

Lab Manual

.NET

PRITESH SENGARA

160470107050

VVPEC CE Sem-6

Contents

Practical 1	1
Introduction to c#	1
Practical 2	9
Print given pattern	9
Print given pattern	11
Prompt a user to input his/her name and country name and print on console	13
Demonstrate inheritance to define Car class and derive Maruti and Mahindra	14
Practical 3	17
Method overloading	17
Constructor overloading	21
Practical 4	23
Reflection Api	23
Practical 5	26
Copy data from one file to another using StreamReader and StreamWriter class.	26
Write a C# Program to Read Lines from a File until the End of File is Reached.	28
List Files in a Directory	30
Practical 6	32
Windows Form Application for Student Registration and store student Details in DataBase.	32
Practical 7	36
Perform validation using Validation Controls	36
Practical 8	40
Introduction to Master Pages	40

Practical 1

Aim:

Introduction to c#

Variables:

Initialization

Scope

Constant

Flow Control

Conditional Statements(if, switch)

Loop(for, while, dowhile, foreach)

Jump(goto, break, continue, return)

Eumerations

Passing Arguments

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using

System.Threading;

namespace P1

{

    class P1

    {

        static int j = 90;

        static void Main(string[] args)

        {

            Console.WriteLine("First

            Program"); int i = 25;

            Console.WriteLine("Scope of Variables.\n1:");
```

```

    for (int j = 0; j < 2; j++)
    {
        Console.WriteLine("{0} {1}\n", j, P1.j);
    }
    Console.WriteLine("2:
"); for (int k =
0; k < 3; k++)
    {
        Console.WriteLine("{0} ", k);
    }
    Console.WriteLine("\n");
    for (int k = 3; k > 0; k--)
    {
        Console.WriteLine("{0} ", k);
    }
    Console.WriteLine("Const
ants"); const int
valConst = 100;
    Console.WriteLine("{0} is constant value", valConst);
    //const int valConst2 = valConst + 9;
        //Console.WriteLine("Another Constant:
{0}", valConst2); Console.WriteLine("\nPredefined
Data Types\n\nValue Types and Reference
Types");

    int valI = 2, valJ = valI;
    Console.WriteLine("valI is: {0} and valJ is:
{1}", valI, valJ); valJ = 90;

```

```

    Console.WriteLine("val i is: {0} and val j is:
    {1}", val i, val j); Vector x, y;

    x = new Vector();
    x.value = 3;
    y = x;

    Console.WriteLine("x is: {0} and y is:{1}",
    x.value, y.value); y.value = 234;

    Console.WriteLine("x is: {0} and y is:{1}", x.value,
    y.value);
    y = null;

    Console.WriteLine("\nInteger
    Types"); sbyte sb = 33;

    short s = 33; int
    _i = 33; long l
    = 33L;

    //Unsigned
    Integers byte
    b = 33;

    ushort us =
    33; uint ui
    = 33U; ulong
    ul = 33UL;

    Console.WriteLine("{0} {1} {2} {3} {4} {5} {6} {7}", sb, s,
    _i, l, b, us,
    ui, ul);

    //Floating point
    types float f =
    11.22334455F;

    double d = 11.2233445566778899;

    Console.Write("\nFloat and

```

```

    Double:\n");

    Console.WriteLine("{0} and
    \n{1}", f, d);

    //Decimal Type
    decimal dec = 111.222333444555666777888999M;
    Console.WriteLine("Decimal:\n{0}", dec);

    //Boolean
    Console.WriteLine("\nBoo
    lean:");

    bool valBoolean = true;
    Console.WriteLine("Status: " +
    valBoolean);

    //Character
    Console.WriteLine("\nCharacter: \nSingle
    Quote \' '); Console.WriteLine("Double
    Quote '\" '); Console.WriteLine("Back Slash
    \\");

    char charA = 'A';
    Console.WriteLine(char
    A); charA = '\0';
    Console.WriteLine("Now null: "
    + charA);
    Console.WriteLine("\a");
    Thread.Sleep(1000);
    Console.Beep(); //another
    notification sound object o1 =
    "Hi, I am an Object";

```

```

object o2 = 34;

string strObj = o1
as string;

Console.WriteLine(strObj);

