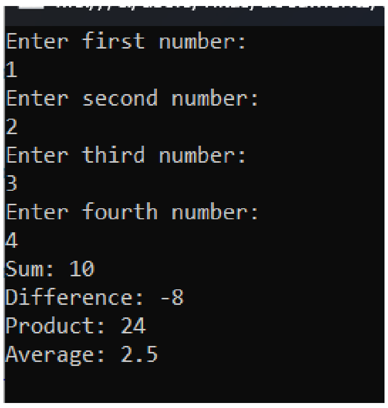
PRACTICAL 1

A] Aim: write a program to take 4 inputs from user and perform arithemetic operations.

CODE:

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

namespace PRACTICAL1
{
    class Program
    {
        static void Main(string[] args)
        {
            double num1, num2, num3, num4;

            Console.WriteLine("Enter first number:");
            num1 = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Enter second number:");
            num2 = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Enter third number:");
            num3 = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Enter fourth number:");
            num4 = Convert.ToDouble(Console.ReadLine());

            double sum = num1 + num2 + num3 + num4;
            double difference = num1 - num2 - num3 - num4;
            double product = num1 * num2 * num3 * num4;
            double average = sum / 4;

            Console.WriteLine("Sum: " + sum);
            Console.WriteLine("Difference: " + difference);
            Console.WriteLine("Product: " + product);
            Console.WriteLine("Average: " + average);

            Console.ReadKey();
        }
    }
}
```

OUTPUT:

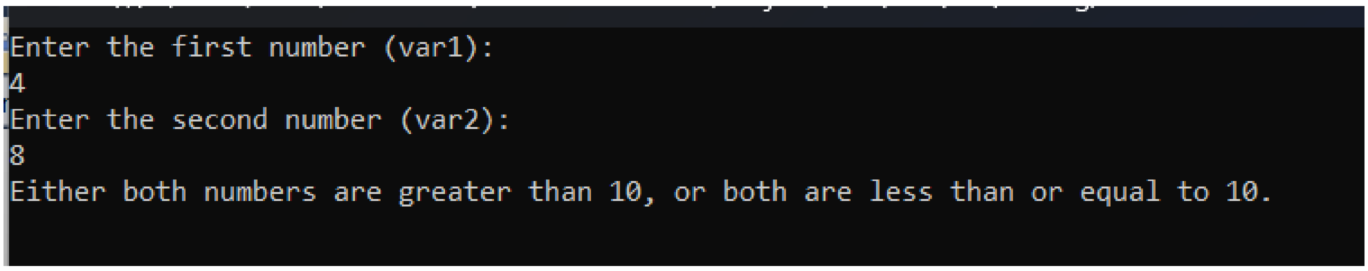


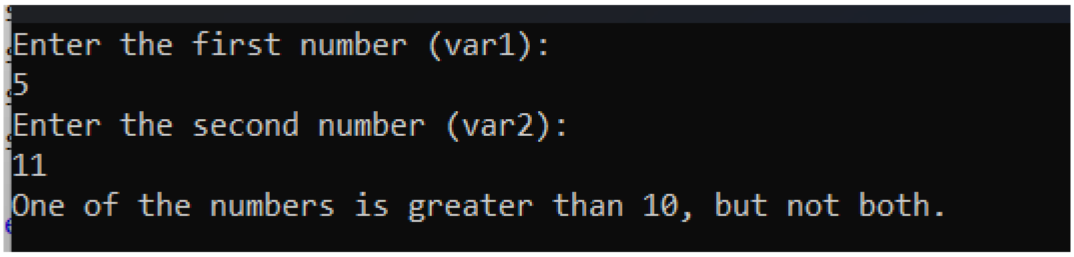
```
Enter first number:
1
Enter second number:
2
Enter third number:
3
Enter fourth number:
4
Sum: 10
Difference: -8
Product: 24
Average: 2.5
```

B] Aim: if you have two integers stored in variables var1 and var2, what Boolean tests can you perform to see if one or the other (but not both) is greater than 10.

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

namespace P1B
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the first number (var1):");
            int var1 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter the second number (var2):");
            int var2 = Convert.ToInt32(Console.ReadLine());

            bool result = (var1 > 10) ^ (var2 > 10);

            if (result)
            {
                Console.WriteLine("One of the numbers is greater than 10, but not 
both.");
            }
            else
            {
                Console.WriteLine("Either both numbers are greater than 10, or both 
are less than or equal to 10.");
            }
            Console.ReadKey();
        }
    }
}
```

OUTPUT:


```
Enter the first number (var1):
4
Enter the second number (var2):
8
Either both numbers are greater than 10, or both are less than or equal to 10.
```

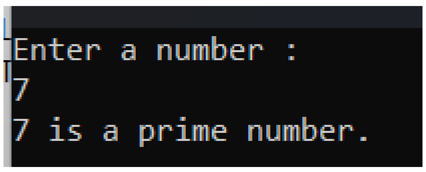

```
Enter the first number (var1):
5
Enter the second number (var2):
11
One of the numbers is greater than 10, but not both.
```

C] Aim: check whether number is prime number or not.

