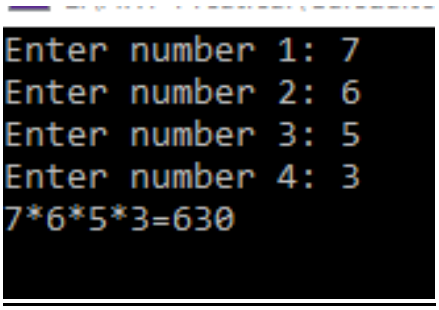
Practical No.: 1

AIM:- Working with basic C# and ASP .NET

A) Create an application that obtains four int values from the user and displays the product.

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

namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int num1, num2, num3, num4, prod;
         Console.Write("Enter number 1: ");
         num1 = Int32.Parse(Console.ReadLine());
         Console.Write("Enter number 2: ");
         num2 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter number 3: ");
         num3 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter number 4: ");
         num4 = Convert.ToInt32(Console.ReadLine());
         prod = num1 * num2 * num3 * num4;
         Console.WriteLine(num1 + "*" + num2 + "*" + num3 + "*" + num4 +
"=" + prod);
         Console.ReadKey();
      }
   }
}
```

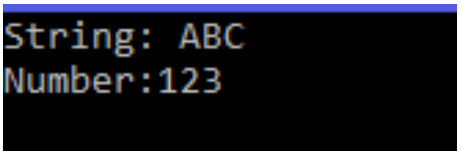
a.) <u>OUTPUT:-</u>


Enter number 1: 7
Enter number 2: 6
Enter number 3: 5
Enter number 4: 3
7*6*5*3=630

B) Create an application to demonstrate string operations.

```
using System;
namespace cmdLineArgs
{
class Program
{
static void Main(string[] args)
{
string str;
str=" ABC"
string n;
n="123"
Console.WriteLine("String:" + str);
Console.WriteLine("Number:" + n);
}
}
}
```

OUTPUT:

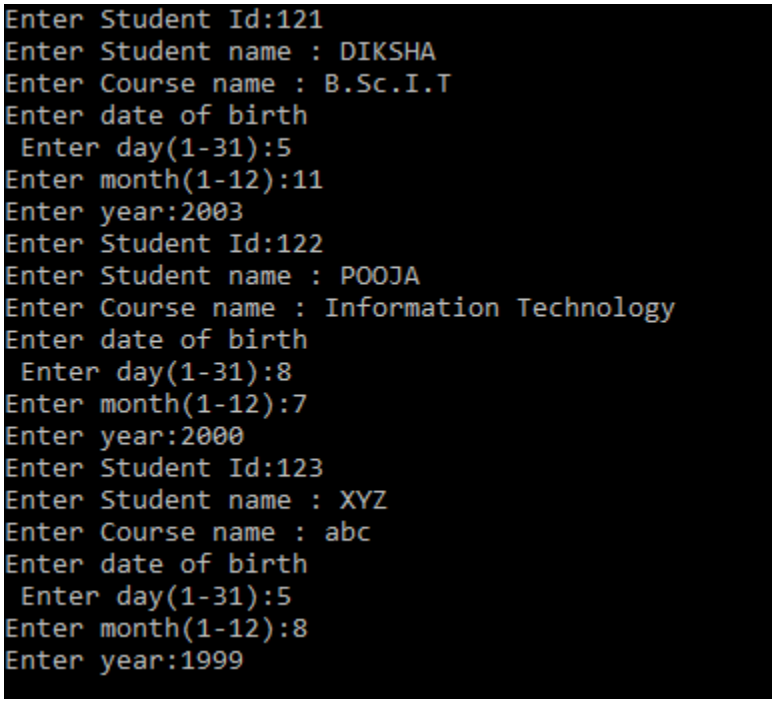


C) Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.

```
using System;
namespace ArrayOfStructs
{
class Program
{
struct Student
{
public string studid, name, cname;
public int day, month, year;
}
static void Main(string[] args)
{
Student[] s = new Student[3];
int i;
for (i = 0; i < 3; i++)
{
Console.Write("Enter Student Id:");
s[i].studid = Console.ReadLine();
Console.Write("Enter Student name : ");
s[i].name = Console.ReadLine();
Console.Write("Enter Course name : ");
s[i].cname = Console.ReadLine();
Console.Write("Enter date of birth\n Enter day(1-31):");
s[i].day = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter month(1-12):");
```

```
s[i].month = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter year:");
s[i].year = Convert.ToInt32(Console.ReadLine());
}
Console.WriteLine("\n\nStudent's List\n");
for (i = 0; i < 3; i++)
{
Console.WriteLine("\nStudent ID : " + s[i].studid);
Console.WriteLine("\nStudent name : " + s[i].name);
Console.WriteLine("\nCourse name : " + s[i].cname);
Console.WriteLine("\nDate of birth(dd-mm-yy) : " + s[i].day + "-" + s[i].month
+
"-" + s[i].year);
} } } }
```

OUTPUT:



```
Student's List

Student ID : 121

Student name : DIKSHA

Course name : B.Sc.I.T

Date of birth(dd-mm-yy) : 5-11-2003

Student ID : 122

Student name : POOJA

Course name : Information Technology

Date of birth(dd-mm-yy) : 8-7-2000

Student ID : 123

Student name : XYZ

Course name : abc

Date of birth(dd-mm-yy) : 5-8-1999
```

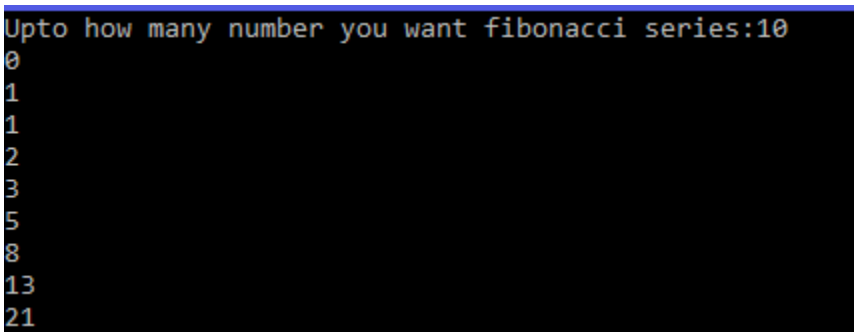
D) Create an application to demonstrate following operations

[i] Fibonacci Series

```
using System;
namespace ConsoleApplication3
{
class Program
{
static void Main(string[] args)
{
int num1=0,num2=1,num3,num4,num,counter;
Console.Write ("Upto how many number you want fibonacci series:");
num=int.Parse(Console.ReadLine());
counter=3;
Console.Write(num1+"\t"+num2);
while(counter<=num)
{
num3 = num1 + num2;
if (counter >= num)

break;
Console.Write("\t" + num3);
num1 = num2;
num2 = num3;
counter++;
}
```

```
}

}

}
```

OUTPUT:



[ii] Test for prime numbers.

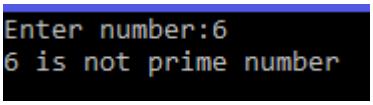
CODE:

```
using System;

namespace testprime

{

class Program

{

static void Main(string[] args)

{

int num, counter;

Console.Write("Enter number:");

num = int.Parse(Console.ReadLine());

for (counter = 2; counter <= num / 2; counter++)
```

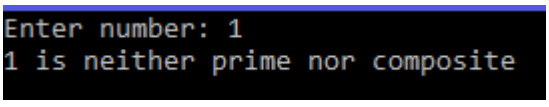
```
{
if ((num % counter) == 0)
break;
}
if (num == 1)
Console.WriteLine(num + "is neither prime nor composite");
else if(counter<(num/2))
Console.WriteLine(num+"is not prime number");
else
Console.WriteLine(num+"is prime number");
}
}
}
```

OUTPUT:

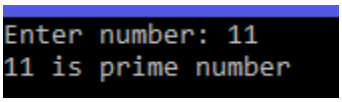
(1st attempt)



(2nd)



(3rd)



[iii] Test for vowels.

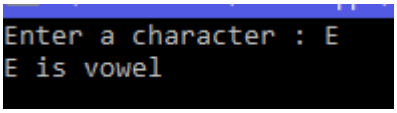
CODE:

```
using System;
namespace vowels
{
class Program
{
static void Main(string[] args)
{
char ch;
Console.Write("Enter a character : ");
ch = (char)Console.Read();
switch (ch)
{
case 'a':
case 'A':
case 'e':
case 'E':
case 'i':
case 'I':
case 'o':
case 'O':
case 'u':
case 'U':
```

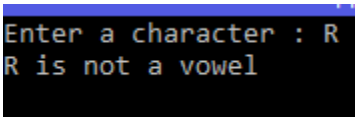
```
Console.WriteLine(ch + "is vowel");
break;
default:
Console.Write(ch + "is not a vowel");
break;
}
Console.ReadKey();
}
}
}
```

OUTPUT:

(1st attempt)



(2nd attempt)



[iv]Use of foreach loop with arrays.

CODE:

```
using System;
class ExampleForEach
{
public static void Main()
{
string[] str = { "Shield", "Evaluation", "DX" };
foreach (String s in str)
{
Console.WriteLine(s);
}
}
```

OUTPUT:

```
Shield
Evaluation
DX
```

[v] Reverse a number and find sum of digits of a number.

CODE:

```
using System;
namespace reverseNumber
{
class Program
{
static void Main(string[] args)
{
int num,actualnumber,revnum=0,digit,sumDigits=0;
Console.Write("Enter number:");
num = int.Parse(Console.ReadLine());
actualnumber = num;
while (num > 0)
{
digit = num % 10;
revnum = revnum * 10 + digit;
sumDigits=sumDigits+digit;
num = num / 10;
}
Console.WriteLine("Reverse of " + actualnumber + "=" + revnum);
Console.WriteLine("Sum of its digits:" + sumDigits);}}}
```

OUTPUT:

```
Enter number:458
Reverse of 458 = 854
Sum of its digits: 17
```

Practical No.: 2

AIM: Working with Object Oriented C# and ASP .NET

A) Create simple application to perform following operations.

[i] Finding Factorial Value

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

namespace factorial
{
   class Program
   {
      static void Main(string[] args)
      {
         int i, number, fact;
         Console.WriteLine("Enter the Number");
         number = int.Parse(Console.ReadLine());
         fact = number;
         for (i = number - 1; i >= 1; i--)
         {
            fact = fact * i;
         }
         Console.WriteLine("\nFactorial of Given Number is: "+fact);
         Console.ReadLine();

      }
   }
}
```

Output:

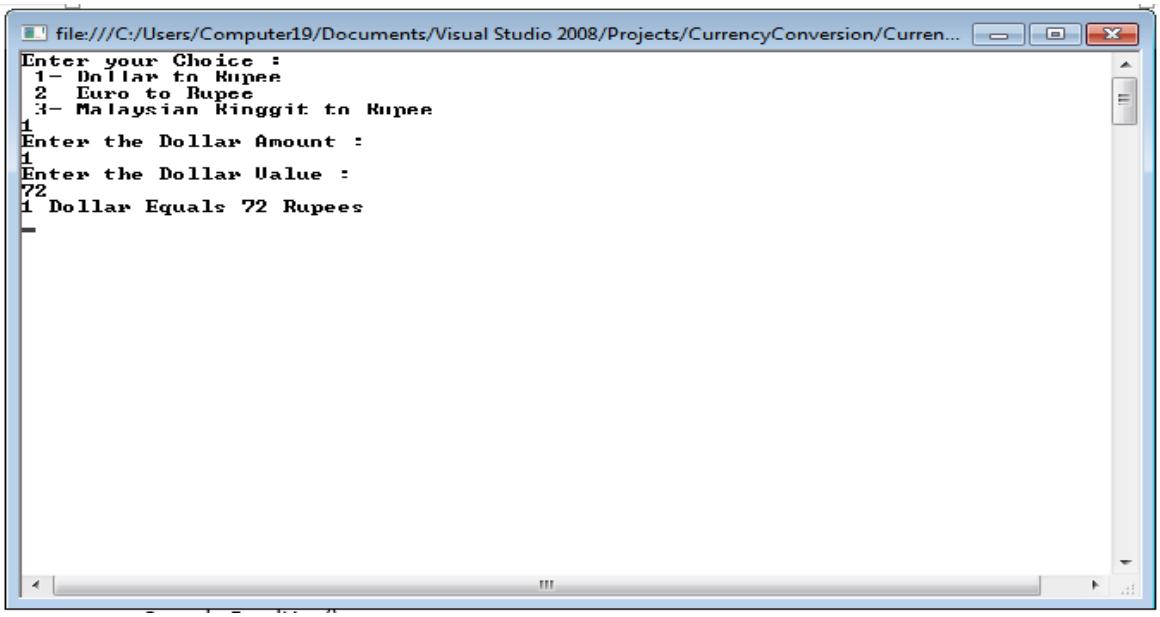
```
Enter the Number
5

Factorial of Given Number is: 120
```

[ii] Money Conversion

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

namespace CurrencyConversion
{
class Program
{
static void Main(string[] args)
{
int choice;
Console.WriteLine(&amp;quot;Enter your Choice :\n 1- Dollar to Rupee \n 2-
Euro to Rupee \n 3- Malaysian
Ringgit to Rupee &amp;quot;);
choice = int.Parse(Console.ReadLine());
switch (choice)
{
case 1:
Double dollar, rupee, val;
Console.WriteLine(&amp;quot;Enter the Dollar Amount :&amp;quot;);
dollar = Double.Parse(Console.ReadLine());
Console.WriteLine(&amp;quot;Enter the Dollar Value :&amp;quot;);
val = double.Parse(Console.ReadLine());
rupee = dollar * val;

Console.WriteLine(&amp;quot;{0} Dollar Equals {1} Rupees&amp;quot;, dollar,
rupee);
break;
case 2:
Double Euro, rupe, valu;
Console.WriteLine(&amp;quot;Enter the Euro Amount :&amp;quot;);
Euro = Double.Parse(Console.ReadLine());
```

```
Console.WriteLine(&amp;quot;Enter the Euro Value :&amp;quot;);
valu = double.Parse(Console.ReadLine());
rupe = Euro * valu;
Console.WriteLine(&amp;quot;{0} Euro Equals {1} Rupees&amp;quot;, Euro, rupe);
break;
case 3:
Double ringit, rup, value;
Console.WriteLine(&amp;quot;Enter the Ringgit Amount :&amp;quot;);
ringit = Double.Parse(Console.ReadLine());
Console.WriteLine(&amp;quot;Enter the Ringgit Value :&amp;quot;);
value = double.Parse(Console.ReadLine());
rup = ringit * value;
Console.WriteLine(&amp;quot;{0} Malaysian Ringgit Equals {1} Rupees&amp;quot;,
ringit, rup);
break;
}
Console.ReadLine();
}
}
}
```

2)A)[ii]



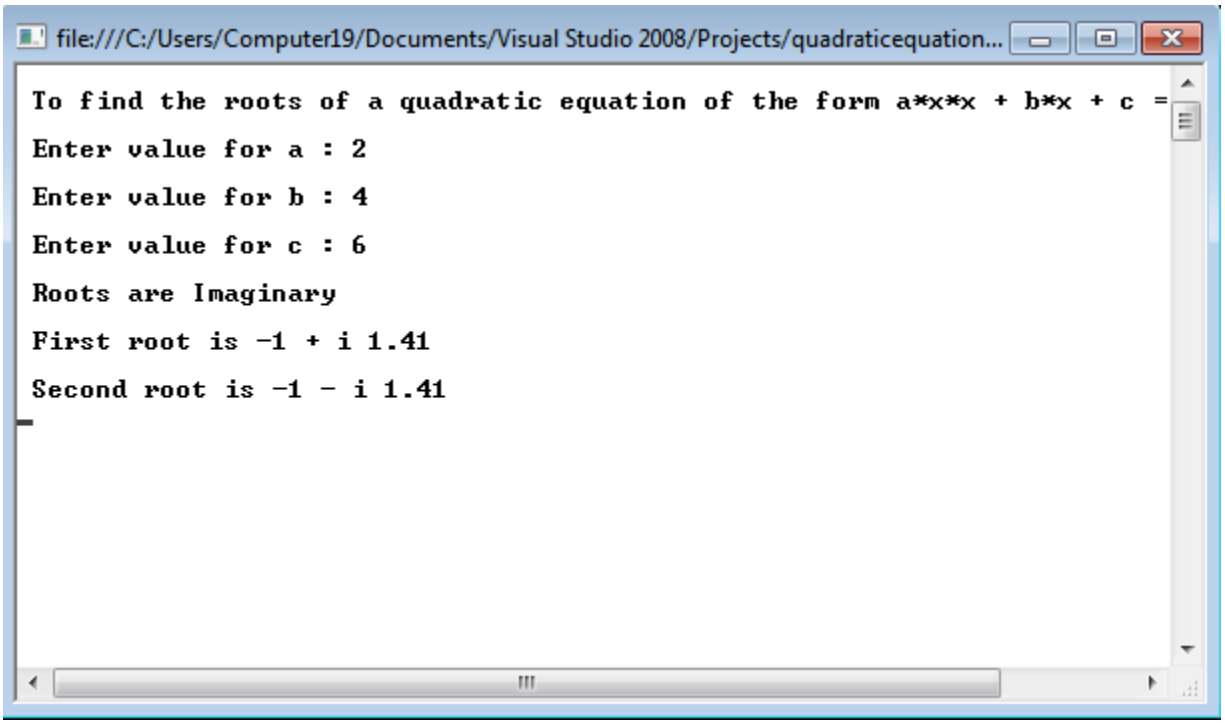
[iii] Quadratic Equation