Console.WriteLine(o1.GetHashCode() + " "
+ o1.GetType());
Console.WriteLine(o2.GetHashCode() + " "
+ o2.GetType());

Console.WriteLine(o1.Equals(o2));

string s1, s2; s1
= "String 1"; s2
= s1;

Console.WriteLine("S1 is: {0} and s2 is {1}", s1, s2); s2
= "New String 1";

Console.WriteLine("S1 is: {0} and s2 is {1}", s1, s2); s1
= "c:\\NewFolder\\Hello\\P1.cs"; Console.WriteLine(s1);
s1 =
@"c:\\NewFolder\\Hello\\P1.
cs";

Console.WriteLine(s1);

s1 = @"We can also
write like
this";

Console.WriteLine(s1); bool
isZero;

Console.WriteLine("\nFlow Control:

```

```

    (i f)\n i s " + i ); i f (i == 0)
    {
        isZero = true;
        Console.WriteLine("i is
        Zero");
    }
    else
    {
        isZero = false;
        Console.WriteLine("i is Non - zero");
    }
    Console.WriteLine("\nType in a
    string:"); string input;
    input =
    Console.ReadLine();
    i f (input == "")
    {
        Console.WriteLine("You typed in an empty string");
    }
    else i f (input.Length < 5)
    {
        Console.WriteLine("The string had less than 5 characters");
    }
    else i f (input.Length < 10)
    {
        Console.WriteLine("The string had at least 5 but less than 10
characters");
    }

```



```
Console.WriteLine("The string was " + input);

//Switch

int integerA = 2;

Console.WriteLine("\nSwitch:"); switch
(integerA)
{
    case 1:
        Console.WriteLine("integerA
        = 1"); break;
    case 2:
        Console.WriteLine("integerA
        = 2"); break;
    case 3:
        Console.WriteLine("integerA
        = 3"); break;
    default:
        Console.WriteLine("integerA is not 1, 2, or 3"); break;
}

WriteGreeting(TimeOfDay.Afternoon);

Console.WriteLine("Argument is: {0}",
args[0]);
}

public enum TimeOfDay
{
    Morning = 0,
    Afternoon = 1,
    Evening = 2
}
```

```
    }

    static void WriteGreeting(TimeOfDay timeOfDay)
    {
        switch (timeOfDay)
        {
            case TimeOfDay.Morning:
                Console.WriteLine("Good
                morning!"); break;
            case TimeOfDay.Afternoon:
                Console.WriteLine("Good
                afternoon!"); break;
            case TimeOfDay.Evening:
                Console.WriteLine("Good
                evening!"); break;
            default:
                Console.WriteLine("Hell
                o!"); break;
        }
    }
}

public class Vector
{
    public int value;
}
}
```

Output:

```
E:\Sem-
6\VS>p1.exe
FirstProgram
ScopeofVariables.
```

```
1:
0 90
1 90
2:
0 1 2
3 2 1 Constants
100is
constantvalue
AnotherConstant:1
09
```

PredefinedDataTypes

```
ValueTypesandReferenceTyp
es valiis:2 andvaljis:2
valiis:2
andvaljis:90 x
is:3andyis:3
x is:234andy is:234
```

IntegerTypes

```
33 33 3333 3333 3333
```

FloatandDoub1

```
e:
11.22334and
11.2233445566779
Decimal:
111.222333444555666777888999
```

Boolean:

```
Status:Tr
ue
```

Character:

```
SingleQuote
'
```

```
DoubleQuote
"
```

```
BackSlash\
```

```
A
```

```
Nownull:
```

```
Hi,I aman Object
```

```
-
```

```
1735802816System.Strin
g 34 System.Int32
False
```

```
S1 is:String1 ands2is String1  
S1 is:String1 ands2is NewString1
```

```
c:\NewFolder\Hello\P1.cs
```

```
c:\NewFolder\Hello\P1.cs We  
canalso write like this
```

```
FlowControl:(  
if) iis 25  
iis Non- zero
```

```
Typeina  
string:  
Pritesh  
The string had atleast5 but less than  
10characters ThestringwasPritesh
```

```
Switch:  
integerA= 2  
Goodmorning!
```

Practical 2

Aim:

Print given pattern.