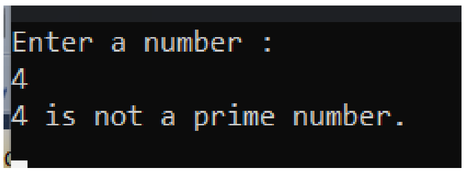
CODE:

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

namespace P1C
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter a number :");
            int number = Convert.ToInt32(Console.ReadLine());
            bool isPrime = true;

            if (number <= 1)
            {
                isPrime = false;
            }
            else
            {
                for (int i = 2; i <= number / 2; i++)
                {
                    if (number % i == 0)
                    {
                        isPrime = false;
                        break;
                    }
                }
            }

            if (isPrime)
            {
                Console.WriteLine(number + " is a prime number.");
            }
            else
            {
                Console.WriteLine(number + " is not a prime number.");
            }
            Console.ReadKey();
        }
    }
}
```

OUTPUT:



```
Enter a number :
7
7 is a prime number.
```



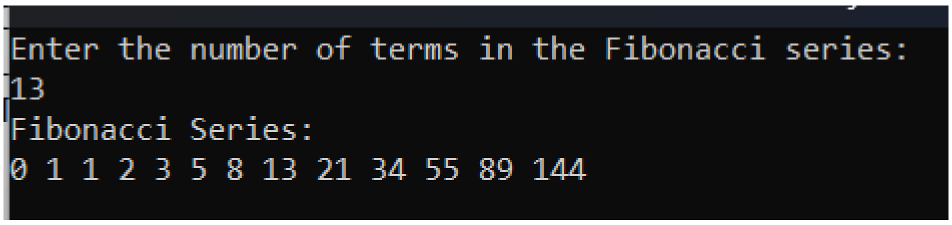
```
Enter a number :
4
4 is not a prime number.
```

D] Aim: write a program to print Fibonacci series

CODE:

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

namespace P1D
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the number of terms in the Fibonacci series:");
            int terms = Convert.ToInt32(Console.ReadLine());

            int o = 0, vo = 1, c = 0;
            Console.WriteLine("Fibonacci Series:");

            for (int i = 1; i <= terms; i++)
            {
                Console.Write(c + " ");
                c = o + vo;
                vo = o;
                o = c;
            }
            Console.ReadKey();
        }
    }
}
```

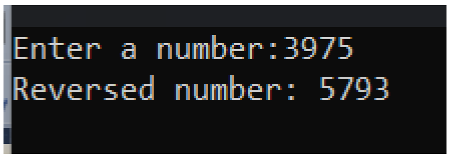
OUTPUT:



```
Enter the number of terms in the Fibonacci series:
13
Fibonacci Series:
0 1 1 2 3 5 8 13 21 34 55 89 144
```

E] Aim: write a program to print Reverse number.

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

namespace P1E
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter a number:");
            int number = Convert.ToInt32(Console.ReadLine());
            int reverse = 0;

            while (number != 0)
            {
                int digit = number % 10;
                reverse = reverse * 10 + digit;
                number /= 10;
            }

            Console.WriteLine("Reversed number: " + reverse);
            Console.ReadKey();
        }
    }
}
```

OUTPUT:

```
Enter a number:3975
Reversed number: 5793
```

Conclusion: We have successfully completed the experiment and achieved the desired results.

PRACTICAL-2

A] Aim: create a calculator using ASP.NET Web Application

Create a new Web Application, remove About.aspx and Default.aspx file from project files in Solution Explorer, Right click on Project -> add -> new item -> select web Form And rename it to Default.aspx

Include the following code in body of Default.aspx

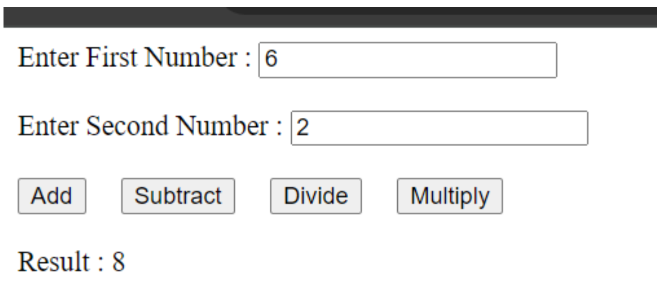
Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="
calculator.WebForm1" %>

<!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>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        Enter First Number :
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <br />
        <br />
        Enter Second Number :
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
        <br />
        <br />
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Add" />
&nbsp;&nbsp;&nbsp;
        <asp:Button ID="Button2" runat="server" onclick="Button2_Click"
            Text="Subtract" />
&nbsp;&nbsp;&nbsp;
        <asp:Button ID="Button3" runat="server" onclick="Button3_Click" Text="Divide" />
&nbsp;&nbsp;&nbsp;
        <asp:Button ID="Button4" runat="server" onclick="Button4_Click"
            Text="Multiply" />
        <br />
        <br />
        <asp:Label ID="Label1" runat="server"></asp:Label>

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

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace calculator
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            int n1 = Convert.ToInt32(TextBox1.Text);
            int n2 = Convert.ToInt32(TextBox2.Text);
            Label1.Text = "Result : " + (n1 / n2);
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            int n1 = Convert.ToInt32(TextBox1.Text);
            int n2 = Convert.ToInt32(TextBox2.Text);
            Label1.Text = "Result : " + (n1 + n2);
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            int n1 = Convert.ToInt32(TextBox1.Text);
            int n2 = Convert.ToInt32(TextBox2.Text);
            Label1.Text = "Result : " + (n1 - n2);
        }

        protected void Button4_Click(object sender, EventArgs e)
        {
            int n1 = Convert.ToInt32(TextBox1.Text);
            int n2 = Convert.ToInt32(TextBox2.Text);
            Label1.Text = "Result : " + (n1 * n2);
        }
    }
}
```

OUTPUT:



Enter First Number : 6

Enter Second Number : 2

Result : 8

B] Aim: Table Control: take student details from user and display that record in table control.

