

Practical 1:

A) Create an application that obtains 4 integer values from the user and displays the product.

Code:

```
using System;
class Product
{
    static void Main()
    {
        int a, b, c, d, product;
        Console.Write("Enter first number: ");
        a = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter second number: ");
        b = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter third number: ");
        c = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter fourth number: ");
        d = Convert.ToInt32(Console.ReadLine());

        product = a * b * c * d;
        Console.WriteLine("Product of numbers = " + product);

        Console.ReadKey();
    }
}
```

Output:

```
Enter first number: 4
Enter second number: 7
Enter third number: 2
Enter fourth number: 9
Product of numbers = 504
```

B) Create an application that displays the Fibonacci series.

Code:

```
using System;
class Fibonacci
{
    static void Main()
    {
        int n, first = 0, second = 1, next;
        Console.Write("Enter number of terms: ");
        n = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Fibonacci Series: ");
        for (int i = 0; i < n; i++)
        {
            if (i <= 1)
                next = i;
            else
            {
                next = first + second;
                first = second;
                second = next;
            }
            Console.Write(next + " ");
        }
        Console.ReadKey();
    }
}
```

Output:

```
Enter number of terms: 8
Fibonacci Series:
0 1 1 2 3 5 8 13 |
```

C) Create an application that obtains an integer from the user and displays whether the number is a prime number or not.

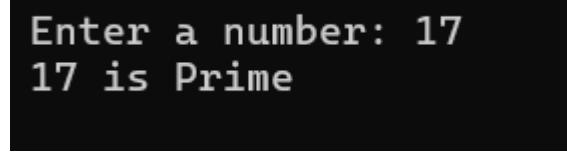
Code:

```
using System;
class Prime
{
    static void Main()
    {
        int n, i, flag = 0;
        Console.Write("Enter a number: ");
        n = Convert.ToInt32(Console.ReadLine());

        for (i = 2; i <= n / 2; i++)
        {
            if (n % i == 0)
            {
                flag = 1;
                break;
            }
        }
        if (flag == 0 && n > 1)
            Console.WriteLine(n + " is Prime");
        else
            Console.WriteLine(n + " is Not Prime");

        Console.ReadKey();
    }
}
```

Output:



```
Enter a number: 17
17 is Prime
```

Practical 2

A) Take four numbers from the user and print the largest number.

Code:

```
using System;
class LargestNumber
{
    static void Main()
    {
        Console.WriteLine("Enter 4 numbers:");
        int a = Convert.ToInt32(Console.ReadLine());
        int b = Convert.ToInt32(Console.ReadLine());
        int c = Convert.ToInt32(Console.ReadLine());
        int d = Convert.ToInt32(Console.ReadLine());

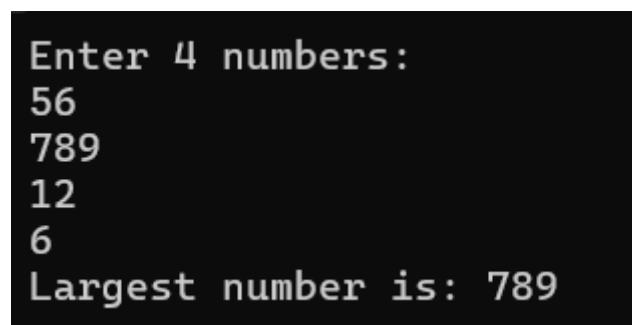
        int largest = a;

        if (b > largest) largest = b;
        if (c > largest) largest = c;
        if (d > largest) largest = d;

        Console.WriteLine("Largest number is: " + largest);

        Console.ReadKey();
    }
}
```

Output:



```
Enter 4 numbers:
56
789
12
6
Largest number is: 789
```

B) Create an application that checks whether an integer is positive or negative

Code:

```
using System;
class PositiveNegative
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        if (num > 0)
            Console.WriteLine("Positive Number");
        else if (num < 0)
            Console.WriteLine("Negative Number");
        else
            Console.WriteLine("Number is ZERO");

        Console.ReadKey();
    }
}
```

Output:

```
Enter a number: -98
Negative Number
```

C) Create an application that checks whether an integer is even or odd.