```
@ @ @ @ @
```

```
@ @ @ @
```

```
@ @ @
```

```
@ @
```

```
@
```

```
using System;
```

```
namespace
```

```
Pattern
```

```
{
```

```
    class PatternExample
```

```
    {
```

```
        public static void Main()
```

```
        {
```

```
            int i,j;
```

```
            for (j = 5; j > 0; j--)
```

```
            {
```

```
                for (i = j; i > 0; i--)
```

```
                    Console.Write("
```

```
                    @ ");
```

```
                    Console.WriteLine
```

```
                    e());
```

```
            }
```

```
        }
```

```
    }
```

Output:

```
E:\Sem-  
6\VS\p2\p2>Pattern1.exe  
@@@@@  
@@@@@  
@@@@@  
@@@  
@@  
@
```

Aim:

Print given pattern.

```

1
1 2
1 2 3
1 2 3 4

```

```

using System;

namespace Pattern
{
    class patternExample
    {
        public static void Main()
        {
            int i, j;
            for (j = 1; j < 5; j++)
            {
                for (i = 1; i <= j; i++)
                {
                    Console.Write(i + " ");
                }
                Console.WriteLine();
            }
        }
    }
}

```

Output:

```

E:\Sem-
6\VS\p2\p2>Pattern2.exe 1
1
1 2
1 2 3
1 2 3 4

```

Aim:

Prompt a user to input his/her name and country name and print on console

```
using System;

public class userdata
{
    public static void Main()
    {
        string name, country; Console.Write("Enter Your Name :
"); name = Console.ReadLine(); Console.Write("Enter Your
Country: "); country = Console.ReadLine();
        Console.WriteLine("Hello " + name + " from country " +
country); Console.ReadKey();
    }
}
```

Output:

```
E:\Sem-
6\VS\p2\p2>Read.exe
Enteryourname:
Pritesh
EnteryourCity:
y: rajkot
HelloPriteshfrom cityRajkot
```


Aim:

Demonstrate inheritance to define Car class and derive Maruti and Mahindra

```
using System;

using
System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Inheritance
{
    class Program
    {
        class Car
        {
            protected String fuel, id, name;
        }

        class Maruti : Car
        {
            internal Maruti(String fuel, String id, String name)
            {
                this.fuel =
                fuel; this.id
                = id;
                this.name =
                name;

                Console.WriteLine("car id is {0} car name is {1} car fuel type {2}", this.id,
                this.name, this.fuel);
            }
        }
    }
}
```

```

    }

    class Mahindra : Car
    {
        internal Mahindra(String fuel, String id, String name)
        {
            this.fuel =
            fuel; this.id
            = id;

            this.name =
            name;

            Console.WriteLine("car id is {0} car name is {1} car fuel type {2}", this.id,
            this.name, this.fuel);
        }
    }

    static void Main(string[] args)
    {
        // Car car = new Car();

        Maruti maruti = new Maruti("petrol", "1", "maruti");

        Mahindra mahindra = new Mahindra("diesel", "2", "mahindra"); Console.ReadKey();
    }
}

```

Output:

```

E:\Sem-
6\VS\p2\p2>Inheritance.exe
This is maruti class
This is Mahindra class . . .

```

Practical 3

Aim:

Method overloading.

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

namespace MethodOverloading
{
    class Vector
    {
        public int x,
        y, z; public
        Vector()
        {
        }
        public Vector(int x, int y, int z)
        {
            this.x
            = x;
            this.y
            = y;
            this.z
            = z;
        }
    }
}
```

```

class Program
{
    public void add(int a, int b)
    {
        int c = a +
        b;

        Console.WriteLine
        e(c);
    }
    public void add(Vector a, Vector b)
    {
        Vector temp = new
        Vector(); temp.x =
        a.x + b.x;
        temp.y = a.y +
        b.y; temp.z =
        a.z + b.z;

        Console.WriteLine("{0} {1} {2}", temp.x, temp.y, temp.z);
    }
    public void add(int [,] x, int [,] y)
    {
        int[,] result = new int[3, 3];

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                result[i, j] = x[i, j] + y[i, j];
                Console.Write(result[i, j] + " ");
            }
        }
    }
}