Create a new Web Application, remove About.aspx and Default.aspx file from project files in Solution Explorer, Right click on Project -> add -> new item -> select web Form And rename it to Default.aspx

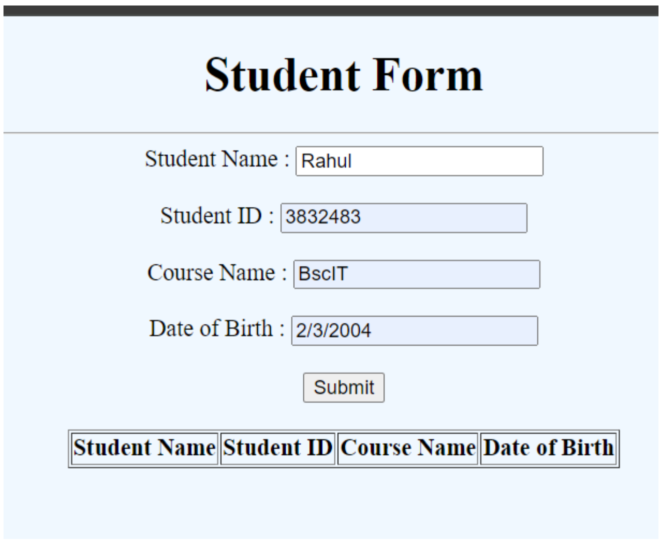
Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="
Table.WebForm1" %>

<!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>
</head>
<body style="background-color:aliceblue">
     <center>
     <h1>Student Form</h1><hr />
     <form id="form1" runat="server">
         <div>
             <label for="Label1">Student Name : </label>
             <asp:TextBox ID="TextBox1" runat="server" /><br /><br />
             <label for="Label3">Student ID : </label>
             <asp:TextBox ID="TextBox2" runat="server" /><br /><br />
             <asp:Label ID="Label3" runat="server" Text="Course Name : "></asp:Label>
             <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox><br /><br />
             <asp:Label ID="Label4" runat="server" Text="Date of Birth : "></asp:Label>
             <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox><br /><br />
             <asp:Button ID="Button1" runat="server" Text="Submit"
OnClick="Button1_Click"/><br/><br />
             <asp:Table ID="Tabel1" runat="server" Border="ActiveBorder">
                 <asp:TableHeaderRow>
                     <asp:TableHeaderCell>Student Name</asp:TableHeaderCell>
                     <asp:TableHeaderCell>Student ID</asp:TableHeaderCell>
                     <asp:TableHeaderCell>Course Name</asp:TableHeaderCell>
                     <asp:TableHeaderCell>Date of Birth</asp:TableHeaderCell>
                 </asp:TableHeaderRow>
             </asp:Table>
         </div>
     </form>
     </center>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Table
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string name = TextBox1.Text;
            string studentID = TextBox2.Text;
            string coursename = TextBox3.Text;
            string DOB = TextBox4.Text;

            TableRow newRow = new TableRow();
            TableCell nameCell = new TableCell();
            TableCell idCell = new TableCell();
            TableCell courseCell = new TableCell();
            TableCell dob = new TableCell();

            nameCell.Text = name;
            idCell.Text = studentID;
            courseCell.Text = coursename;
            dob.Text = DOB;

            newRow.Cells.Add(nameCell);
            newRow.Cells.Add(idCell);
            newRow.Cells.Add(courseCell);
            newRow.Cells.Add(dob);

            Tabel1.Rows.Add(newRow);

            TextBox1.Text = string.Empty;
            TextBox2.Text = string.Empty;
            TextBox3.Text = string.Empty;
            TextBox4.Text = string.Empty;
        }