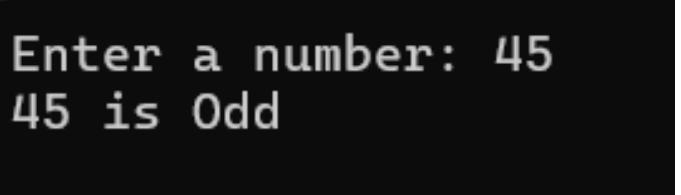
Code:

```
using System;
class EvenOdd
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        if (num % 2 == 0)
            Console.WriteLine(num + " is Even");
        else
            Console.WriteLine(num + " is Odd");

        Console.ReadKey();
    }
}
```

Output:



```
Enter a number: 45
45 is Odd
```

D) Create an application to show the implementation of Switch Case

Code:

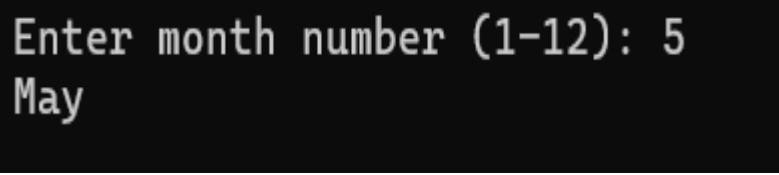
```
using System;
class CalendarSwitch
{
```

```
static void Main()
{
    Console.Write("Enter month number (1-12): ");
    int month = Convert.ToInt32(Console.ReadLine());

    switch (month)
    {
        case 1: Console.WriteLine("January"); break;
        case 2: Console.WriteLine("February"); break;
        case 3: Console.WriteLine("March"); break;
        case 4: Console.WriteLine("April"); break;
        case 5: Console.WriteLine("May"); break;
        case 6: Console.WriteLine("June"); break;
        case 7: Console.WriteLine("July"); break;
        case 8: Console.WriteLine("August"); break;
        case 9: Console.WriteLine("September"); break;
        case 10: Console.WriteLine("October"); break;
        case 11: Console.WriteLine("November"); break;
        case 12: Console.WriteLine("December"); break;
        default: Console.WriteLine("Invalid Month Number"); break;
    }

    Console.ReadKey();
}
```

Output:



```
Enter month number (1-12): 5
May
```

Practical 3

A) Show the use of while loop in C#

Code:

```
using System;
class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Incremented Loop");
        int i = 1;
        while (i <= 10)
        {
            Console.Write(i + "\t");
            i++;
        }

        Console.WriteLine("\nDecrement Loop");
        int j = 10;
        while (j >= 1)
        {
            Console.Write(j + "\t");
            j--;
        }
        Console.ReadKey();
    }
}
```

Output:

Incremented Loop	1	2	3	4	5	6	7	8	9	10
Decrement Loop	10	9	8	7	6	5	4	3	2	1

B) Showcase the use of Do While Loop in C#.**Code:**

```
using System;

class Program
{
    public static void Main(string[] args)
    {
        PrintAscending(1, 10);
        Console.WriteLine(); // line break
        PrintDescending(10, 1);
    }

    // Method to print numbers in ascending order
    static void PrintAscending(int start, int end)
    {
        int i = start;
        do
        {
            Console.Write(i + "\t");
            i++;
        } while (i <= end);
    }

    // Method to print numbers in descending order
    static void PrintDescending(int start, int end)
    {
        int j = start;
        do
        {
            Console.Write(j + "\t");
            j--;
        } while (j >= end);
    }
}
```

Output:

1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1
1	2	3	4	5	6	7	8	9	10

C) Display the use of For and For Each Loop in C#.

Code:

```
using System;
class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Incremented Loop");
        for (int i = 1; i <= 10; i++)
        {
            Console.Write(i + "\t");
        }

        Console.WriteLine("\nDecrementated Loop");
        for (int j = 10; j >= 1; j--)
        {
            Console.Write(j + "\t");
        }

        Console.WriteLine("\nFor Each Loop");
        int[] num = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

        foreach (int s in num)
        {
            Console.Write(s + "\t");
        }
    }
}
```

Output:

Incremented Loop
1 2 3 4 5 6 7 8 9 10
Decrementated Loop
10 9 8 7 6 5 4 3 2 1
For Each Loop
1 2 3 4 5 6 7 8 9 10

Practical 4

A) Create an application that takes a number from the user and returns its factorial.

Code:

```
using System;
class Program
{
    public static void Main(string[] args)
    {
        Console.Write("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int fact = 1;

        for (int i = 1; i <= num; i++)
        {
            fact = fact * i;
        }

        Console.WriteLine("Factorial of " + num + " is: " + fact);
        Console.ReadKey();
    }
}
```

Output:

```
Enter a number: 9
Factorial of 9 is: 362880
```

B) Create a money Conversion application in C#.

Code:

```
using System;
class Program
{
    public static void Main(string[] args)
    {
        Console.Write("Enter amount in USD: ");
        double usd = Convert.ToDouble(Console.ReadLine());

        double inr = usd * 88;
        Console.WriteLine("Amount in INR = " + inr);

        Console.ReadKey();
    }
}
```

Output:

```
Enter amount in USD: 100
Amount in INR = 8800
i
```

C) Create a temperature Conversion application in C#.