```
using System;

namespace example
{
   class Quadraticroots
   {
      double a, b, c;

      public void read()
      {
         Console.WriteLine(" \n To find the roots of a quadratic equation of
the form a*x*x + b*x + c = 0");
         Console.Write("\n Enter value for a : ");
         a = double.Parse(Console.ReadLine());
         Console.Write("\n Enter value for b : ");
         b = double.Parse(Console.ReadLine());
         Console.Write("\n Enter value for c : ");
         c = double.Parse(Console.ReadLine());
      }
      public void compute()
      {
         int m;
         double r1, r2, d1;
         d1 = b * b - 4 * a * c;
         if (a == 0)
            m = 1;
         else if (d1 > 0)
            m = 2;
         else if (d1 == 0)
            m = 3;
         else
            m = 4;
         switch (m)
```

```
        {
            case 1: Console.WriteLine("\n Not a Quadratic equation, Linear
equation");
                Console.ReadLine();
                break;
            case 2: Console.WriteLine("\n Roots are Real and Distinct");
                r1 = (-b + Math.Sqrt(d1)) / (2 * a);
                r2 = (-b - Math.Sqrt(d1)) / (2 * a);
                Console.WriteLine("\n First root is {0:#.##}", r1);
                Console.WriteLine("\n Second root is {0:#.##}", r2);
                Console.ReadLine();
                break;
            case 3: Console.WriteLine("\n Roots are Real and Equal");
                r1 = r2 = (-b) / (2 * a);
                Console.WriteLine("\n First root is {0:#.##}", r1);
                Console.WriteLine("\n Second root is {0:#.##}", r2);
                Console.ReadLine();
                break;
            case 4: Console.WriteLine("\n Roots are Imaginary");
                r1 = (-b) / (2 * a);
                r2 = Math.Sqrt(-d1) / (2 * a);
                Console.WriteLine("\n First root is {0:#.##} + i {1:#.##}", r1, r2);
                Console.WriteLine("\n Second root is {0:#.##} - i {1:#.##}", r1,
r2);
                Console.ReadLine();
                break;
        }
    }
  }

  class Roots
  {
    public static void Main()
    {
      Quadraticroots qr = new Quadraticroots();
      qr.read();
```

```
            qr.compute();
        }
    }
}
```

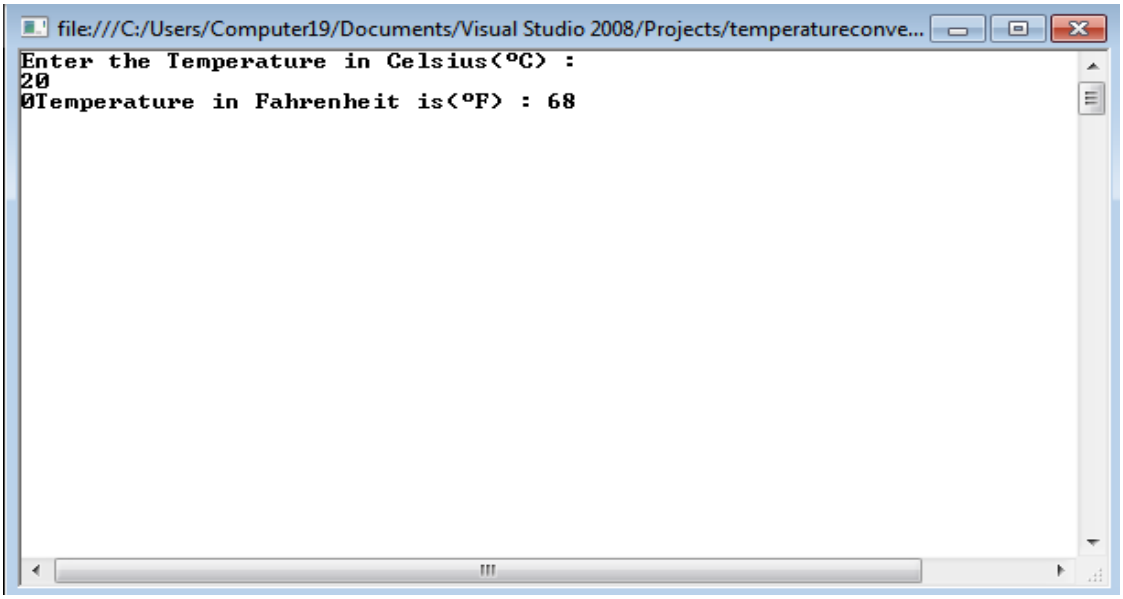
2)A)[iii]



[iv] Temperature Conversion

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

namespace temperatureconversion
{
   class Program
   {
      static void Main(string[] args)
      {
         int celsius, faren;
         Console.WriteLine("Enter the Temperature in Celsius(°C) : ");
         celsius = int.Parse(Console.ReadLine());
         faren = (celsius * 9) / 5 + 32;
         Console.WriteLine("0Temperature in Fahrenheit is(°F) : " + faren);
         Console.ReadLine();
      }
   }
}
```

2)A)[iv]



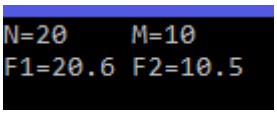
B) Create simple application to demonstrate use of following concepts.

[i] Function Overloading

```
using System;
namespace swap
{
class Overloading
{
public void swap(ref int n, ref int m)
{
int t;
t = n;
n = m;
m = t;
}
public void swap(ref float f1, ref float f2)
{
float f;
f = f1;
f1 = f2;
f2 = f;
}
}
class program
{
static void Main(string[] args)
{
Overloading objOverloading = new Overloading();
int n = 10, m = 20;
objOverloading.swap(ref n, ref m);
Console.WriteLine("N=" + n + "\tM=" + m);
float f1 = 10.5f, f2 = 20.6f;
objOverloading.swap(ref f1, ref f2);
Console.WriteLine("F1=" + f1 + "\tF2=" + f2);
```

```
}
 }
}
```

OUTPUT:



[ii] Inheritance

(a) Single Inheritance

Write a program to implement single inheritance from following figure. Accept and
display data for one table.