    }
}
```

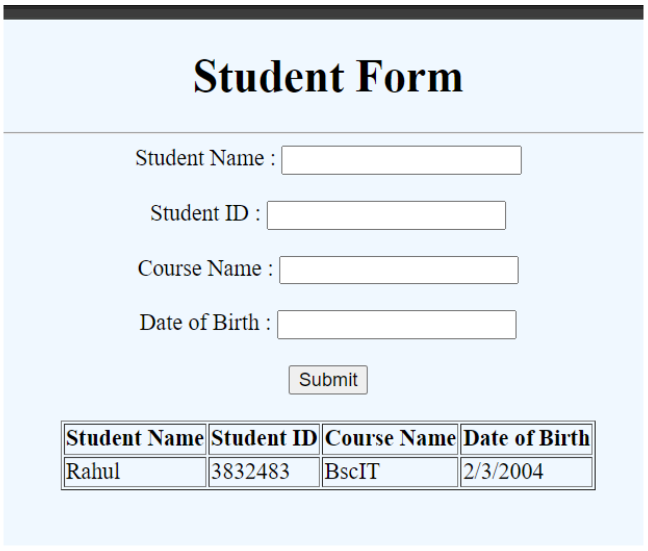
OUTPUT:



Student Form

Student Name :	Rahul
Student ID :	3832483
Course Name :	BscIT
Date of Birth :	2/3/2004

Student Name	Student ID	Course Name	Date of Birth



Student Form

Student Name :	
Student ID :	
Course Name :	
Date of Birth :	

Student Name	Student ID	Course Name	Date of Birth
Rahul	3832483	BscIT	2/3/2004

Conclusion: We have successfully completed the experiment and achieved the desired results.

PRACTICAL-3

Aim: Implement a calendar control to display different date and time formats.

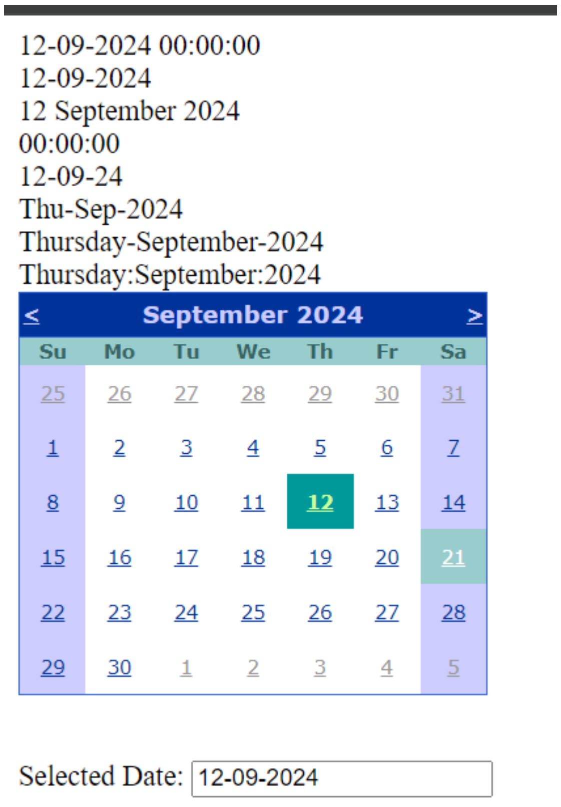
WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="P3.WebForm1" %>

<!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>
</head>
<body>
    <form id="form1" runat="server">
    <div style="height: 297px">

        <asp:Calendar ID="Calendar1" runat="server" BackColor="White"
            BorderColor="#3366CC" BorderWidth="1px" CellPadding="1"
            DayNameFormat="Shortest" Font-Names="Verdana" Font-Size="8pt"
            ForeColor="#003399" Height="226px"
            onselectionchanged="Calendar1_SelectionChanged" Width="263px">
            <DayHeaderStyle BackColor="#99CCCC" ForeColor="#336666" Height="1px" />
            <NextPrevStyle Font-Size="8pt" ForeColor="#CCCCFF" />
            <OtherMonthDayStyle ForeColor="#999999" />
            <SelectedDayStyle BackColor="#009999" Font-Bold="True" ForeColor="#CCFF99" />
            <SelectorStyle BackColor="#99CCCC" ForeColor="#336666" />
            <TitleStyle BackColor="#003399" BorderColor="#3366CC" BorderWidth="1px"
                Font-Bold="True" Font-Size="10pt" ForeColor="#CCCCFF" Height="25px" />
            <TodayDayStyle BackColor="#99CCCC" ForeColor="White" />
            <WeekendDayStyle BackColor="#CCCCFF" />
        </asp:Calendar>
        <br />
        <br />
        Selected Date:
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

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

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace P3
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Calendar1_SelectionChanged(object sender, EventArgs e)
        {
            TextBox1.Text = Calendar1.SelectedDate.ToString("dd-MM-yyyy");
            Response.Write(Calendar1.SelectedDate.ToString());
            Response.Write("<br>" + Calendar1.SelectedDate.ToShortDateString());
            Response.Write("<br>" + Calendar1.SelectedDate.ToLongDateString());
            Response.Write("<br>" + Calendar1.SelectedDate.ToLongTimeString());
            Response.Write("<br>" + Calendar1.SelectedDate.ToString("dd/MM/yy"));
            Response.Write("<br>" + Calendar1.SelectedDate.ToString("ddd/MMM/yyyy"));
            Response.Write("<br>" + Calendar1.SelectedDate.ToString("dddd/MMMM/yyyy"));
            Response.Write("<br>" + Calendar1.SelectedDate.ToString("dddd:MMMM:yyyy"));
        }
    }
}
```

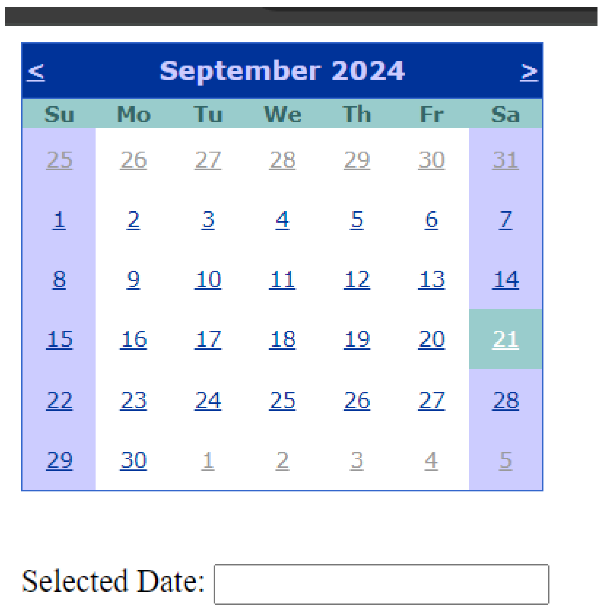
OUTPUT:



```
12-09-2024 00:00:00
12-09-2024
12 September 2024
00:00:00
12-09-24
Thu-Sep-2024
Thursday-September-2024
Thursday:September:2024
```

