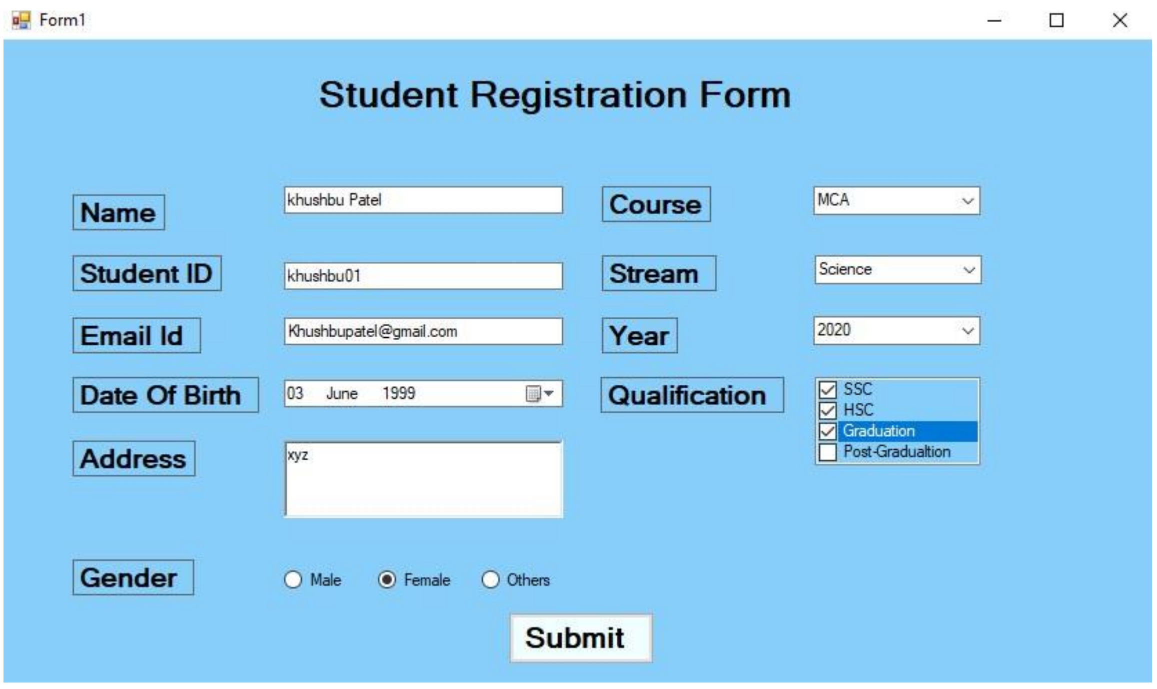
Aim: Design ASP.net Web Application Registration form of using web server controllers.

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.Windows.Forms;

namespace Registration
{ public partial class Form1 : Form
   { public Form1()
       {
          InitializeComponent();
       } private void label1_Click(object sender,
 EventArgs e) {