Code:

```
using System;
class Program
{
    public static void Main(string[] args)
    {
        Console.Write("Enter temperature in Celsius: ");
        double c = Convert.ToDouble(Console.ReadLine());
        double f = (c * 9 / 5) + 32;
```

```

Console.WriteLine("Temperature in Fahrenheit = " + f);

Console.Write("\nEnter temperature in Fahrenheit: ");
f = Convert.ToDouble(Console.ReadLine());
c = (f - 32) * 5 / 9;
Console.WriteLine("Temperature in Celsius = " + c);

Console.ReadKey();
}
}

```

Output:

```

Enter temperature in Celsius: 98
Temperature in Fahrenheit = 208.4

Enter temperature in Fahrenheit: 98
Temperature in Celsius = 36.666666666666664

```

Practical 5

A) Display Function Overloading in C#.

Code:

```

using System;
namespace Overloading
{
    class SwapDemo
    {
        public void Swap(int a, int b)
        {
            int temp = a;

```

```

        a = b;
        b = temp;
        Console.WriteLine("After Swapping (int): a = " + a + ", b = " + b);
    }

    public void Swap(double a, double b)
    {
        double temp = a;
        a = b;
        b = temp;
        Console.WriteLine("After Swapping (double): a = " + a + ", b = " + b);
    }
}

class MainClass
{
    static void Main()
    {
        SwapDemo s = new SwapDemo();

        Console.WriteLine("Function Overloading Example (Swapping)\n");

        s.Swap(10, 20);
        s.Swap(12.5, 45.7);

        Console.ReadKey();
    }
}

```

Output:

```
Function Overloading Example (Swapping)
```

```
After Swapping (int): a = 20, b = 10
```

```
After Swapping (double): a = 45.7, b = 12.5
```

B) Showcase Parameterized Constructor in C#.

Code:

```
using System;
```

```
namespace Constructor
```

```
{
```

```
    class paraconstrctor
```

```
{
```

```
    public paraconstrctor(int x, int y)
```

```
{
```

```
        int c = x + y;
```

```
        Console.WriteLine("The Sum of the two numbers is "+c);
```

```
}
```

```
}
```

```
class MainClass
```

```
{
```

```
    static void Main()
```

```
{
```

```
    Console.WriteLine("\t");
```

```
    Console.WriteLine("Enter the first Number: ");
```

```
    int a = Convert.ToInt32(Console.ReadLine());
```

```
    Console.WriteLine("\nEnter the Second Number: ");
```

```
    int b = Convert.ToInt32(Console.ReadLine());
```

```

    paraconstrctor v = new paraconstrctor(a, b);

    Console.Read();
}

}
}

```

Output:

```

Enter the first Number:
35

Enter the Second Number:
75
The Sum of the two numbers is 110

```

C) Show Copy Constructor in C#.

Code:

```
using System;
```

```

namespace copyConstructor
{
    class employee
    {
        private string name;
        private int age;

        public employee(employee emp)
        {
            name = emp.name;
            age = emp.age;
        }
    }
}
```

```

public employee(string name, int age)
{
    this.name = name;
    this.age = age;
}

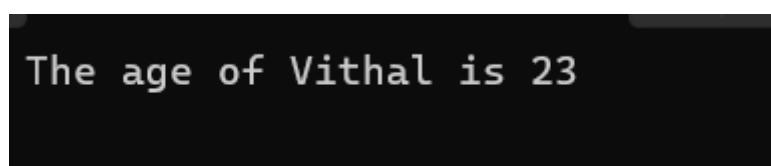
public string Details
{
    get
    {
        return "The age of " + name + " is " + age.ToString();
    }
}
}

class empdetail
{
    static void Main()
    {
        employee emp1 = new employee("Vithal", 23);
        employee emp2 = new employee(emp1);

        Console.WriteLine(emp2.Details);
        Console.ReadLine();
    }
}

```

Output:



Practical 6 Calendar Control

Code:

calndrCtrl.aspx:

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Calendar Control Demo</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:Calendar ID="Calendar1" runat="server"
            BackColor="#FFFFCC"
            BorderColor="#FFCC66"
            BorderWidth="1px"
            DayNameFormat="Shortest"
            Font-Names="Verdana"
            Font-Size="8pt"
            ForeColor="#663399"
            Height="200px"
            Width="300px"
            NextPrevFormat="ShortMonth"
            ShowGridLines="True"
            OnDayRender="Calendar1_DayRender"
            OnSelectionChanged="Calendar1_SelectionChanged">