<			September 2024			>
Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Selected Date: 12-09-2024



<			September 2024			>
Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Selected Date:

Conclusion: We have successfully completed the experiment and achieved the desired results.

PRACTICAL-4

A] Aim: create a Registration form, apply validation and print submitted successful message.

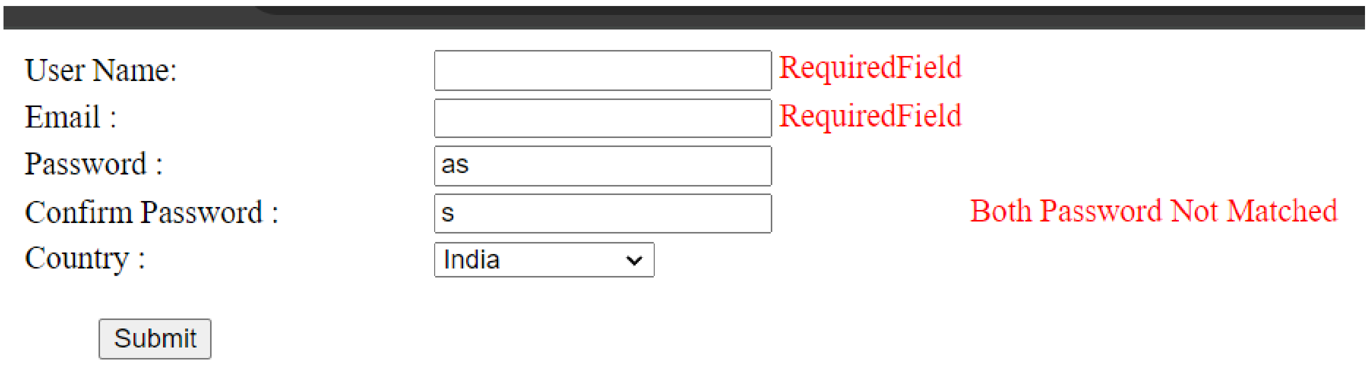
WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="P4.WebForm1" %>

<!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>
    <style type="text/css">
        .style1
        {
            width: 100%;
        }
        .style2
        {
            width: 202px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        <table class="style1">
            <tr>
                <td class="style2">
                    User Name:</td>
                <td>
                    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
                        ControlToValidate="TextBox1" ErrorMessage="RequiredField"
ForeColor="Red"></asp:RequiredFieldValidator>
                </td>
                <td>
                    &nbsp;</td>
            </tr>
            <tr>
                <td class="style2">
                    Email :</td>
                <td>
                    <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
                        ControlToValidate="TextBox2" ErrorMessage="RequiredField"
ForeColor="Red"></asp:RequiredFieldValidator>
                    <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server"
                        ControlToValidate="TextBox2" ErrorMessage="Enter valid Email"
ForeColor="Red"
                        ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*"></asp:RegularExpressionValidator>
                </td>
                <td>
                    &nbsp;</td>
            </tr>
            <tr>
                <td class="style2">
                    Password :</td>
                <td>
                    <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
                        ControlToValidate="TextBox3" ErrorMessage="RequiredField"
ForeColor="Red"></asp:RequiredFieldValidator>
                </td>
                <td>
                    &nbsp;</td>
            </tr>
            <tr>
                <td class="style2">
                    Confirm Password :</td>
                <td>
                    <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
                        ControlToValidate="TextBox4" ErrorMessage="RequiredField"
ForeColor="Red"></asp:RequiredFieldValidator>
                    <asp:CompareValidator ID="CompareValidator1" runat="server"
                        ControlToCompare="TextBox3" ControlToValidate="TextBox4"
                        ErrorMessage="Both Password Not Matched"
ForeColor="Red"></asp:CompareValidator>
                </td>
                <td>
                    &nbsp;</td>
            </tr>
            <tr>
                <td class="style2">
                    Country :</td>
                <td>
                    <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True">
                        <asp:ListItem>Select Country</asp:ListItem>
                        <asp:ListItem>India</asp:ListItem>
                        <asp:ListItem>US</asp:ListItem>
                        <asp:ListItem>Italy</asp:ListItem>
                    </asp:DropDownList>
                </td>
                <td>
                    &nbsp;</td>
            </tr>
        </table>