```

```

    }

    Console.WriteLine();

}

static void Main(string[] args)
{
    Program p = new Program();

    p.add(10, 20);

    Vector a = new
    Vector(1, 2, 3);
    Vector b = new
    Vector(4, 5, 6);

    p.add(a, b);

    int[,] x= new int[3, 3];

    Console.Write("Enter first
    Matrix"); for (int i =
    0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            x[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    int[,] y = new int[3, 3];

```

```

        Console.WriteLine("Enter second
        Matrix"); for (int i = 0;
        i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                y[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        p.add(x, y); Console.ReadLine();
    }
}

```

Output:

```

E:\Sem-
6\VS\p2\p2>P3.1.exe
EnterNumber1:
1
EnterNumber2:
2
AdditionofNumber:3

EnterVector1
: 1
2
EnterVector2
: 3
1
Additionof vector: x=4, y=3

Additionof
twome trics:
Addition:6
Addition:8
Addition:
10
Addition:
12

```

Aim:

Constructor overloading

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

namespace constructoroverload
{
    class StudentData
    {
        String branch,
        name; int
        enrollment;

        public StudentData()
        {
        }
        public StudentData(String name)
        {
            this.name = name;
            Console.WriteLine("{0}",
            this.name);
        }
        public StudentData(String name, int enrollment)
        {
            this.name = name;
            this.enrollment =
            enrollment;
            Console.WriteLine("{0} {1}",this.name,this.enrollment );
        }
        public StudentData(String name, int enrollment, String branch)
        {
            this.name = name;
            this.enrollment =
            enrollment;
            this.branch=branch;
            Console.WriteLine("{0} {1} {2}", this.name,
            this.enrollment,this.branch);
        }
    }
}
```

```

class Overload
{
    static void Main(string[] args)
    {
        StudentData s1 = new
            StudentData("Pritesh"); StudentData s2
            = new StudentData("Pritesh",63);
        StudentData s3 = new
            StudentData("Pritesh", 63, "CE");
        Console.ReadLine();
    }
}

```

Output:

```

E:\Sem-
6\VS\p2\p2>P3.2.exe
FirstConstructor initi
ated..
SecondConstructor initia
ted.
-
ThirdConstructor initiated..

```


Practical 4

Aim:

Reflection Api

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

namespace p4
{
    class StudentData
    {
        String name,
        branch; String
        enrollment;
        public
        StudentData()
        {
        }
        public StudentData(String name)
        {
            this.name = name;
            Console.WriteLine("{0} ", this.name);
        }
        public StudentData(String name, String enrollment)
        {

```

```

        this.name = name;

        this.enrollment =
            enrollment;

        Console.WriteLine("{0} {1}", this.name, this.enrollment);
    }

    public StudentData(String name, String enrollment, String branch)
    {
        this.name = name;

        this.enrollment =
            enrollment;

        this.branch = branch;

        Console.WriteLine("{0} {1} {2}", this.name,
this.enrollment, this.branch);
    }

    public void print()
    {
        Console.WriteLine("{0} ", this.name);
    }
}

class Program
{
    static void Main(string[] args)
    {
        Type T =
            Type.GetType("p4.StudentData");

        Console.WriteLine("constructor");

        ConstructorInfo[] c =
            T.GetConstructors(); foreach
            (ConstructorInfo constructor in
                c)

```

```

    {
        Console.WriteLine(constructor.ToString());
    }

    Console.WriteLine
    ("Methods");

    MethodInfo[] m =
    T.GetMethods();

    foreach
    (MethodInfo method
    in m)
    {
        Console.WriteLine(method.ToString());
    }

    Console.ReadKey();
}
}
}

```

Output:

```

E:\Sem-
6\VS\p2\p2>Reflection.exe
System.Int32get_ID
System.Voidset_ID
System.Stringget_Name
System.Voidset_Name
System.VoidprintID
System.VoidprintName
System.StringToString
System.BooleanEquals
System.Int32GetHashCode
System.TypeGetType

Properties System.Int32ID

System.StringName Constructors
Void.ctor(Int32,System.String) Void.ctor()

```

Practical 5

Aim:

Copy data from one file to another using StreamReader and StreamWriter class.