CODE:

Furniture.cs

```
using System;
namespace SingleInheritance
{
class Furniture
{
string material;
float price;
public void getdata()
{
Console.Write("Enter material : ");
material = Console.ReadLine();
Console.Write("Enter price : ");
price = float.Parse(Console.ReadLine());
}
public void showdata()
{
Console.WriteLine("Material : " + material);
Console.WriteLine("Price : " + price);
} } }
```

Table.cs

```
using System;
namespace SingleInheritance
{
class Table:Furniture
{
int height, surface_area;
public void getdata()
{
```

```
base.getdata();
Console.Write("Enter height: ");
height = int.Parse(Console.ReadLine());
Console.Write("Enter surface area: ");
surface_area = int.Parse(Console.ReadLine());
}
public void showdata()
{
base.showdata();
Console.WriteLine("Height : " + height);
Console.WriteLine("Surface Area : " + surface_area);
} } }
```

Program.cs

```
using System;
namespace SingleInheritance
{
class Program
{
static void Main(string[] args)
{
Table t1 = new Table();
t1.getdata();
t1.showdata();
} } }
```

OUTPUT:

```
Enter material : wood
Enter price : 1220
Enter height: 35
Enter surface area: 26
Material : wood
Price : 1220
Height : 35
Surface Area : 26
```

[ii](b) **Multiple inheritance**

CODE:

Gross.cs

```
using System;
namespace MultipleInheritance
{
interface Gross
{
int ta
{
get;
set;
}
int da
{
get;
set;
}
int GrossSal();
} }
```

Employee.cs

```
using System;
namespace MultipleInheritance
{
class Employee
{
string name;
public Employee(string name)
{ this.name = name; }
public int BasicSal(int basicSal)
{ return basicSal; }
public void ShowData()
{
Console.WriteLine("Name : " + name);
} } }
```

Salary.cs

```
using System;
namespace MultipleInheritance
{
class Salary:employee,Gross
{
int hra;
public Salary(string name, int hra):base(name)
{ this.hra = hra; }
public int ta
{
get {return S_ta; }
set { S_ta = value; }
}
private int S_ta;
public int da
{
get { return S_da; }
set { S_da = value; }
}
private int S_da;
public int GrossSal()
{
int gSal;
gSal = hra + ta + da + BasicSal(15000);
return gSal;
}
public void dispSal()
{ base.ShowData();
Console.WriteLine("Gross Sal : " + GrossSal());
} } }
```

Program.cs

```
using System;
namespace MultipleInheritance
{
class Program
```

```
{
static void Main(string[] args)
{
Salary s = new Salary("Prachit", 35000);
s.da = 20000;
s.ta = 30000;
s.dispSal();
} } }
```

OUTPUT:

```
Name :Prachit
Gross Sal :100000
```

(ii)[c] **Heirarchical Inheritance**

CODE:

Employee.cs

```
using System;
namespace HeirarchicalInheritance
{
class employee
{
public virtual void display()
{
Console.WriteLine("Display of employee class called ");
} } }
```

Programmer.cs

```
using System;
namespace HeirarchicalInheritance
{
class Programmer:employee
{
```

```
public void display()
{
Console.WriteLine(" Display of Programmer class called ");
} } }
```

Manager.cs

```
using System;
namespace HeirarchicalInheritance
{
class Manager
{
public void display()
{
Console.WriteLine("Display of manager class called ");
} } }
```

Program.cs

```
using System;
namespace HeirarchicalInheritance
{
class Program
{
static void Main(string[] args)
{
Programmer objProgrammer;
Manager objManager;
Console.Write("Whose details you want to use to see \n
1.Programmer \n
2.Manager");
int choice=int.Parse(Console.ReadLine());
if(choice==1)
{
objProgrammer=new Programmer();
objProgrammer.display();
}
else if(choice==2)
{
```

```
objManager=new Manager();
objManager.display();
}
else
{
Console.WriteLine("Wrong choice entered");
} } } }
```

OUTPUT:

```
Whose details you want to use to see
1.Programmer
2.Manager1
Display of Programmer class called
Whose details you want to use to see
1.Programmer
2.Manager2
Display of manager class called
Whose details you want to use to see
1.Programmer
2.Manager6
Wrong choice entered
```

(ii)[d] **Multilevel Inheritance**

Result.cs

```
using System;
namespace multilevelinheritance
{
class Result:Test
{
int total;
public Result(int roll_no, string name, int marks1, int marks2)
: base(roll_no, name, marks1, marks2)
{
total = getMarks1() + getMarks2();
}
public void display()
{
base.display();
Console.WriteLine("Total: " + total);
} } }
```

Test.cs

```
using System;
namespace multilevelinheritance
{
class Test:student
{
int marks1, marks2;
public Test(int roll_no, string name, int marks1, int marks2)
: base(roll_no, name)
{
this.marks1 = marks1;
this.marks2 = marks2;
}
```

```
public int getMarks1()
{
return marks1;
}
public int getMarks2()
{
return marks2;
}
public void dispaly()
{
base.display();
Console.WriteLine("Marks1: " + marks1);
Console.WriteLine("Marks2: " + marks2);
} } }
```

Student.cs

```
using System;
namespace multilevelinheritance
{
class student
{
int roll_no;
string name;
public student(int roll_no, string name)
{
this.roll_no = roll_no;
this.name = name;
}
public student() { }
public void display()
{
Console.WriteLine("Roll no: " + roll_no);
Console.WriteLine("Name: " + name);
} } }
```

Program.cs

```
using System;
namespace multilevelinheritance
{
class Program
{
static void Main(string[] args)
{
Result r1 = new Result(101, "Prachit", 50, 70);
r1.display();
} } }
```

OUTPUT:

```
Roll no: 101
Name: Prachit
Marks1: 50
Marks2: 70
Total: 120
```

[iii] **Constructor Overloading**

Salary.cs

```
using System;
namespace SalaryConstructure
{
class Salary
{
int basic, ta, da, hra;
public Salary()
{
da = 9000;
```

```
hra = 6000;
}
public void getdata()
{
Console.Write("Enter basic salary : ");
basic = int.Parse(Console.ReadLine());
Console.Write("Enter travelling allowance : ");
ta = int.Parse(Console.ReadLine());
}
public void showdata()
{
Console.WriteLine("Basic salary : " + basic);
Console.WriteLine("Dearness allowence : " + da);
Console.WriteLine("Housing rent allowence : " + hra);
Console.WriteLine("Travelling allowence : " + ta);
Console.WriteLine("Gross Salary : " + (basic + da + hra + ta));
} } }
```

Program.cs

```
using System;
namespace SalaryConstructure
{
class Program
{
static void Main(string[] args)
{
Salary s = new Salary();
s.getdata();
s.showdata();
} } }
```

OUTPUT:

Enter basic salary : 52000
Enter travelling allowance : 3000
Basic salary : 52000
Dearness allowence : 9000
Housing rent allowence : 6000
Travelling allowence : 3000
Gross Salary : 70000

(iv) **Interfaces**

ODDEVEN.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace InterFaceDemo {

   interface IOne {

      void ONE(); //Pure Abstract Method Signature

   }

   interface ITwo {

      void TWO();

   }

   interface IThree: IOne {

      void THREE();

   }
```

```
interface IFour {
    void FOUR();
}
interface IFive: IThree {
    void FIVE();
}
interface IEVEN: ITwo, IFour {}
class ODDEVEN: IEVEN, IFive //Must Implement all the abstract method, in Derived class.
{
    public void ONE() //Implementation of Abstract Method.
    {
        Console.WriteLine("This is ONE");
    }
    public void TWO() {
        Console.WriteLine("This is TWO");
    }
    public void THREE() {
        Console.WriteLine("This is THERE");
    }
    public void FOUR() {
        Console.WriteLine("This is FOUR");
```

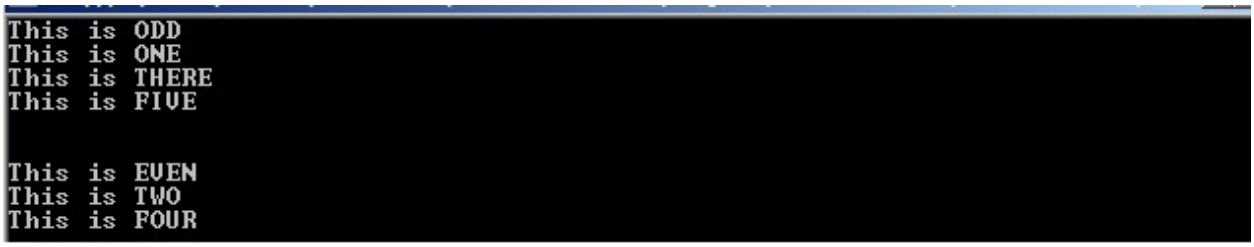
```
        }
        public void FIVE() {
            Console.WriteLine("This is FIVE");
        }
    }
}
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace InterFaceDemo {
    class Program {
        static void Main(string[] args) {
            Console.WriteLine("This is ODD");
            IFive obj1 = new ODDEVEN();
            obj1.ONE();
            obj1.THREE();
            obj1.FIVE();
            Console.WriteLine("\n\nThis is EVEN");
            IEVEN obj2 = new ODDEVEN();
            obj2.TWO();
```