       }

       private void Form1_Load(object sender, EventArgs e) {

       }

       private void label2_Click(object sender, EventArgs e) {

       }
       private void button1_Click(object sender, EventArgs e)
       {
          MessageBox.Show("Successfully Registered!"); }
   }
}
```

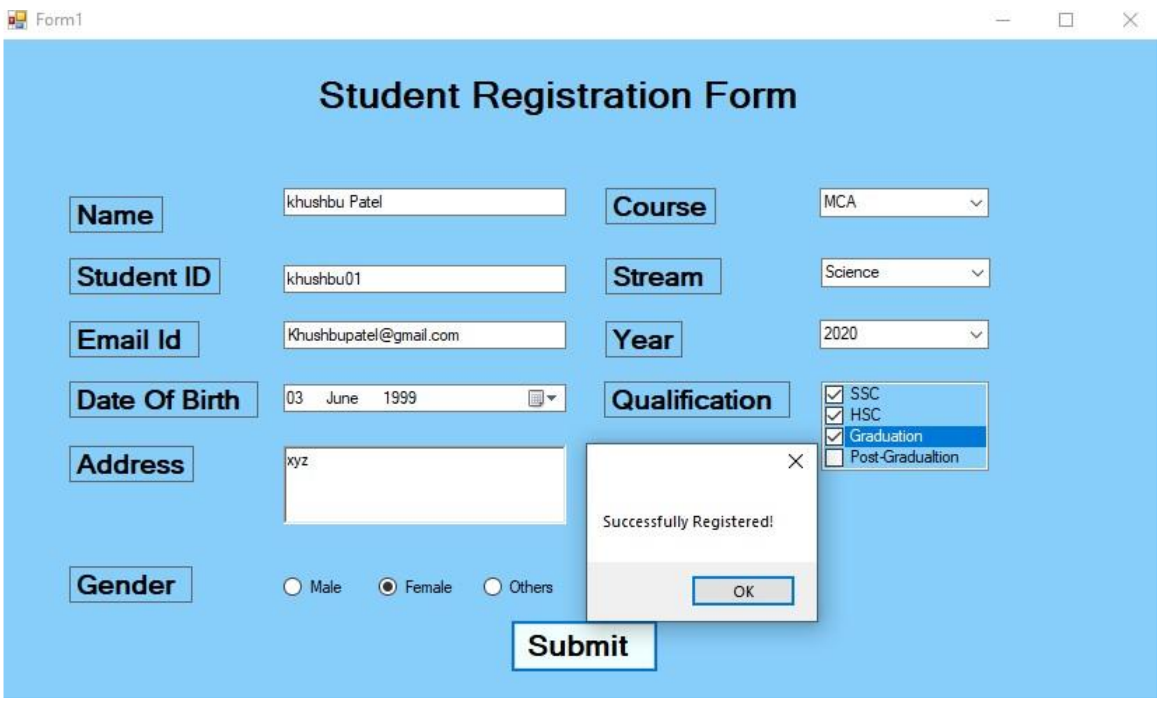
Output:



Form1

Student Registration Form

Name	khushbu Patel	Course	MCA
Student ID	khushbu01	Stream	Science
Email Id	Khushbupatel@gmail.com	Year	2020
Date Of Birth	03 June 1999	Qualification	✓ SSC ✓ HSC ✓ Graduation ☐ Post-Gradualtion
Address	xyz		
Gender	○ Male ● Female ○ Others		



Form1

Student Registration Form

Name	khushbu Patel	Course	MCA
Student ID	khushbu01	Stream	Science
Email Id	Khushbupatel@gmail.com	Year	2020
Date Of Birth	03 June 1999	Qualification	✓ SSC ✓ HSC ✓ Graduation ☐ Post-Gradualtion
Address	xyz		
Gender	○ Male ● Female ○ Others		

Successfully Registered!

Practical No.8

Aim: Create ASP.NET program based on validation control.

Code:

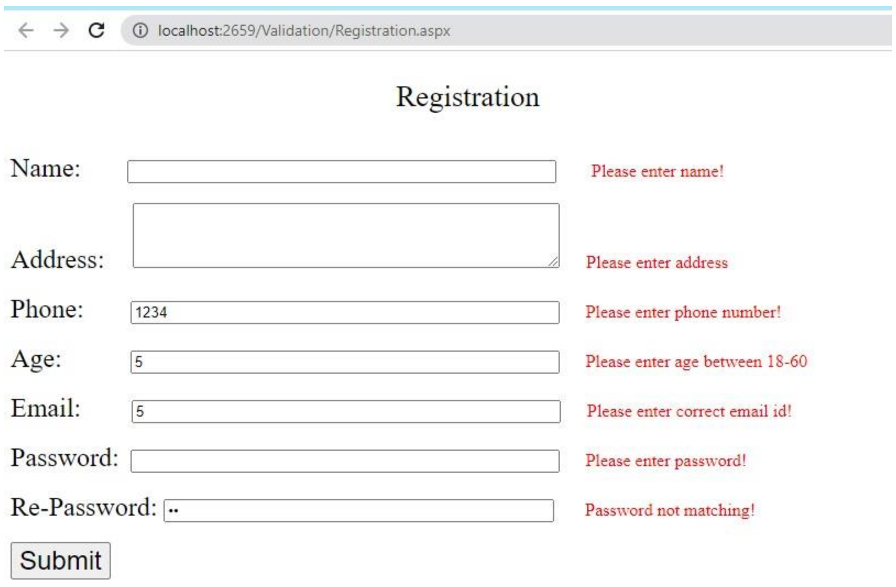
Register.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Registration.aspx.cs"
Inherits="Registration" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
        <head runat="server">
                <title>Untitled Page</title>
                </head>
<body>
<form id="form1" runat="server">
<div style="height: 529px; width: 831px">
  <br />
        <center>
        <asp:Label ID="Label1" runat="server" Text="Registration" Font-Size="20pt"></asp:Label>
        </center>
  <br />
  <br />
        <asp:Label ID="Label2" runat="server" Font-Size="18pt" Text="Name: "></asp:Label>
        <asp:TextBox ID="TextBox1" runat="server" ontextchanged="TextBox1_TextChanged"
        Width="382px">
        </asp:TextBox>

        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ControlToValidate="TextBox1" ErrorMessage="Please enter name!">
        </asp:RequiredFieldValidator>
   <br />
   <br />
        <asp:Label ID="Label3" runat="server" Font-Size="18pt" Text="Address: "></asp:Label>
        <asp:TextBox ID="TextBox2" runat="server" ontextchanged="TextBox2_TextChanged"
        Width="382px" Height="53px" Rows="3" TextMode="MultiLine"> </asp:TextBox>

        <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
        ControlToValidate="TextBox2" ErrorMessage="Please enter address">
        </asp:RequiredFieldValidator>
    <br />
     <br />
        <asp:Label ID="Label4" runat="server" Font-Size="18pt" Text="Phone: "></asp:Label>