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

namespace CopyFile2
{
    public class
        FileCopy
    {
        public void copyFile(String file1, String file2)
        {
            using (StreamReader reader = new
                StreamReader(file1)) using
                (StreamWriter writer = new
                StreamWriter(file2))
            {
                String line = null;
                while ((line = reader.ReadLine()) != null)
                {
                    writer.WriteLine(line);
                }
            }
        }
    }
}
```

```

    }

    }

}

class copyfile2
{
    static void Main(string[] args)
    {
        FileCopy fc = new FileCopy();

        String file1 = @"D:\Pritesh\DOTNET
PRACTICAL\DOTNET\file1.txt"; String file2 =
@"D:\Pritesh\DOTNET PRACTICAL\DOTNET\file1.txt";

        fc.copyFile(file1, file2);

    }

}

}

```

Output:

E:\Sem-6\VS\p2\p2>Reflection.exe

F1.txt: Hello vvp...

F2.txt: Hello vvp...

Aim:

Write a C# Program to Read Lines from a File until the End of File is Reached.

```
using System;

using System.Collections.Generic;

using System.Linq;

using
System.Text;

using
System.IO;

namespace CopyFile
{
    class CopyFile
    {
        static void Main(string[] args)
        {
            String file1 = @"D:\Pritesh\DOTNET
PRACTICAL\DOTNET\file1.txt"; String file2 =
@"D:\Pritesh\DOTNET PRACTICAL\DOTNET\file2.txt";
            using (StreamReader reader = new
StreamReader(file1))
            using (StreamWriter writer = new
                StreamWriter(file2))
                writer.Write(reader.ReadToEnd());
        }
    }
}
```

Output:

```
F1.txt:  
Hello World.....  
hii
```

```
how are you ???
```

```
F2.txt:  
Hello World.....  
hii
```

```
how are you ???
```

Aim:

List Files in a Directory

```
using System;

using System.Collections.Generic;

using System.Linq;

using
System.Text;

using
System.IO;

namespace CountFileDirectory
{
    class Program
    {
        {
            static void Main(string[] args)
            {
                string[] Directories =
Directory.GetDirectories(@"D:\Pritesh\DOTNET PRACTICAL\DOTNET");

                foreach (string dir in Directories)
                {
                    Console.WriteLine(dir);
                }

                string[] Files = Directory.GetFiles(@"D:\Pritesh\DOTNET
PRACTICAL\DOTNET"); foreach (string f in Files)
                {
                    Console.WriteLine(f);
                }

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


Output:

```
E:\Sem-6\VS\p2\p2>csc  
filecount.cs Filecount  
Constructoroverload  
Copyfile1  
Copyfile  
2  
DataEntr  
y DotNet  
Filecoun  
t  
Inheritance  
Demo  
Methodoverload  
Pattern1  
Pattern  
2  
File1.  
txt  
File2.  
txt
```

Practical 6

Aim:

Windows Form Application for Student Registration and store student Details in DataBase.

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;
using System.Data.SqlClient;
using System.IO;

namespace StudentRegistration
{
    public partial class Form1 : Form
    {
        String gender="";
        string imgPath,
        imgstudent;
        private object
```

```

radioButton1;

public Form1()
{
    InitializeComponent();
}

private void label1_Click(object sender, EventArgs e)
{
}

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

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    gender = "Male";
}

private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
}

private void btnSave_Click(object sender, EventArgs e)
{
    string source = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\  Documents\Dat
abase1.mdf;Integrated Security=True;Connect Timeout=30";

    string select = "select
count(*) from Student";

    SqlConnection conn = new
SqlConnection(source); SqlCommand cmd =
new SqlCommand(select, conn);

```

```

conn.Open();

int i =

Convert.ToInt16(cmd.ExecuteScalar());

int pkStudent = i + 1;

string insert = "insert into Student (pkStudent,
fname,lname,dob,imgstudent,gender,mobile,email) values ( "
+ pkStudent + ", '"
+ txtFname.Text + "', '" + txtLname.Text + "', '" +
dob.Value.Date + "', '" + (imgPath == null ? "" :
imgPath) + "', '" +
gender + "', '" + txtMobile.Text + "', '" +
txtEmail.Text + "')"; cmd = new
SqlCommand(insert, conn);

i = cmd.ExecuteNonQuery();

MessageBox.Show("You are

Done!!!");