```
            obj2.FOUR();
            Console.ReadLine();
        }
    }
}
```

(iv) output- interface



(C) Create simple application to demonstrate use of following concepts

[i] Using Delegates and events

TrafficSignal.cs

```
using System;
namespace TrafficDelegateExample
{
public delegate void TrafficDel();
class TrafficSignal
{
public static void Yellow()
{
Console.WriteLine("Yellow light signals to get ready");
}
public static void Green()
{
Console.WriteLine("Green light signals to go");
}
public static void Red()
{
Console.WriteLine("Red light signals to stop");
}
TrafficDel[] td = new TrafficDel[3];
```

```
public void IdentifySignal()
{
td[0] = new TrafficDel(Yellow);
td[1] = new TrafficDel(Green);
td[2] = new TrafficDel(Red);
}
public void display()
{
td[0]();
td[1]();
td[2]();
}
} }
```

Program.cs

```
using System;
namespace TrafficDelegateExample
{
class Program
{
static void Main(string[] args)
{
```

```
TrafficSignal ts = new TrafficSignal();
ts.IdentifySignal();
ts.display();
} } }
```

OUTPUT:

```
Yellow light signals to get ready
Green light signals to go
Red light signals to stop
```

[ii] Exception handling

NotEvenException.cs

```
using System;
namespace ExceptionHandlingExample
{
class NotEvenException:Exception
{
public NotEvenException(string msg)
: base(msg)
{
}
} }
```

Program.cs

```
using System;
```

```
namespace ExceptionHandlingExample
{
class Program
{
static void Main(string[] args)
{
int num;
try
{
Console.Write("Enter a number: ");
num = int.Parse(Console.ReadLine());
if ((num % 2) != 0) throw new NotEvenException("Not an even number ");
else
Console.WriteLine("Its even number ");
}
catch (NotEvenException e) { Console.WriteLine(e.Message); }
} } }
```

OUTPUT:

```
Enter a number: 5
Not an even number
----------------------------------------------------------
```

Practical No.: 3

AIM:- Working with Web Forms and Controls.

A)Demonstrate the use of Calendar control to perform followingoperations.

a) Display messages in a calendar control
b) Display vacation in a calendarcontrol
c) Selected day in a calendar control using style
d) Difference between two calendardates

Calender properties set for this example:
aspx code:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="calendar.aspx.cs"
Inherits="YourNamespace.CalendarExample" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Calendar Example</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>

         <!-- Selected day in a calendar control using style -->
         <asp:Calendar ID="calendarSelectedDay" runat="server"
OnSelectionChanged="calendarSelectedDay_SelectionChanged"
OnDayRender="calendarVacation_DayRender"
OnayRender="calendarMessages_DayRender"></asp:Calendar>

         <!-- Calculate difference between two calendar dates -->
```

```
        <asp:TextBox ID="txtDate1" runat="server" placeholder="Date 1
(yyyy-MM-dd)"></asp:TextBox>
        <asp:TextBox ID="txtDate2" runat="server" placeholder="Date 2
(yyyy-MM-dd)"></asp:TextBox>
        <asp:Button ID="btnCalculateDifference" runat="server"
Text="Calculate Difference" OnClick="btnCalculateDifference_Click" />
        <asp:Label ID="lblDateDifference" runat="server"
Text=""></asp:Label>
      </div>
   </form>
</body>
</html>
```

aspx.cs code:

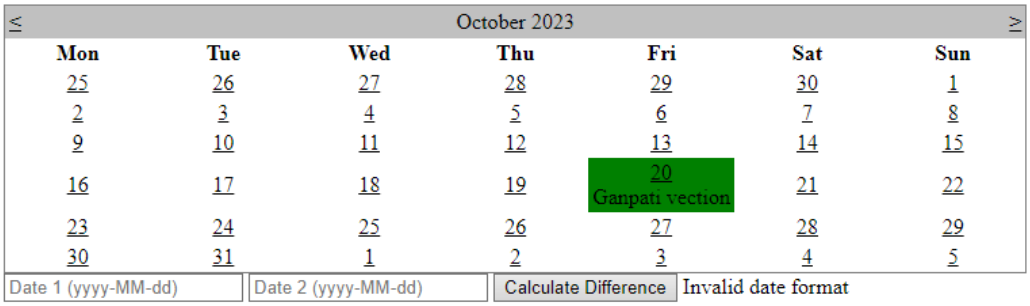
```
using System;
using System.Collections.Generic;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace YourNamespace
{
   public partial class CalendarExample : Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            // Add messages for specific dates
            DateTime dateWithMessage1 = new DateTime(2023, 10, 5);
            DateTime dateWithMessage2 = new DateTime(2023, 10, 10);

            List<DateTime> messageDates = new List<DateTime> {
dateWithMessage1, dateWithMessage2 };
            Session["MessageDates"] = messageDates;
```

```
            // Add vacation dates
            DateTime vacationDate1 = new DateTime(2023, 09, 15);
            DateTime vacationDate2 = new DateTime(2023, 10, 20);

            List<DateTime> vacationDates = new List<DateTime> {
vacationDate1, vacationDate2 };
            Session["VacationDates"] = vacationDates;
        }
    }

    protected void calendarMessages_DayRender(object sender,
DayRenderEventArgs e)
    {
        List<DateTime> messageDates =
(List<DateTime>)Session["MessageDates"];

        if (messageDates != null)
        {
            foreach (DateTime date in messageDates)
            {
                if (e.Day.Date == date)
                {
                    e.Cell.BackColor = System.Drawing.Color.Yellow;
                    e.Cell.Controls.Add(new LiteralControl("<br />messege
here"));
                }
            }
        }
    }

    protected void calendarVacation_DayRender(object sender,
DayRenderEventArgs e)
    {
        List<DateTime> vacationDates =
(List<DateTime>)Session["VacationDates"];
```

```
        if (vacationDates != null)
        {
            foreach (DateTime date in vacationDates)
            {
                if (e.Day.Date == date)
                {
                    e.Cell.BackColor = System.Drawing.Color.Green;
                    e.Cell.Controls.Add(new LiteralControl("<br />Ganpati
vection"));
                }
            }
        }
    }

    protected void calendarSelectedDay_SelectionChanged(object sender,
EventArgs e)
    {
        calendarSelectedDay.SelectedDayStyle.BackColor =
System.Drawing.Color.Red;
    }

    protected void btnCalculateDifference_Click(object sender, EventArgs
e)
    {
        if (DateTime.TryParse(txtDate1.Text, out DateTime date1) &&
DateTime.TryParse(txtDate2.Text, out DateTime date2))
        {
            TimeSpan difference = date1 - date2;
            int daysDifference = difference.Days;
            lblDateDifference.Text = $"Days difference: {daysDifference}";
        }
        else
        {
            lblDateDifference.Text = "Invalid date format";
        }
    }
```

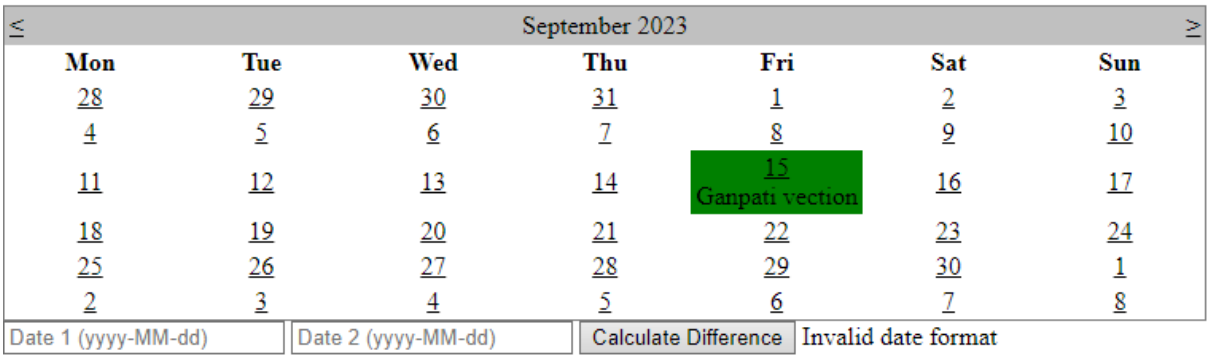
```
    }
}
```

OUTPUT:-



<	October 2023					>
Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20 Ganpati vection	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Date 1 (yyyy-MM-dd) Date 2 (yyyy-MM-dd) Calculate Difference Invalid date format