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
        <asp:TextBox ID="TextBox3" runat="server" ontextchanged="TextBox3_TextChanged"
```

```
            Width="382px" MaxLength="10"></asp:TextBox>

            <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
            ControlToValidate="TextBox3" ErrorMessage="Please enter phone number!">
            </asp:RequiredFieldValidator>
       <br />
         <br />
            <asp:Label ID="Label5" runat="server" Font-Size="18pt" Text="Age: "></asp:Label>
             <asp:TextBox ID="TextBox4" runat="server" ontextchanged="TextBox4_TextChanged"
            Width="382px"></asp:TextBox>

            <asp:RangeValidator ID="RangeValidator2" runat="server"
            ControlToValidate="TextBox4" ErrorMessage="Please enter age between 18-60"
            MaximumValue="60" MinimumValue="18" Type="Integer"></asp:RangeValidator>
         <br />
         <br />
           <asp:Label ID="Label6" runat="server" Font-Size="18pt" Text="Email: "></asp:Label>
           <asp:TextBox ID="TextBox5" runat="server" ontextchanged="TextBox5_TextChanged"
            Width="382px"></asp:TextBox>

            <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
            ControlToValidate="TextBox5" ErrorMessage="Please enter correct email id!"
            ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">
            </asp:RegularExpressionValidator>
       <br />
       <br />
            <asp:Label ID="Label7" runat="server" Font-Size="18pt" Text="Password: "></asp:Label>
            &nbsp;<asp:TextBox ID="TextBox6" runat="server" ontextchanged="TextBox6_TextChanged"
            Width="382px" TextMode="Password"></asp:TextBox>

            <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
            ControlToValidate="TextBox1" ErrorMessage="Please enter password!">
            </asp:RequiredFieldValidator>
     <br />
     <br />
            <asp:Label ID="Label8" runat="server" Font-Size="18pt" Text="Re-Password: "></asp:Label>
            <asp:TextBox ID="TextBox7" runat="server" ontextchanged="TextBox1_TextChanged"
            Width="347px" TextMode="Password"></asp:TextBox>

            <asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="TextBox6"
            ControlToValidate="TextBox7" ErrorMessage="Password not matching!">
            </asp:CompareValidator>
     <br />
     <br />
            <asp:Button ID="Button1" runat="server" Text="Submit" Font-Size="18pt" />
      <br />
  <br />
  </div>
```

```
</form>
</body>
</html>
```