InitializeComponent();
}

private void btnCancel_Click(object sender, EventArgs e)
{
    Environment.Exit(0);
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void rdoFemale_CheckedChanged(object sender, EventArgs e)
{
    gender = "Female";
}

private void btnImage_Click(object sender, EventArgs e)

```

```
{  
  
    openFileDialog1.Filter = "Jpg|*.jpg";  
  
    if (openFileDialog1.ShowDialog() == DialogResult.OK)  
    {  
  
        imgPath = @"D:\Pritesh\Pics\..\!\\" +  
        openFileDialog1.SafeFileName; pictureBox1.Image =  
        Image.FromFile(openFileDialog1.FileName);  
        //MessageBox.Show(imgPath);  
    }  
}  
}
```

Output:



The screenshot shows a web form for student registration. It includes input fields for 'First Name' and 'Last Name'. For 'Gender', there are radio buttons for 'Male' (selected) and 'Female'. For 'subject', there are checkboxes for 's1' and 's2'. There are two buttons: 'Save' (highlighted with a blue border) and 'Upload'. The 'Upload' button is positioned below a small image placeholder that shows a blurry picture of a person's face.

Practical 7

Aim:

Perform validation using Validation Controls

WebForm1.cs

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Practical7.WebForm1" %>

<!DOCTYPE html>

<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="Name"></asp:Label> &nbsp;<asp:TextBox ID="txtName"
runat="server"></asp:TextBox>

            <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="txtName" ErrorMessage="field must not be empty"
ForeColor="Red" ToolTip="Enter value">*</asp:RequiredFieldValidator>

            <br />

            <br />

            <asp:Label ID="Label2" runat="server" Text="Password"></asp:Label>

            &nbsp;<asp:TextBox ID="txtPwd" runat="server" TextMode="Password"
></asp:TextBox>

            <asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="txtCPwd" ControlToValidate="txtPwd"
ErrorMessage="Password & cpassword must be same" ForeColor="Red"
```

```

        ToolTip="Enter pasword">*</asp:CompareValidator>

        <br />

        <br />

        <asp:Label ID="Label3" runat="server" Text="C
Password"></asp:Label> &nbsp;<asp:TextBox ID="txtCPwd" runat="server"
TextMode="Password"></asp:TextBox>

        <br />

        <br />

        <asp:Label ID="Label4" runat="server" Text="Sem"></asp:Label>
&nbsp;<asp:TextBox ID="txtSem" runat="server"></asp:TextBox>

        <asp:RangeValidator ID="RangeValidator1" runat="server"
ControlToValidate="txtSem" ErrorMessage="Not valid sem"
ForeColor="Red" MaximumValue="8" MinimumValue="1" ToolTip="Enter sem"
Type="Integer">*</asp:RangeValidator>

        <asp:CustomValidator ID="CustomValidator1" runat="server"
ControlToValidate="txtSem" ErrorMessage="enter even semester"
ForeColor="Red" OnServerValidate="CustomValidator1_ServerValidate"
ToolTip="enter even semester">*</asp:CustomValidator>

        <br />

        <br />

        <asp:Label ID="Label6" runat="server" Text="Phone
no"></asp:Label> &nbsp;<asp:TextBox ID="txtPhone"
runat="server"></asp:TextBox>

        <asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server"
ControlToValidate="txtPhone" ErrorMessage="Invalid phone no"
ForeColor="Red" ToolTip="Enter phone" ValidationExpression="[0-
9]{10}">*</asp:RegularExpressionValidator>

        <br />

        <br />

        <asp:Label ID="Label5" runat="server"
Text="Email"></asp:Label> &nbsp;<asp:TextBox ID="txtEmail"

```

```
runat="server"></asp:TextBox>
```

```

        <asp:RegularExpressionValidator
ID="RegularExpressionValidator2" runat="server"
ControlToValidate="txtEmail" ErrorMessage="Invalid email"
ForeColor="Red" ToolTip="Enter email"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.\w+)*"></asp:RegularExpressionValidator>

        <br />

        <br />

        <br />

        <asp:Button ID="Button2" runat="server" Text="Submit" />

        <br />

        <asp:ValidationSummary ID="ValidationSummary1" runat="server" />

        <br />
    </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 Practical7

```



```

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

        protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args)
        {
            if (Convert.ToInt16(args.Value) % 2 == 0)
            {
                args.IsValid = true;
            }
            else
            {
                args.IsValid = false;
            }
        }
    }
}