            <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
            <NextPrevStyle BorderStyle="Solid" BorderWidth="2px" Font-Size="9pt" ForeColor="#FFFFCC"
/>
            <OtherMonthDayStyle BackColor="#FFCC99" BorderStyle="Solid" ForeColor="#CC9966" />
            <SelectedDayStyle BackColor="Red" Font-Bold="True" />
    
```

```

<SelectorStyle BackColor="#FFCC66" />
<TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt" ForeColor="#FFFFCC" />
<TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
<WeekendDayStyle Height="50px" />
</asp:Calendar>

<br />
<asp:Button ID="btnResult" runat="server" Text="Show Result" OnClick="btnResult_Click" />
<asp:Button ID="btnReset" runat="server" Text="Reset" OnClick="btnReset_Click" />

<br /><br />
<asp:Label ID="Label1" runat="server"></asp:Label><br />
<asp:Label ID="Label2" runat="server"></asp:Label><br />
<asp:Label ID="Label3" runat="server"></asp:Label><br />
<asp:Label ID="Label4" runat="server"></asp:Label><br />
<asp:Label ID="Label5" runat="server"></asp:Label><br />
</form>
</body>
</html>

```

calndrCtrl.aspx.cs:

```

using System;
using System.Web.UI.WebControls;

namespace CalendarControlDemo
{
    public partial class calndrCtrl : System.Web.UI.Page
    {
        protected void btnResult_Click(object sender, EventArgs e)
        {
            Calendar1.Caption = "xyz";
            Calendar1.FirstDayOfWeek = FirstDayOfWeek.Sunday;
        }
    }
}

```

```

Calendar1.NextPrevFormat = NextPrevFormat.ShortMonth;
Calendar1.TitleFormat = TitleFormat.Month;

Label2.Text = "Today's Date: " + Calendar1.TodaysDate.ToShortDateString();
Label3.Text = "Ganpati Vacation Start: 13-09-2018";

TimeSpan d = new DateTime(2018, 9, 13) - DateTime.Now;
Label4.Text = "Days Remaining For Ganpati Vacation: " + d.Days.ToString();

TimeSpan d1 = new DateTime(2018, 12, 31) - DateTime.Now;
Label5.Text = "Days Remaining for New Year: " + d1.Days.ToString();

if (Calendar1.SelectedDate == new DateTime(2018, 9, 13))
    Label3.Text = "<b>Ganpati Festival Start</b>";
if (Calendar1.SelectedDate == new DateTime(2018, 9, 23))
    Label3.Text = "<b>Ganpati Festival End</b>";
}

protected void Calendar1_DayRender(object sender,
System.Web.UI.WebControls.DayRenderEventArgs e)
{
    // Teachers Day - 5th September
    if (e.Day.Date.Day == 5 && e.Day.Date.Month == 9)
    {
        e.Cell.BackColor = System.Drawing.Color.Yellow;
        Label lbl = new Label();
        lbl.Text = "<br>Teachers Day!";
        e.Cell.Controls.Add(lbl);

        Image g1 = new Image();
        g1.ImageUrl = "td.jpg";
        g1.Height = 20;
        g1.Width = 20;
    }
}

```

```

        e.Cell.Controls.Add(g1);
    }

    // Ganpati Festival Highlight (13th Sept onwards)
    if (e.Day.Date.Day == 13 && e.Day.Date.Month == 9)
    {
        Calendar1.SelectedDate = new DateTime(2018, 9, 13);
        Calendar1.SelectedDates.SelectRange(Calendar1.SelectedDate,
        Calendar1.SelectedDate.AddDays(10));

        Label lbl1 = new Label();
        lbl1.Text = "<br>Ganpati!";
        e.Cell.Controls.Add(lbl1);
    }
}

protected void btnReset_Click(object sender, EventArgs e)
{
    Label1.Text = "";
    Label2.Text = "";
    Label3.Text = "";
    Label4.Text = "";
    Label5.Text = "";
    Calendar1.SelectedDates.Clear();
}

protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    Label1.Text = "Your Selected Date: " + Calendar1.SelectedDate.ToShortDateString();
}

}
}

```

Output:

Aug		September 2025					Oct	
Su	Mo	Tu	We	Th	Fr	Sa		
31	1	2	3	4	5 Teachers Day! 	6		
7	8	9	10	11	12	13 Ganpati!		
14	15	16-17 Calendar		18	19	20		
21	22	23	24	25	26	27		
28	29	30	1	2	3	4		
5	6	7	8	9	10	11		

Your Selected Date: 17-09-2025

Practical 7 Adrotator

Code:

Default.aspx(markup):

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

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Ad Rotator Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:AdRotator ID="AdRotator1" runat="server">
```

```

        DataSourceID="XmlDataSource1"
        Target="_blank"
        Width="300px" Height="250px" />

<asp:XmlDataSource ID="XmlDataSource1" runat="server"
    DataFile="~/App_Data/ADFILE.xml"
    XPath="Advertisements/Ad">
</asp:XmlDataSource>
</form>
</body>
</html>