Output:

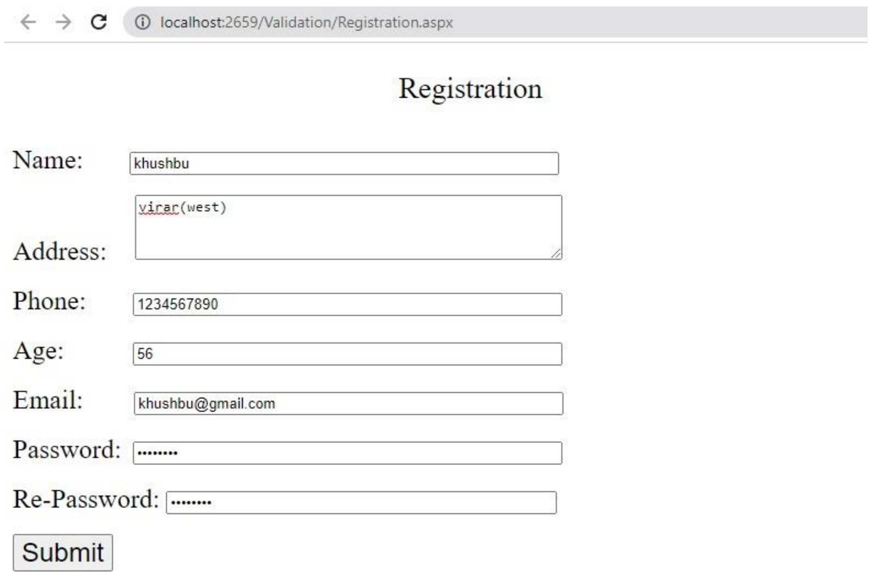


localhost:2659/Validation/Registration.aspx

Registration

Name:		Please enter name!
Address:		Please enter address
Phone:	1234	Please enter phone number!
Age:	5	Please enter age between 18-60
Email:	5	Please enter correct email id!
Password:		Please enter password!
Re-Password:	••	Password not matching!

Submit



localhost:2659/Validation/Registration.aspx

Registration

Name:	khushbu
Address:	virar(west)
Phone:	1234567890
Age:	56
Email:	khushbu@gmail.com
Password:	••••••••
Re-Password:	••••••••

Submit

Practical No.9

Aim: Create ASP.NET program using master page & themes and skins.

Code:

SkinFile:

```
<%--
Default skin template. The following skins are provided as examples only.

1.Named control skin. The SkinId should be uniquely defined because
  duplicate SkinId's per control type are not allowed in the same theme.

<asp:GridView runat="server" SkinId="gridviewSkin" BackColor="White" >
  <AlternatingRowStyle BackColor="Blue" />
</asp:GridView>

2.Default skin. The SkinId is not defined. Only one default control skin
  per control type is allowed in the same theme.

<asp:Image runat="server" ImageUrl="~/images/image1.jpg" />
--%>
<asp:Button runat="server" BackColor="#08260f" ForeColor="White" font-size="18px" font-name="Times
New Roman"/>
<asp:Label runat="server" Backcolor="White" ForeColor="Black" font-size="20px" font-name="Times
New Roman"/>
```

Style.css:

```
body
{ background-color:#d5f5dc; font-
   family:Times New Roman;
   font-size:14px;
}
.footer
{ background-color:Lime;
   height:50px;
   position:relative;
   top:450px;
   margin:0px 0px 10px;
   padding:5px;
   color:Black;
}
```

MasterPage.master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <link href="App_Themes/Theme1/Style.css" rel="stylesheet" type="text/css" />
   <asp:ContentPlaceHolder id="head" runat="server">
   </asp:ContentPlaceHolder>
</head>
<body>
   <form id="form1" runat="server">
   <div class="wrapper">
      <div class="menu">
      <ul>
         <li><a href="home.aspx">Home Page</a></li>
         <li><a href="category.aspx">Category</a></li>
         <li><a href="about.aspx">About Us</a></li> </ul>

      </div>
   <div class="content">
      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

      </asp:ContentPlaceHolder>
   </div>
   <div class="footer">
   <center><h1>Copyright @viva.com</h1></center>
   </div>
   </div>
   </form>
</body>
</html>
```

MasterPage2.master:

```
<%@ Master Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="MasterPage2.master.cs" Inherits="MasterPage2" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server"> </asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
This is a nested master page!
</asp:Content>
```

Home.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="home.aspx.cs" Inherits="home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server"> </asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    Welcome to viva!
    <br />
<br />
    <asp:Label ID="Label1" runat="server" Text="Label">Name: </asp:Label>
    &nbsp;
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Button ID="Button1" runat="server" Text="Submit" /> <br
    />
</asp:Content>
```

About.aspx:

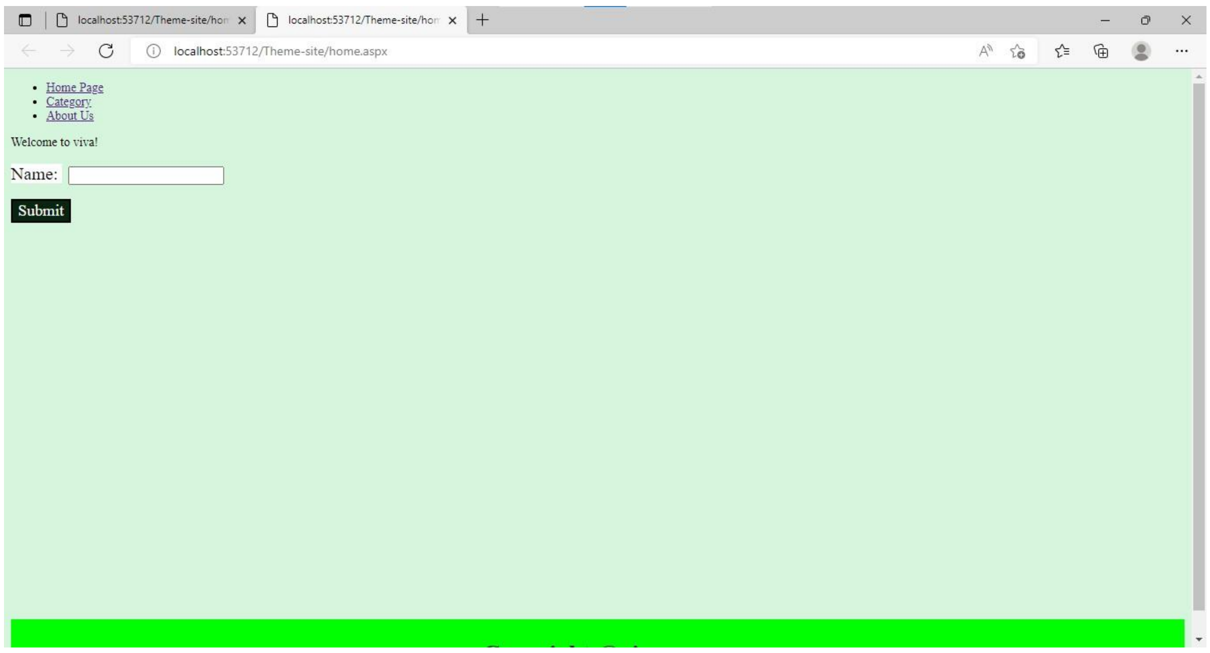
```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="about.aspx.cs" Inherits="about" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server"> </asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
<center><h3>About Viva!</h3></center>
<asp:Button ID="Button1" runat="server" Text="About" />

</asp:Content>
```