```

Output:

localhost:49482/WebForm1.aspx x +

localhost:49482/WebForm1.aspx

Name

Password *

C Password

Sem

Phone no

Email

- Password & cpassword must be same

Practical 8

Aim:

Introduction to Master Pages

Site1.Master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs"
Inherits="WebApplication1.Site1" %>
```

```
<!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>
  <asp:ContentPlaceHolder ID="head"
runat="server"> </asp:ContentPlaceholder> <style
type="text/css">
    .style1 {
        width: 97px; height:
        141px;
    }
    .style2
    {
        width: 97px; height:
        105px;
    }
    .style3
    {
        width: 97px; height:
        99px;
    }
    .style4
    {
        width: 9px;
    }
  </style>
</head>
<body>
  <form id="form1" runat="server">
    <table height="50%" width="50%">
      <tr>
        <td class="style2" colspan="2">
          <asp:Label ID="lblheader" runat="server"
            Text="Header"></asp:Label> </td>
        </tr>
      <tr>
```

```

        <td class="style4">
            <asp:Button ID="btnsearch" runat="server" Text="search" />

            <asp:TextBox ID="txtsearch"
runat="server"></asp:TextBox> </td>
        <td class="style3">
            <asp:ContentPlaceholder
            ID="ContentPlaceholder1" runat="server">
                content page
            </asp:ContentPlaceholder>
        </td>
    </tr>
</tr>
    <td class="style1" colspan="2">
        <asp:Label ID="lblfooter" runat="server"
Text="Footer"></asp:Label> </td>
    </tr>
</table>
</form>
</body>
</html>

```

Site1.Master.cs:

```

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

namespace WebApplication1
{
    public partial class Site1
    : System.Web.UI.MasterPage {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        public Label LblHeader { get
        {
            return lblheader;
        }
        }
        public Button BtnSearch { get
        {
            return btnsearch;
        }
        }
        public TextBox TxtSearch { get
        {
            return txtsearch;
        }
        }
    }
}

```

```

    }
}
}

```

WebForm1.aspx:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication1.WebForm1" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
    runat="server"> <asp:TextBox ID="txtname" runat="server" ></asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Set Header"
onclick="Button1_Click" />
</asp:Content>

```

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 WebApplication1
{
    public partial class WebForm1 :
    System.Web.UI.Page {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            ((Site1)Master).LblHeader.Text = txtname.Text;
        }

    }
}

```

WebForm2.aspx:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="WebApplication1.WebForm2" %>
    <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
        runat="server"> <asp:GridView ID="grdstudent"
runat="server"> </asp:GridView>
</asp:Content>

```

WebForm2.aspx.cs:

```

using System;
using System.Collections.Generic; using
System.Linq;
using System.Web; using
System.Web.UI;
using System.Web.UI.WebControls; using
System.Data.SqlClient; namespace
WebApplication1

{
    public partial class WebForm2 :
        System.Web.UI.Page

    {
        protected void Page_Init(object sender, EventArgs e)
        {
            ((Site1)Master).BtnSearch.Click += new
                EventHandler(BtnSearch_Click);
        }

        void BtnSearch_Click(object sender, EventArgs e)
        {
            getData();
        }

        protected void Page_Load(object sender, EventArgs e)
        {}

        void getData() {
            string s= ((Site1)Master).TxtSearch.Text;
            Console.WriteLine(s);
            string source = @"Data Source=Pritesh\SQLExpress;Initial
                Catalog=DemoDb;Integrated Security=True;Pooling=False";
            string select = "select * from tblstudent where fname like '%" +
                ((Site1)Master).TxtSearch.Text + "%' ";
            SqlConnection con = new
                SqlConnection(source); SqlCommand cmd = new
                SqlCommand(select, con); con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            grdstudent.DataSource = rdr;
            grdstudent.DataBind();
            con.Close();
        }
    }
}

```

Output:

ABC

search	<input type="text"/>	ABC	Set Header
--------	----------------------	-----	------------

Footer

Header

search	<input type="text"/>
ABC	

pkstudent	fname	lname	gender	subject	imgStudent
18	ABC	gdag	m	s1 s2	IMG-20170326-WA0009.jpg
21	ABC	iggf	m	s1 s2	IMG-20170326-WA0009.jpg

Footer