    </div>
    <div style="margin-left: 40px">
        <br />
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Submit" />
    </div>
    </form>
</body>
</html>
```

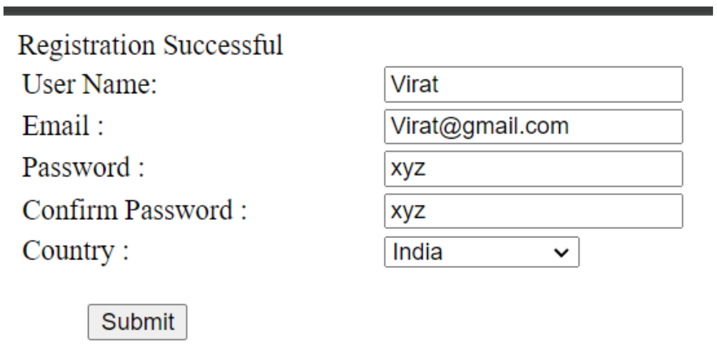
WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace P4
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Write("Registration Successful");
        }
    }
}
```

OUTPUT:



User Name:		RequiredField
Email :		RequiredField
Password :	as	
Confirm Password :	s	Both Password Not Matched
Country :	India	



Registration Successful

User Name:	Virat
Email :	Virat@gmail.com
Password :	xyz
Confirm Password :	xyz
Country :	India

B] Aim: AdRotator: create 3 advertisement banners, for each banner provide navigate link.

1) create new web site
2) select ASP.NET empty web site
3) right click on project name > add new item > web form

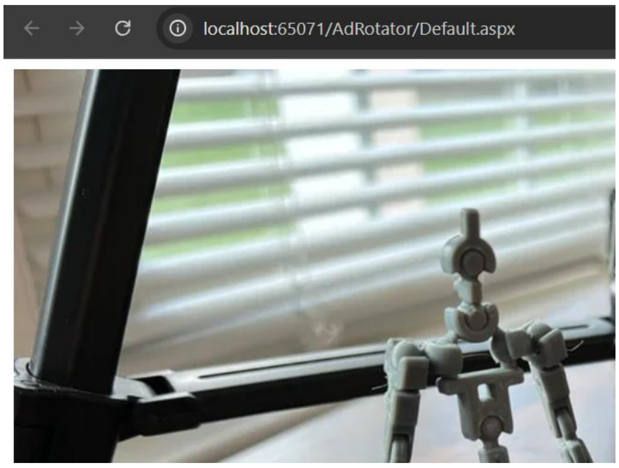
Default.aspx

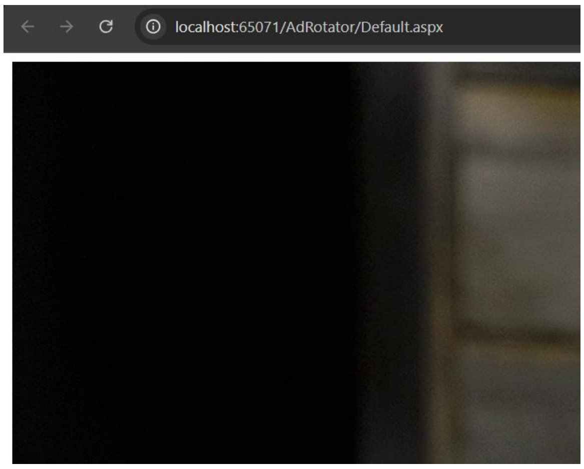
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
%>

<!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>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        <asp:AdRotator ID="AdRotator1" runat="server"
            AdvertisementFile="~/XMLFile.xml" />

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

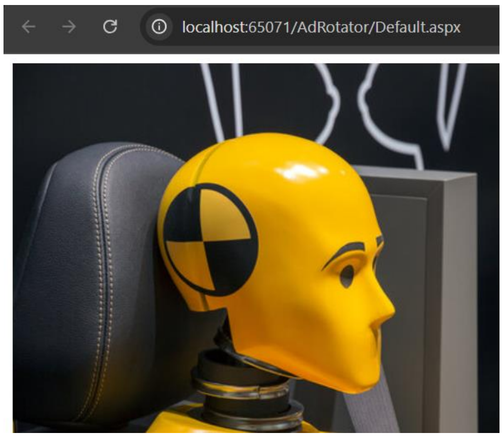
XMLFile.xml

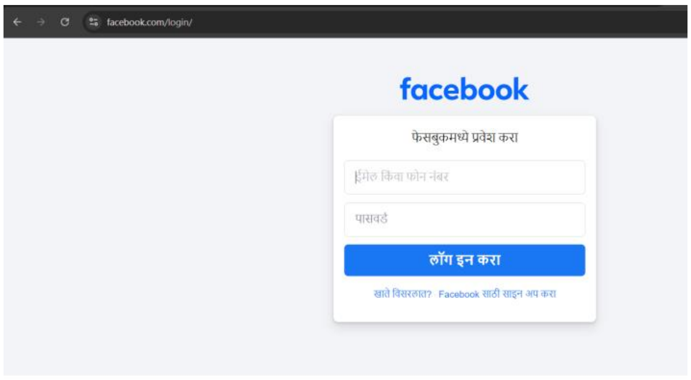
```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
  <Ad>
    <ImageUrl>doll.gif</ImageUrl>
    <AlternateText>Doll Image</AlternateText>
    <NavigateUrl>  https://www.google.com/ </NavigateUrl>
    <Impressions>20</Impressions>
    <Keyword>doll</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>r4.webp</ImageUrl>
    <AlternateText>Toys Image</AlternateText>
    <NavigateUrl>https://www.facebook.com/login/</NavigateUrl>
    <Impressions>20</Impressions>
    <Keyword>Toys</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>robot.jpg</ImageUrl>
    <AlternateText>Robot Image</AlternateText>
    <NavigateUrl>https://www.youtube.com/</NavigateUrl>
    <Impressions>20</Impressions>
    <Keyword>robot</Keyword>
  </Ad>
</Advertisements>
```

OUTPUT:









Conclusion: We have successfully completed the experiment and achieved the desired results.