<	September 2023					>
Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15 Ganpati vection	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Date 1 (yyyy-MM-dd) Date 2 (yyyy-MM-dd) Calculate Difference Invalid date format

B) Demonstrate the use of Treeview control perform following operations.

a) Treeview control and datalist b) Treeview operations

Add XML File

Website -> Add -> XML File and Name it ‘stdetail’.

stdetail.xml

XML CODE

NAME YOUR FILE AS "stdetail.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<studentdetail xmlns="http://example.com/studentdetail">
    <student>
        <sid>1</sid>
        <sname>Tushar</sname>
        <sclass>TYIT</sclass>
    </student>
    <student>
        <sid>2</sid>
        <sname>Sonali</sname>
        <sclass>TYCS</sclass>
    </student>
    <student>
        <sid>3</sid>
        <sname>Yashashree</sname>
        <sclass>TYIT</sclass>
    </student>
    <student>
        <sid>4</sid>
        <sname>Vedshree</sname>
        <sclass>TYCS</sclass>
    </student>
</studentdetail>
```

ASPX code

NAME YOUR FILE AS "Treeview.aspx"

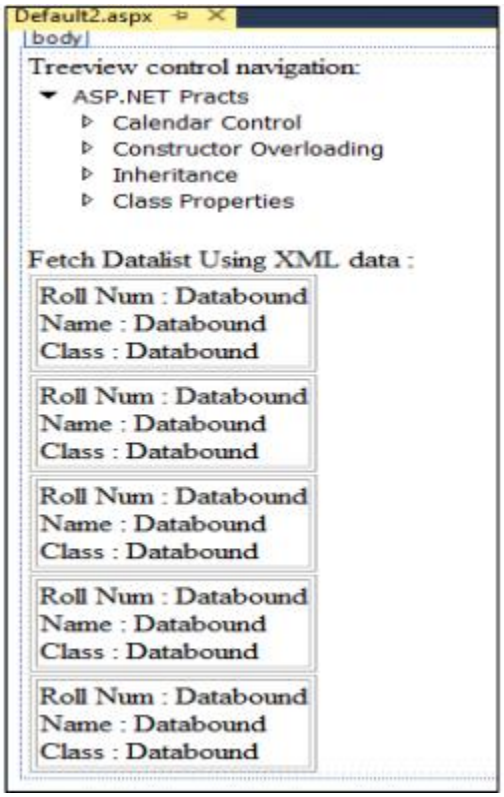
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Treeview.aspx.cs" Inherits="exp2b.Treeview" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
  <form id="form1" runat="server">
 <div>
   Treeview control navigation:
   <asp:TreeView ID="TreeView1" runat="server" Width="150px"
ImageSet="Arrows">
    <HoverNodeStyle Font-Underline="True" ForeColor="#5555DD" />
    <Nodes>
      <asp:TreeNode Text="ASP.NET Practice" Value="PracticeNode">
        <asp:TreeNode Text="Calendar Control" Value="RED"
NavigateUrl="~/calndrCtrl.aspx"></asp:TreeNode>
        <asp:TreeNode Text="Constructor Overloading" Value="GREEN"
NavigateUrl="~/clsconstrc.aspx"></asp:TreeNode>
        <asp:TreeNode Text="Inheritance" Value="BLUE"
NavigateUrl="~/singleInh.aspx"></asp:TreeNode>
        <asp:TreeNode Text="Class Properties" Value="ClassProperties"
NavigateUrl="~/clsProp.aspx"></asp:TreeNode>
      </asp:TreeNode>
    </Nodes>
    <NodeStyle Font-Names="Tahoma" Font-Size="10pt" ForeColor="Black"
HorizontalPadding="5px" NodeSpacing="0px" VerticalPadding="0px" />
    <ParentNodeStyle Font-Bold="False" />
    <SelectedNodeStyle Font-Underline="True" ForeColor="#5555DD"
HorizontalPadding="0px" VerticalPadding="0px" />
   </asp:TreeView>
   <br />
```

```
   Fetch DataList Using XML data:
  </div>
  <asp:DataList ID="DataList1" runat="server">
   <ItemTemplate>
    <table class="table" border="1">
     <tr>
      <td>Roll Num: <%# Eval("sid") %><br />
       Name: <%# Eval("sname") %><br />
       Class: <%# Eval("sclass") %>
      </td>
     </tr>
    </table>
   </ItemTemplate>
  </asp:DataList>
</form>

</body>
</html>
```

ASPX.cs CODE

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

namespace exp2b
{
   public partial class Treeview : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
```

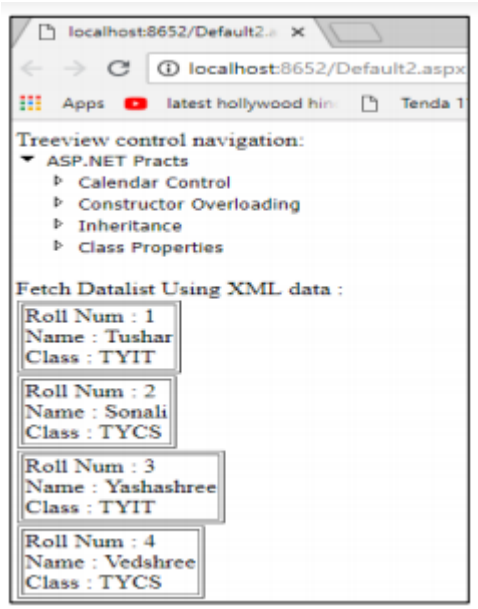
```
            BindData();
        }
    }
    protected void BindData()
    {
        DataSet ds = new DataSet();
        ds.ReadXml(Server.MapPath("stdetail.xml"));
        if (ds != null && ds.HasChanges())
        {
            DataList1.DataSource = ds;
            DataList1.DataBind();
        }
        else
        {
            DataList1.DataBind();
        }
    }
  }
}
```



Default2.aspx.cs

```
using System.Data;
public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
if (!IsPostBack)
{
BindData();
}
}
protected void BindData()
{
DataSet ds = new DataSet();
ds.ReadXml(Server.MapPath("stdetail.xml"));
if (ds != null && ds.HasChanges())
{
DataList1.DataSource = ds;
DataList1.DataBind();
}
else
{
DataList1.DataBind();
}
}
}
```

OUTPUT:-


localhost:8652/Default2.
localhost:8652/Default2.aspx
Apps
latest hollywood hin
Tenda 1
Treeview control navigation:
ASP.NET Practs
Calendar Control
Constructor Overloading
Inheritance
Class Properties
Fetch Datalist Using XML data :
Roll Num : 1
Name : Tushar
Class : TYIT
Roll Num : 2
Name : Sonali
Class : TYCS
Roll Num : 3
Name : Yashashree
Class : TYIT
Roll Num : 4
Name : Vedshree
Class : TYCS

Practical No.: 4

AIM: Working with form controls

A) Create a web form to demonstrate the Adrotator Control.

XML File

```
<Advertisements>
<Ad>
<ImageUrl>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>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>rose3.jpeg</ImageUrl>
<NavigateUrl>http://www.flowers2moscow.com</NavigateUrl>
<AlternateText>Send flowers to Russia</AlternateText>
<Impressions>20</Impressions>
<Keyword>russia</Keyword>
</Ad>
</Advertisements>
```

Default.aspx

```
<asp:AdRotator ID="AdRotator1" runat="server"
DataSourceID="XmlDataSource1" />
<asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/ADFILE.xml"></asp:XmlDataSource>
```

OUTPUT:

B) Create web form to demonstrate use User Controls.

MyUserControl.ascx

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="MyUserControl.ascx.cs" Inherits="MyUserControl" %>
<h3>This is User Contro1 </h3>
<table>
<tr>
<td>Name</td>
<td>
<asp:TextBox ID="txtName" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>City</td>
<td><asp:TextBox ID="txtcity" runat="server"></asp:TextBox></td>
</tr>

<tr>
<td></td>
<td>
</td>
</tr>
<tr>
<td></td>
<td>
<asp:Button ID="txtSave" runat="server" Text="Save"
onclick="txtSave_Click" />
</td>
</tr>
</table><br />
<asp:Label ID="Label1" runat="server" ForeColor="White" Text="
"></asp:Label>
```