Category.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="category.aspx.cs" Inherits="category" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server"> </asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
<center><h3>Categories Viva!</h3></center>
<asp:Button ID="Button1" runat="server" Text="Categories" />
    <asp:DropDownList ID="DropDownList1" runat="server">
        <asp:ListItem>MBA</asp:ListItem>
        <asp:ListItem>MCA</asp:ListItem>
    </asp:DropDownList>
</asp:Content>
```

Output:

Home page:



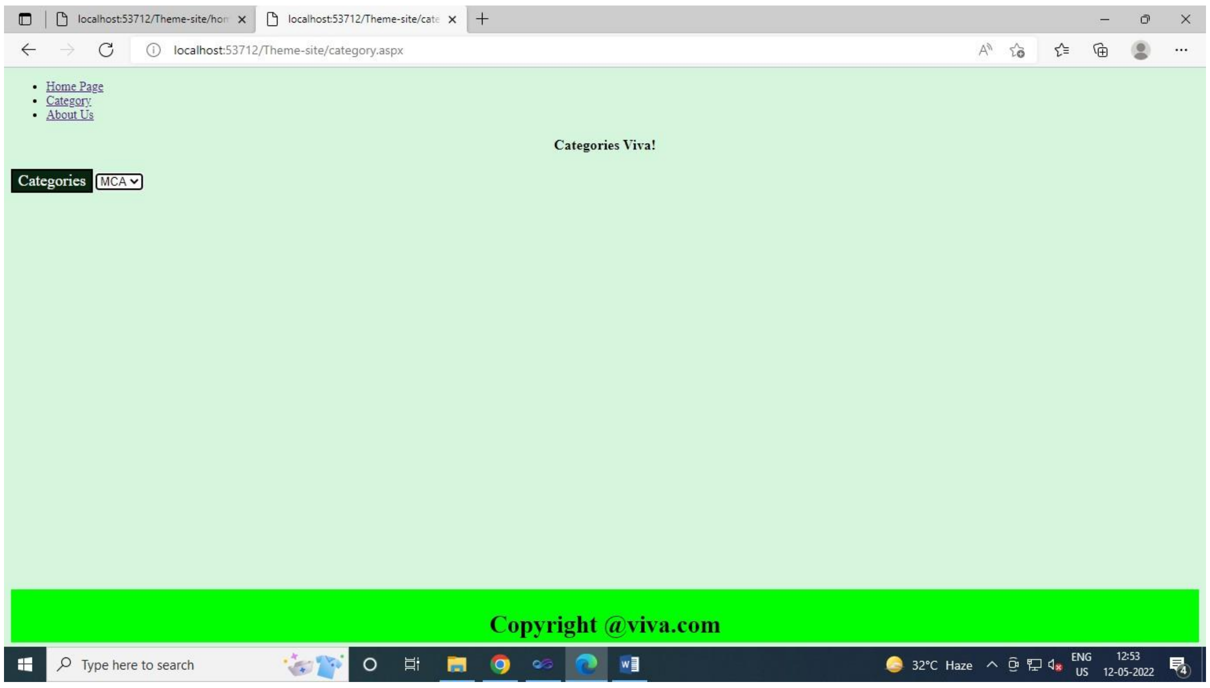
localhost:53712/Theme-site/home.aspx

- Home Page
- Category
- About Us

Welcome to viva!

Name:

Category Page:



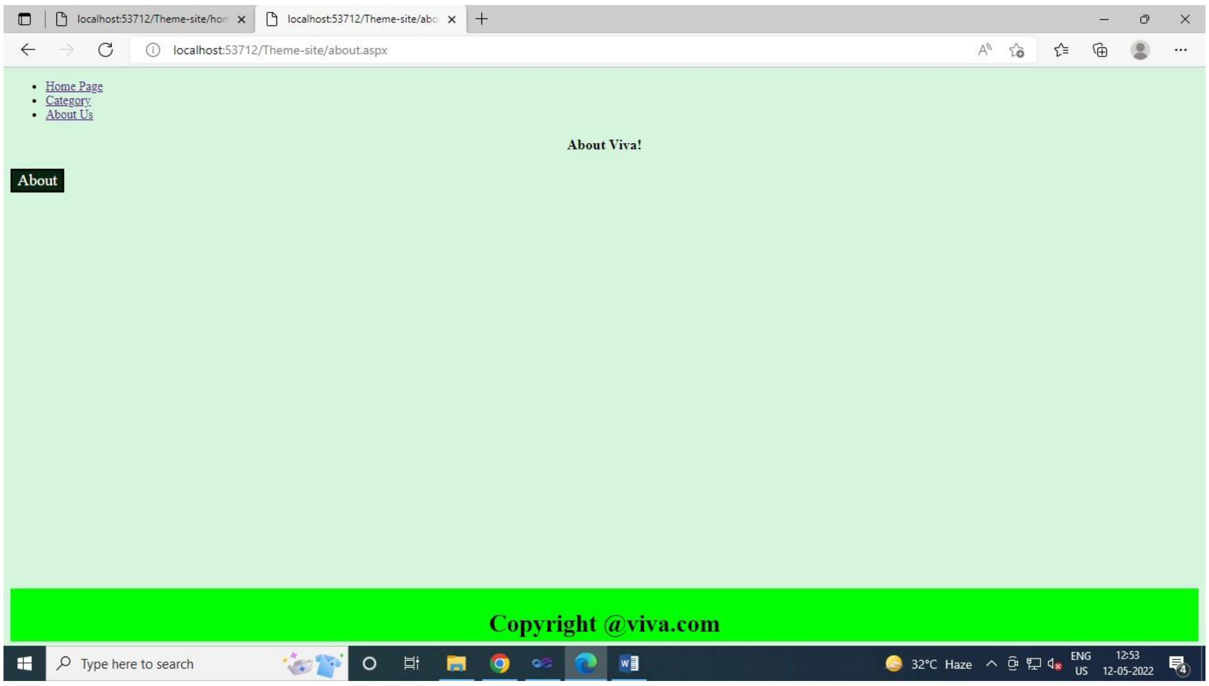
localhost:53712/Theme-site/category.aspx

- Home Page
- Category
- About Us

Categories Viva!

Copyright @viva.com

About Page:


localhost:53712/Theme-site/hom
localhost:53712/Theme-site/abo
localhost:53712/Theme-site/about.aspx
Type here to search
32°C Haze
ENG
US
12:53
12-05-2022

- Home Page
- Category
- About Us

About Viva!

About

Copyright @viva.com