PRACTICAL-5

A] Aim: take a input from user and check number is positive and apply requirement exception handling.

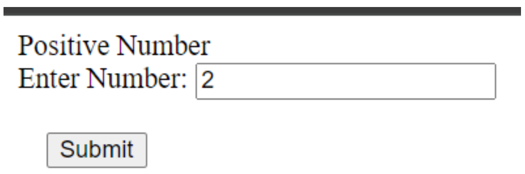
WebForm1.aspx

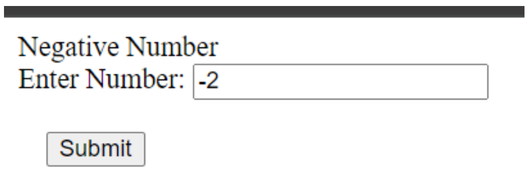
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="P5A.WebForm1" %>

<!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>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label ID="Label1" runat="server" Text="Enter Number:"></asp:Label>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </div>
    <br />
    &nbsp;&nbsp;&nbsp;
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Submit" />
    <br />
    </form>
</body>
</html>
```

WebForm1.aspx.cs

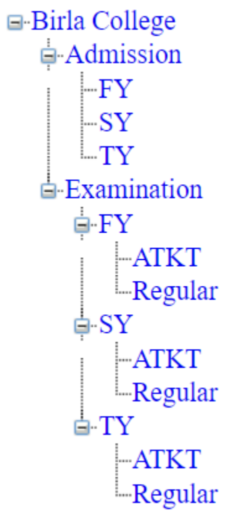
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace P5A
{
    class NegativeException:Exception
    {
        public NegativeException(string msg)
            : base(msg)
        { }
    }
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            int num;
            try
            {
                num = int.Parse(TextBox1.Text);
                if (num < 0)
                {
                    throw new NegativeException("Negative Number");
                }
                else
                {
                    Response.Write("Positive Number");
                }
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

OUTPUT:


Positive Number
Enter Number: 2


Negative Number
Enter Number: -2

B] Aim: create a Tree View Control

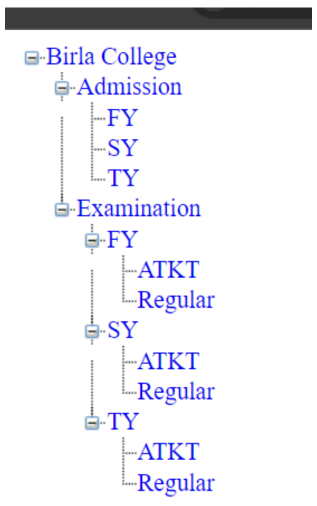

```
Birla College
    Admission
        FY
        SY
        TY
    Examination
        FY
            ATKT
            Regular
        SY
            ATKT
            Regular
        TY
            ATKT
            Regular
```

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="P5B.WebForm1" %>

<!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>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        <asp:TreeView ID="TreeView1" runat="server" ShowLines="True">
            <Nodes>
                <asp:TreeNode Text="Birla College" Value="Birla College">
                    <asp:TreeNode Text="Admission" Value="Admission">
                        <asp:TreeNode Text="FY" Value="FY"></asp:TreeNode>
                        <asp:TreeNode Text="SY" Value="SY"></asp:TreeNode>
                        <asp:TreeNode Text="TY" Value="TY"></asp:TreeNode>
                    </asp:TreeNode>
                    <asp:TreeNode Text="Examination" Value="Examination">
                        <asp:TreeNode Text="FY" Value="FY">
                            <asp:TreeNode Text="ATKT" Value="ATKT"></asp:TreeNode>
                            <asp:TreeNode Text="Regular" Value="Regular"></asp:TreeNode>
                        </asp:TreeNode>
                        <asp:TreeNode Text="SY" Value="SY">
                            <asp:TreeNode Text="ATKT" Value="ATKT"></asp:TreeNode>
                            <asp:TreeNode Text="Regular" Value="Regular"></asp:TreeNode>
                        </asp:TreeNode>
                        <asp:TreeNode Text="TY" Value="TY">
                            <asp:TreeNode Text="ATKT" Value="ATKT"></asp:TreeNode>
                            <asp:TreeNode Text="Regular" Value="Regular"></asp:TreeNode>
                        </asp:TreeNode>
                    </asp:TreeNode>
                </asp:TreeNode>
            </Nodes>
        </asp:TreeView>

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

OUTPUT:



```
⊟Birla College
   ⊟Admission
      FY
      SY
      TY
   ⊟Examination
      ⊟FY
         ATKT
         Regular
      ⊟SY
         ATKT
         Regular
      ⊟TY
         ATKT
         Regular
```

Conclusion: We have successfully completed the experiment and achieved the desired results.

PRACTICAL-6

A] Aim: heirarchial inheritance

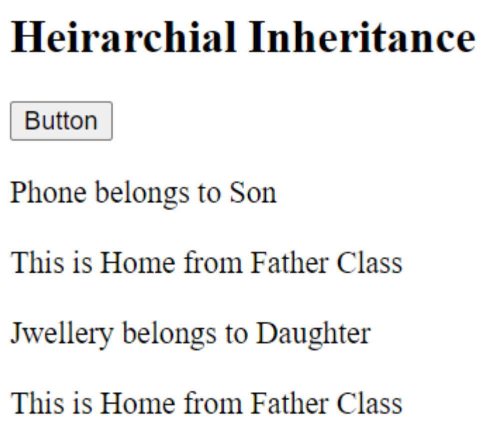
WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="P6A.WebForm1" %>

<!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>
</head>
<body style="height: 209px">
    <form id="form1" runat="server">
    <h2>
        Heirarchial Inheritance
    </h2>
    <div>
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
        <br />
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label3" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
    </div>

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

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace P6A
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            Son obj = new Son();
            Daughter obj1 = new Daughter();
            Label1.Text = Convert.ToString(obj.Phone());
            Label2.Text = Convert.ToString(obj.Home());
            Label3.Text = Convert.ToString(obj1.Jwell());
            Label4.Text = Convert.ToString(obj1.Home());

        }
        public class Father
        {
            public string Home()
            {
                return ("This is Home from Father Class");
            }
        }
        public class Son : Father
        {
            public string Phone()
            {
                return ("Phone belongs to Son");
            }
        }
        public class Daughter : Father
        {
            public string Jwell()
            {
                return ("Jwellery belongs to Daughter");
            }
        }
    }
}
```

OUTPUT:



Heirarchial Inheritance

```
Phone belongs to Son

This is Home from Father Class

Jwellery belongs to Daughter

This is Home from Father Class
```

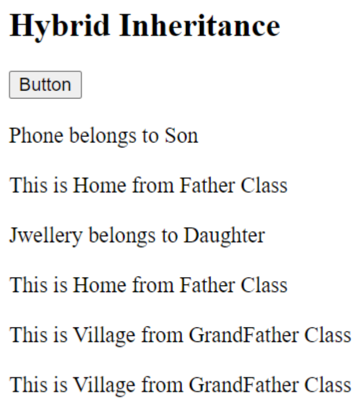
B] Aim: hybrid inheritance

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="P6A.WebForm1" %>

<!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>
</head>
<body style="height: 209px">
    <form id="form1" runat="server">
    <h2>
        Hybrid Inheritance
    </h2>
    <div>
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
        <br />
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label3" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label5" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Label ID="Label6" runat="server" Text="Label"></asp:Label>
    </div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace P6A
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {

            Son obj = new Son();
            Daughter obj1 = new Daughter();
            Label1.Text = Convert.ToString(obj.Phone());
            Label2.Text = Convert.ToString(obj.Home());
            Label3.Text = Convert.ToString(obj1.Jwell());
            Label4.Text = Convert.ToString(obj1.Home());
            Label5.Text = Convert.ToString(obj.Village());
            Label6.Text = Convert.ToString(obj1.Village());

        }
        public class GrandFather
        {
            public string Village()
            {
                return ("This is Village from GrandFather Class");
            }
        }

        public class Father : GrandFather
        {
            public string Home()
            {
                return ("This is Home from Father Class");
            }
        }
        public class Son : Father
        {
            public string Phone()
            {
                return ("Phone belongs to Son");
            }
        }
        public class Daughter : Father
        {
            public string Jwell()
            {
                return ("Jwellery belongs to Daughter");
            }
        }
    }
}
```
OUTPUT:



Hybrid Inheritance

```
Button

Phone belongs to Son

This is Home from Father Class

Jwellery belongs to Daughter

This is Home from Father Class

This is Village from GrandFather Class

This is Village from GrandFather Class
```

Conclusion: We have successfully completed the experiment and achieved the desired results.

PRACTICAL-7

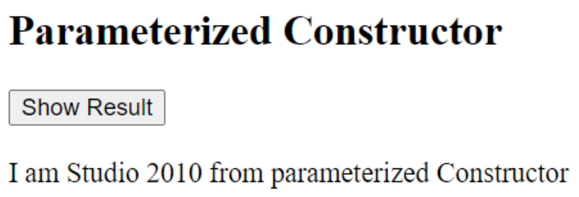
Aim: pramaterized constructor

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="ParameterizedConstructor.WebForm1" %>

<!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>
</head>
<body>
    <form id="form1" runat="server">
    <h2>
        Parameterized Constructor
    </h2>
    <div style="height: 171px">

        <asp:Button ID="Button1" runat="server" Text="Show Result"
            onclick="Button1_Click" />
        <br />
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

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

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace ParameterizedConstructor
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            paramterize obj = new paramterize("Studio", "2010");
            Label1.Text = obj.print();
        }
        public class paramterize
        {
            string name;
            string version;
            public paramterize(string name, string version)
            {
                this.name = name;
                this.version = version;
            }
            public string print()
            {
                return (" I am " + this.name + " " + this.version + " from parameterized 
Constructor");
            }
        }
    }
}
```



Parameterized Constructor

I am Studio 2010 from parameterized Constructor

Conclusion: We have successfully completed the experiment and achieved the desired results.