MyUserControl.ascx.cs

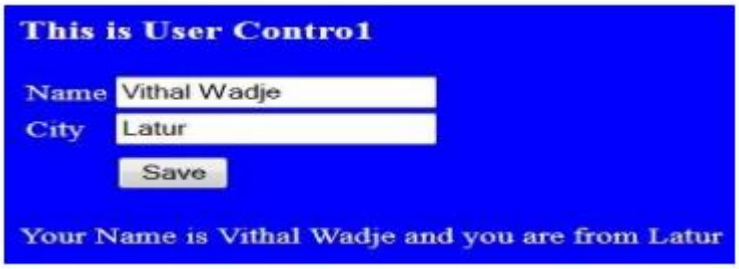
```
protected void txtSave_Click(object sender, EventArgs e)
```

```
{
Label1.Text = "Your Name is " + txtName.Text + " and you are from " +
txtcity.Text;
}
```

UserControlDisplay.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UserControlDisplay.aspx.cs" Inherits="UserControlDisplay"
%>
<%@ Register Src="~/MyUserControl.ascx" TagPrefix="uc"
TagName="Student"%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<uc:Student ID="studentcontrol" runat="server" />
</div>
</form>
</body>
</html>
```

OUTPUT :



Practical No.: 5

AIM:- Working with Navigation, Beautification and Master page

A) Create a web application to demonstrate use of Master Page withapplying Styles and Themes for page beautification.

MasterPage.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">

<title>Master Page Demo</title>
<link href="css/my.css" rel="stylesheet" />
<asp:ContentPlaceHolder ID="head" runat="server">
</asp:ContentPlaceHolder>
<style type="text/css">
.auto-style1 {
position: absolute;
top: 373px;
left: 1028px;
bottom: 303px;
}
.auto-style2 {
position: absolute;
top: 537px;
left: 1016px;
z-index: 1;
}
</style>
</head>
<body>
```

```
<!DOCTYPE html>
<form id="form1" runat="server">
<html>
<head>
<title>Master</title>
<link rel="stylesheet" type="text/css" href="StyleSheet.css">
</head>
<body>
<header id="header">
<h1>Demo Of Master Page</h1>
</header>
<nav id="nav">
<ul>
<li><a href="home.aspx">Insight</a></li>
<li><a href="#">Products</a></li>
<li><a href="#">Downloads</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</nav>
<aside id="side">
<h1>Info</h1>
<a href="#"><p>Product Type 1</p></a>
<a href="#"><p>Product Type 2</p></a>
<a href="#"><p>Product Type 3<a href="#"><asp:ScriptManager
ID="ScriptManager1"
runat="server">
</asp:ScriptManager>
</a>
</p>
<asp:Button ID="Button2" runat="server" CssClass="auto-style1"
style="z-index: 1"
Text="Button" />
<asp:Button ID="Button1" runat="server" CssClass="auto-style2"
Text="Button" />
</aside>
<div id="con">
```

```
<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</div>
<footer id="footer">
copyright @Sambare
</footer>
</body>
</html>
</form>
</body>
</html>
```

MasterDisplay.aspx

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="MasterDisplay.aspx.cs"
Inherits="MasterDisplay" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<h1>Home page</h1>
</asp:Content>
```

StyleSheet.css

```
#header{
color: blueviolet;
text-align: center;
font-size: 20px;
}
#nav{
background-color:darkseagreen;
padding: 5px;
```

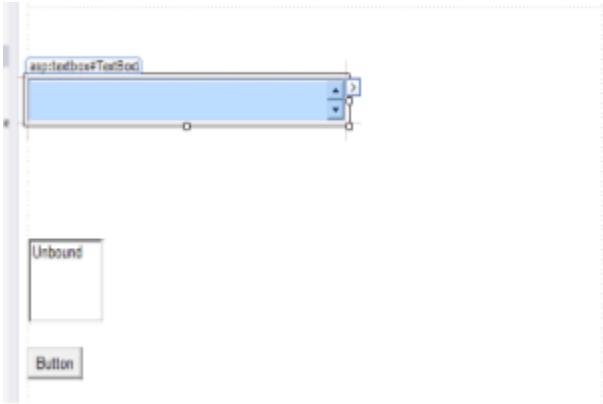
```
}
ul{
list-style-type: none;
}
li a {
color:crimson ;
font-size: 30px;
column-width: 5%;
}
li
{
display: inline;
padding-left: 2px;
column-width: 20px;
}
a{
text-decoration: none;
margin-left:20px
}

li a:hover{
background-color: aqua;
color:coral ;
padding:1%;
}
#side{
text-align: center;
float: right;
width: 15%;
padding-bottom: 79%;
background-color: #F1FAEE;
}
#article{
background-color: burlywood;
padding: 10px;
padding-bottom: 75%;
```

```
}
#footer{
background-color: #C7EFCF;
text-align:center;
padding-bottom: 5%;
font-size: 20px;
}
#con{
border:double;
border-color:burlywood;
}
```

Practical no 6 :

6 a)Create a web application bind data in a multiline textbox by querying in another textbox.

Default.aspx (create a web page with following design):-



Web.config:-

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.5.2" />

<httpRuntimetargetFramework="4.5.2" />

</system.web>

<connectionStrings>

<add name="connStr" connectionString="Data

Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\tushars\Documents\Visual Studio

2015\WebSites\Workshop\App_Data\Database.mdf';Integrated Security=True" />

</connectionStrings>

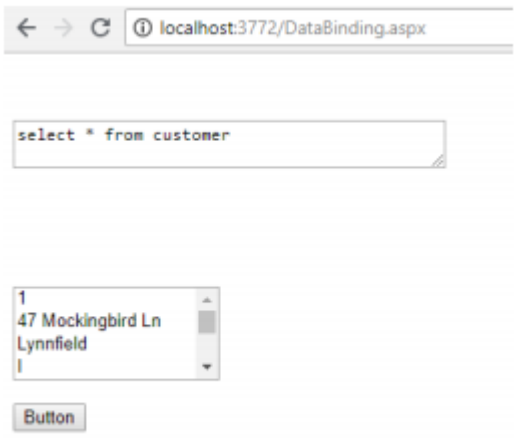
</configuration>

Default.aspx.cs:-

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;
usingSystem.Data;
usingSystem.Data.SqlClient;
usingSystem.Configuration;
public partial class DataBinding : System.Web.UI.Page
{
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
                stringconnStr =
                ConfigurationManager.ConnectionStrings["connStr"].ConnectionStrin
                g;
                SqlConnection con = new SqlConnection(connStr);
                con.Open();
                SqlCommandcmd = new SqlCommand(TextBox1.Text, con);
                SqlDataReader reader = cmd.ExecuteReader();
                ListBox1.Items.Clear();
```

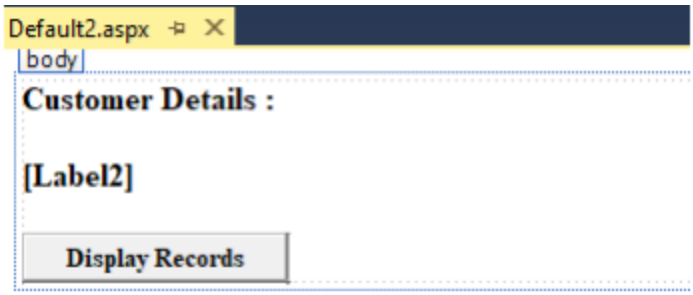
```
            while (reader.Read())
            {
                //To add new blank line in the text area

                for (int i = 0; i <reader.FieldCount - 1; i++)
                {
                    ListBox1.Items.Add(reader[i].ToString());
                }
            }
            reader.Close();
            con.Close();
        }
}
```

6a) output:-



6 b)Create a web application to display records by using database.