```

ADFILE.xml:

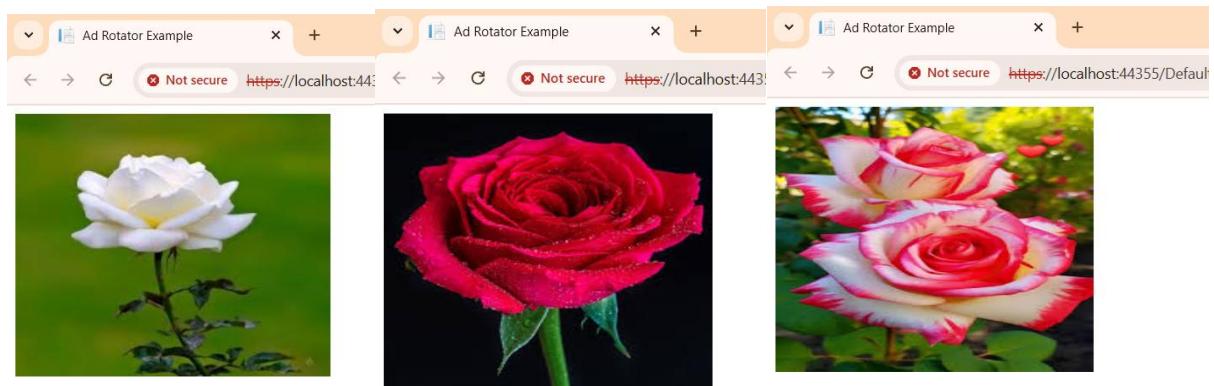
```

<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
    <Ad>
        <ImageUrl>~/images/rose1.jpg</ImageUrl>
        <NavigateUrl>http://www.1800flowers.com</NavigateUrl>
        <AlternateText>Order flowers, roses, gifts and more</AlternateText>
        <Impressions>20</Impressions>
        <Keyword>flowers</Keyword>
    </Ad>
    <Ad>
        <ImageUrl>~/images/rose2.jpg</ImageUrl>
        <NavigateUrl>http://www.babybouquets.com.au</NavigateUrl>
        <AlternateText>Order roses and flowers</AlternateText>
        <Impressions>20</Impressions>
        <Keyword>gifts</Keyword>
    </Ad>
    <Ad>
        <ImageUrl>~/images/rose3.jpg</ImageUrl>
        <NavigateUrl>http://www.flowers2moscow.com</NavigateUrl>
        <AlternateText>Send flowers to Russia</AlternateText>
        <Impressions>20</Impressions>
    </Ad>
</Advertisements>

```

```
<Keyword>russia</Keyword>
</Ad>
</Advertisements>
```

Output:



Practical 8 User Control

Code:

MyUserControl.ascx:

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="MyUserControl.ascx.cs"
Inherits="UserControlDemo.MyUserControl" %>
```

```
<h3>This is User Control</h3>
<table>
  <tr>
    <td>Name</td>
    <td><asp:TextBox ID="txtName" runat="server" /></td>
  </tr>
  <tr>
    <td>City</td>
```

```

<td><asp:TextBox ID="txtCity" runat="server" /></td>
</tr>
<tr>
<td></td>
<td>
<asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
</td>
</tr>
</table>
<br />
<asp:Label ID="Label1" runat="server" ForeColor="Black" />

```

MyUserControl.ascx.cs:

```

using System;
using System.Web.UI;

namespace UserControlDemo
{
    public partial class MyUserControl : UserControl
    {
        protected void btnSave_Click(object sender, EventArgs e)
        {
            Label1.Text = "Your Name is " + txtName.Text + " and you are from " + txtCity.Text;
        }
    }
}

```

UserControlDisplay.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="UserControlDisplay.aspx.cs"
Inherits="UserControlDemo.UserControlDisplay" %>
<%@ Register Src="~/MyUserControl.ascx" TagPrefix="uc" TagName="Student" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">

```

```
<head runat="server">
    <title>User Control Demo</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <uc:Student ID="studentcontrol" runat="server" />
        </div>
    </form>
</body>
</html>
```

Output:

This is User Control

Name	<input type="text"/>
City	<input type="text"/>
<input type="button" value="Save"/>	

This is User Control

Name	<input type="text" value="Rajesh"/>
City	<input type="text" value="Mumbai"/>
<input type="button" value="Save"/>	

Your Name is Rajesh and you are from Mumbai

Practical 9 View State

Code:

Default.aspx:

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ViewState Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="margin:20px">
            <asp:Label ID="Label1" runat="server" Text="Name: "></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />
            <asp:Label ID="Label2" runat="server" Text="Password: "></asp:Label>
            <asp:TextBox ID="TextBox2" runat="server" TextMode="Password"></asp:TextBox><br /><br />
            <asp:Button ID="Button1" runat="server" Text="Save to ViewState & Clear"
                OnClick="Button1_Click" />
            <asp:Button ID="Button3" runat="server" Text="Load from ViewState"
                OnClick="Button3_Click" /><br /><br />
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;
namespace ViewStateDemo
{
```

```

public partial class _Default : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        // Store textbox values in ViewState
        ViewState["name"] = TextBox1.Text;
        ViewState["password"] = TextBox2.Text;

        // Clear the textboxes
        TextBox1.Text = string.Empty;
        TextBox2.Text = string.Empty;
    }

    protected void Button3_Click(object sender, EventArgs e)
    {
        // Retrieve values from ViewState if they exist
        if (ViewState["name"] != null)
        {
            TextBox1.Text = ViewState["name"].ToString();
        }
        if (ViewState["password"] != null)
        {
            TextBox2.Text = ViewState["password"].ToString();
        }
    }
}

```

Output:

Name:

Password:

After Clicking Load from ViewState:

Name:

Password:

Practical 10 Validation Control

Code:

RegistrationForm.aspx:

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

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>User Registration Form</title>
    <style>
        body { font-family: Arial; margin: 40px; background-color: #f9f9f9; }
        h2 { color: #333; }
        .form-container { background: #fff; padding: 20px; width: 400px; border-radius: 10px; box-
shadow: 0 2px 5px rgba(0,0,0,0.2); }
        .form-group { margin-bottom: 15px; }
        .form-label { display: block; font-weight: bold; margin-bottom: 5px; }
        .error { color: red; font-size: 13px; }
        .success { color: green; font-weight: bold; }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="form-container">
            <h2>User Registration</h2>

            <!-- Full Name -->
            <div class="form-group">
                <label class="form-label">Full Name:</label>
                <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvName" runat="server"
                    ControlToValidate="txtName" ErrorMessage="Full Name is required"
                    CssClass="error"></asp:RequiredFieldValidator>
```

```

</div>

<!-- Email -->
<div class="form-group">
    <label class="form-label">Email:</label>
    <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="rfvEmail" runat="server"
        ControlToValidate="txtEmail" ErrorMessage="Email is required"
        CssClass="error"></asp:RequiredFieldValidator>
    <asp:RegularExpressionValidator ID="revEmail" runat="server"
        ControlToValidate="txtEmail"
        ValidationExpression="^[\w+@[a-zA-Z_]+\.[a-zA-Z]{2,3}$"
        ErrorMessage="Invalid Email Format"
        CssClass="error"></asp:RegularExpressionValidator>
</div>

<!-- Password -->
<div class="form-group">
    <label class="form-label">Password:</label>
    <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
    <asp:RequiredFieldValidator ID="rfvPassword" runat="server"
        ControlToValidate="txtPassword" ErrorMessage="Password is required"
        CssClass="error"></asp:RequiredFieldValidator>
</div>

<!-- Confirm Password -->
<div class="form-group">
    <label class="form-label">Confirm Password:</label>
    <asp:TextBox ID="txtConfirmPassword" runat="server"
        TextMode="Password"></asp:TextBox>
    <asp:CompareValidator ID="cvPassword" runat="server"
        ControlToValidate="txtConfirmPassword" ControlToCompare="txtPassword"
        ErrorMessage="Passwords do not match" CssClass="error"></asp:CompareValidator>
</div>

```

```

<!-- Age -->
<div class="form-group">
    <label class="form-label">Age:</label>
    <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
    <asp:RangeValidator ID="rvAge" runat="server"
        ControlToValidate="txtAge" MinimumValue="18" MaximumValue="60" Type="Integer"
        ErrorMessage="Age must be between 18 and 60"
        CssClass="error"></asp:RangeValidator>
</div>

<!-- Phone -->
<div class="form-group">
    <label class="form-label">Phone Number:</label>
    <asp:TextBox ID="txtPhone" runat="server"></asp:TextBox>
    <asp:RegularExpressionValidator ID="revPhone" runat="server"
        ControlToValidate="txtPhone"
        ValidationExpression="^([6-9]\d{9})$"
        ErrorMessage="Enter a valid 10-digit phone number"
        CssClass="error"></asp:RegularExpressionValidator>
</div>

<!-- Terms -->
<div class="form-group">
    <asp:CheckBox ID="chkTerms" runat="server" Text="I accept the Terms & Conditions" />
    <asp:CustomValidator ID="cvTerms" runat="server"
        ErrorMessage="You must accept Terms"
        OnServerValidate="cvTerms_ServerValidate" CssClass="error"></asp:CustomValidator>
</div>

<!-- Validation Summary -->
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
    HeaderText="⚠ Please fix the following errors:" CssClass="error" />

<br />

```

```

    <asp:Button ID="btnRegister" runat="server" Text="Register" OnClick="btnRegister_Click"
/>
    <br /><br />
    <asp:Label ID="lblMessage" runat="server"></asp:Label>
</div>
</form>
</body>
</html>

```

RegistrationForm.aspx.cs:

```

using System;

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

        // Custom validator for Terms checkbox
        protected void cvTerms_ServerValidate(object source,
System.Web.UI.WebControls.ServerValidateEventArgs args)
        {
            args.IsValid = chkTerms.Checked;
        }

        // On Register button click
        protected void btnRegister_Click(object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                lblMessage.Text = "✓ Registration Successful! Welcome, " + txtName.Text;
                lblMessage.CssClass = "success";
            }
        }
    }
}

```

```
        }
    }
}
}
```

Output:

User Registration

Full Name:

Email:

Password:

Confirm Password:

Age:

Phone Number:

I accept the Terms & Conditions

 **Registration Successful! Welcome, Rohan Gupta**

User Registration

Full Name:

Full Name is required

Email:

Email is required

Password:

Password is required

Confirm Password:

Age:

Phone Number:

I accept the Terms & Conditions

 Please fix the following errors:

- Full Name is required
- Email is required
- Password is required

Practical 11 Various States of ASP.NET Pages

Code:

PagesStateDemo.aspx:

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ASP.NET Page States Demo</title>
    <style>
        body { font-family: Arial; margin: 20px; }
        .box { margin-bottom: 15px; padding: 10px; border: 1px solid #ccc; width: 400px; }
        .heading { font-weight: bold; color: darkblue; }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <h2>ASP.NET Page States Demonstration</h2>

        <!-- ViewState -->
        <div class="box">
            <span class="heading">ViewState Example:</span><br />
            Counter: <asp:Label ID="lblCounter" runat="server" Text="0"></asp:Label><br />
            <asp:Button ID="btnIncrease" runat="server" Text="Increase Counter"
                OnClick="btnIncrease_Click" />
        </div>

        <!-- Session State -->
        <div class="box">
            <span class="heading">Session State Example:</span><br />
            Enter your name: <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
        </div>
    </form>
</body>
```

```

<asp:Button ID="btnSaveSession" runat="server" Text="Save in Session"
OnClick="btnSaveSession_Click" /><br />
<asp:Label ID="lblSessionValue" runat="server" ForeColor="Green"></asp:Label>
</div>

<!-- Application State -->
<div class="box">
    <span class="heading">Application State Example:</span><br />
    Total Visitors: <asp:Label ID="lblVisitors" runat="server"></asp:Label>
</div>

<!-- QueryString -->
<div class="box">
    <span class="heading">QueryString Example:</span><br />
    <asp:HyperLink ID="hlQuery" runat="server"
        NavigateUrl("~/PageStatesDemo.aspx?course=ASP.NET"
        Text="Click here with QueryString"></asp:HyperLink><br />
    <asp:Label ID="lblQueryString" runat="server" ForeColor="Purple"></asp:Label>
</div>
</form>
</body>
</html>

```

PagesStateDemo.aspx.cs:

```

using System;

namespace StateDemo
{
    public partial class PageStatesDemo : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Application State (Count Visitors)
            if (Application["Visitors"] == null)

```

```
Application["Visitors"] = 1;
else
    Application["Visitors"] = (int)Application["Visitors"] + 1;

lblVisitors.Text = Application["Visitors"].ToString();

// QueryString Example
if (Request.QueryString["course"] != null)
{
    lblQueryString.Text = "You selected course: " + Request.QueryString["course"];
}

// ViewState Example
protected void btnIncrease_Click(object sender, EventArgs e)
{
    int count = 0;
    if (ViewState["Counter"] != null)
    {
        count = (int)ViewState["Counter"];
    }
    count++;
    ViewState["Counter"] = count;
    lblCounter.Text = count.ToString();
}

// Session Example
protected void btnSaveSession_Click(object sender, EventArgs e)
{
    Session["UserName"] = txtName.Text;
    lblSessionValue.Text = "Hello, " + Session["UserName"];
}
```

Output:

ASP.NET Page States Demonstration

ViewState Example:

Counter: 3

[Increase Counter](#)

Session State Example:

Enter your name:

[Save in Session](#)

Hello, Kiran Bhatt

Application State Example:

Total Visitors: 5

QueryString Example:

[Click here with QueryString](#)

Practical 12 Website Navigation controls and Site Map

Code:

Web.sitemap:

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0">
    <siteMapNode url "~/Home.aspx" title="Home" description="Home Page">
        <siteMapNode url "~/About.aspx" title="About Us" description="About Page"/>
        <siteMapNode url "~/Courses.aspx" title="Courses" description="Courses Page"/>
        <siteMapNode url "~/BScIT.aspx" title="B.Sc IT" description="IT Course"/>
        <siteMapNode url "~/AIML.aspx" title="AI &amp; ML" description="AI ML
Course"/>
    </siteMapNode>
    <siteMapNode url "~/Contact.aspx" title="Contact Us" description="Contact Page"/>
    </siteMapNode>
</siteMap>
```

Site.Master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="NavDemo.Site" %>
```

```
<!DOCTYPE html>
<html>
<head runat="server">
    <title>Navigation Demo Website</title>
    <style>
        body { font-family: Arial; margin: 0; padding: 0; }
        .header { background: #003366; color: white; padding: 15px; }
        .nav { background: #f0f0f0; padding: 10px; }
        .content { margin: 20px; }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="header">
            <h2>Navigation Controls Demo</h2>
        </div>

        <!-- Navigation Menu -->
        <div class="nav">
            <asp:Menu ID="Menu1" runat="server" DataSourceID="SiteMapDataSource1"
                Orientation="Horizontal" StaticDisplayLevels="2"
                BackColor="#e0e0e0" ForeColor="Black">
            </asp:Menu>
            <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />
        </div>

        <!-- Breadcrumb (SiteMapPath) -->
        <div style="margin:10px 20px;">
            <asp:SiteMapPath ID="SiteMapPath1" runat="server" PathSeparator=" > " />
        </div>
```

```
<!-- Page Content -->
<div class="content">
    <asp:ContentPlaceHolder ID="MainContent" runat="server" />
</div>
</form>
</body>
</html>
```

Home.aspx:

```
<%@ Page Title="Home" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="Home.aspx.cs" Inherits="NavDemo.Home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Welcome to the Home Page</h2>
    <p>This is the landing page of the Navigation Demo Website.</p>
</asp:Content>
```

About.aspx:

```
<%@ Page Title="About Us" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="About.aspx.cs" Inherits="NavDemo.About" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h2>About Us</h2>
    <p>Learn more about our organization and goals.</p>
</asp:Content>
```

Courses.aspx:

```
<%@ Page Title="Courses" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Courses.aspx.cs" Inherits="NavDemo.Courses" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Courses Offered</h2>
    <p>Explore the programs we offer in IT and AI/ML.</p>
</asp:Content>
```

BScIT.aspx:

```
<%@ Page Title="B.Sc IT" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="BScIT.aspx.cs" Inherits="NavDemo.BScIT" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h2>B.Sc IT</h2>
    <p>This course covers core Information Technology concepts.</p>
</asp:Content>
```

AIML.aspx:

```
<siteMapNode url="~/Courses.aspx" title="Courses" description="Courses Page">
    <siteMapNode url="~/BScIT.aspx" title="B.Sc IT" description="IT Course"/>
    <siteMapNode url="~/AIML.aspx" title="AI &amp; ML" description="AI ML Course"/>
</siteMapNode>
<siteMapNode url="~/Contact.aspx" title="Contact Us" description="Contact Page"/>
```

Contact.aspx:

```
<%@ Page Title="Contact Us" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Contact.aspx.cs" Inherits="NavDemo.Contact" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Contact Us</h2>
    <p>Reach out to us through this page.</p>
</asp:Content>
```

Output:

The screenshot shows a website layout with a dark blue header bar. The header contains the text "Navigation Controls Demo". Below the header, there is a navigation bar with links: "Home", "About Us", "Courses", and "Contact Us".

The main content area has three sections:

- Welcome to the Home Page**: This section contains the text "Welcome to the Home Page" and a note: "This is the landing page of the Navigation Demo Website."
- Courses Offered**: This section contains the text "Explore the programs we offer in IT and AI/ML."
- Contact Us**: This section contains the text "Reach out to us through this page."