Default.aspx (create a web page with following design):-

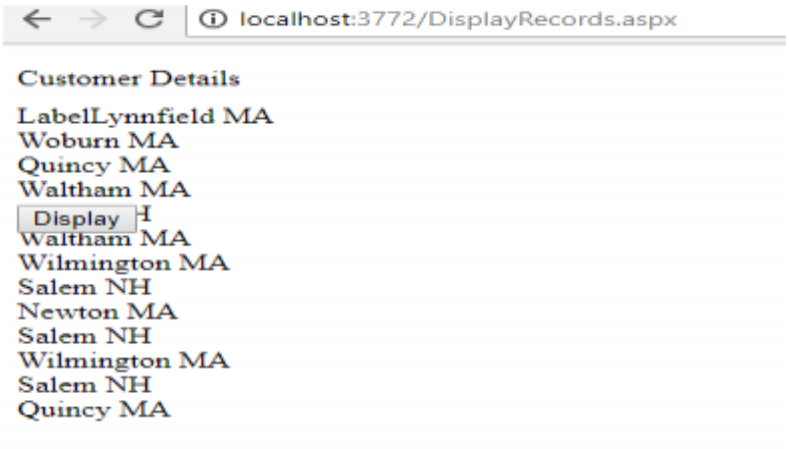


Default.aspx.cs :-

```
protected void Button1_Click(object sender, EventArgs e)
{
        stringconnStr =
        ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
        SqlConnection con = new SqlConnection(connStr);
        SqlCommandcmd = new SqlCommand("Select City, State from Customer",
        con);
        con.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
                Label1.Text += reader["City"].ToString() + " " +
                reader["State"].ToString() +
                "<br>";
        }



        reader.Close();
```

```
        con.Close();
}
```

6 b) output



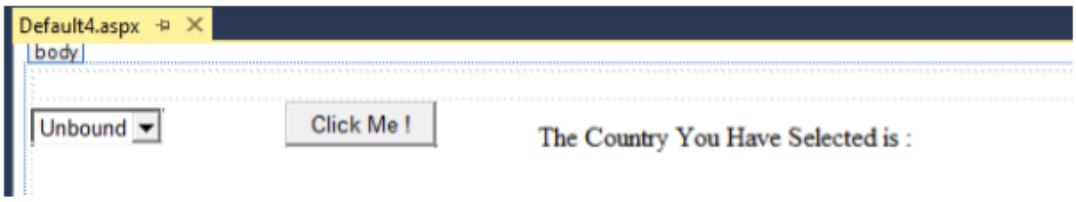
Practical no 7:-

7 (a): Create a web application to display Databinding using Dropdownlist control.

Default.aspx (create a web page with following design):-

1. Create a web page with DropDownList control, one Button and one Label control.

2. Use code to bind the data to DropDownList.



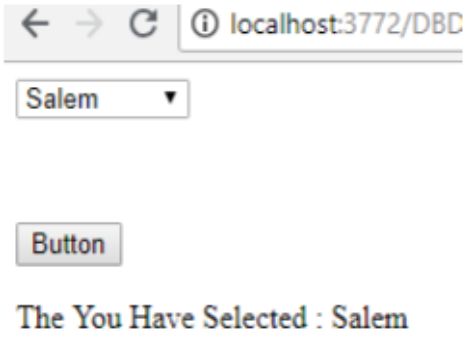
Default.aspx.cs:-

```
using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data;

usingSystem.Data.SqlClient;

usingSystem.Configuration;

public partial class DBDropDown : System.Web.UI.Page

{
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack == false)
    {
        stringconnStr =
        ConfigurationManager.ConnectionStrings["connStr"].Connecti
        onString;
        SqlConnection con = new SqlConnection(connStr);
        SqlCommandcmd = new SqlCommand("Select Distinct City
        from Customer", con);
        con.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        DropDownList1.DataSource = reader;
        DropDownList1.DataTextField = "City";
        DropDownList1.DataBind();
        reader.Close();
        con.Close();
    }

}

protected void Button1_Click(object sender, EventArgs e)
{
Label1.Text = "The You Have Selected : " + DropDownList1.SelectedValue;     }}
```

7 a) output:-

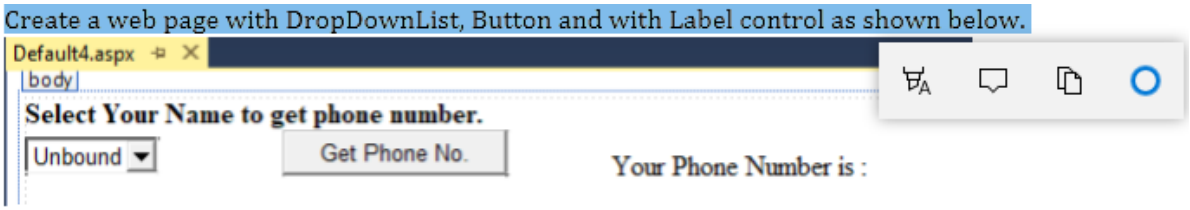

Lynnfield
Lynnfield
Newton
Quincy
Salem
Waltham
Wilmington
Woburn


localhost:3772/DBD
Salem
Button
The You Have Selected : Salem

7 (b): Create a web application for to display the phone no of an author using database.

Default.aspx (create a web page with following design):-

Create a web page with DropDownList, Button and with Label control as shown below.

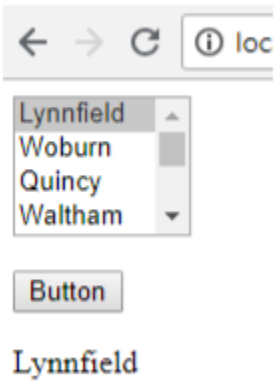


Default.aspx.cs:-

```
using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data;

usingSystem.Data.SqlClient;

usingSystem.Configuration;


public partial class PostalCodeByCity : System.Web.UI.Page

{

        protected void Button1_Click(object sender, EventArgs e)

        {

                Label1.Text = ListBox1.SelectedValue;
```

```
    }
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {
            stringconnStr =
            ConfigurationManager.ConnectionStrings["connStr"].Connecti
            onString;
            SqlConnection con = new SqlConnection(connStr);
            SqlCommandcmd = new SqlCommand("Select Distinct
            POSTAL_CODE from Customer", con);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            ListBox1.DataSource = reader;
            ListBox1.DataTextField = "City";          ListBox1.DataValueField
            = "POSTAL_CODE";          ListBox1.DataBind();
            reader.Close();
            con.Close();



        }
    }
}
```

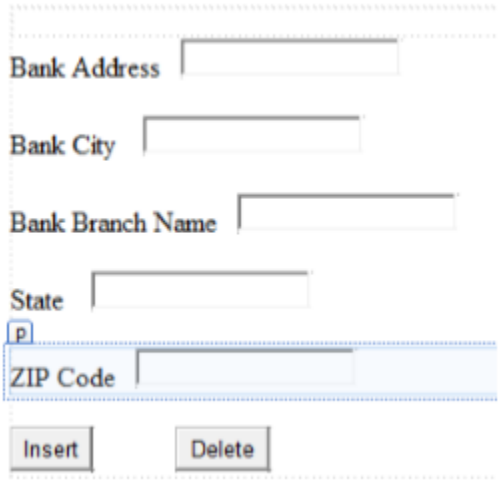
7b) output :-



---

7 (c): Create a web application for inserting and deleting record from a database. (Using Execute-Non Query).

Default.aspx (create a web page with following design):-



Default.aspx.cs:-

```
using System;
usingSystem.Collections.Generic;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;
usingSystem.Data;
usingSystem.Data.SqlClient;
usingSystem.Configuration;
public partial class ExecuteNonQuery : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        stringconnStr =
        ConfigurationManager.ConnectionStrings["connStr"].ConnectionStrin
        g;      SqlConnection con = new SqlConnection(connStr);
        string InsertQuery = "insert into BRANCH values(@ADDRESS, @CITY,
        @NAME, @STATE, @ZIP_CODE)";
        SqlCommandcmd = new SqlCommand(InsertQuery, con);
        cmd.Parameters.AddWithValue("@ADDRESS", TextBox1.Text);
        cmd.Parameters.AddWithValue("@CITY", TextBox2.Text);
        cmd.Parameters.AddWithValue("@NAME", TextBox3.Text);
        cmd.Parameters.AddWithValue("@STATE", TextBox4.Text);
        cmd.Parameters.AddWithValue("@ZIP_CODE", TextBox5.Text);
        con.Open();
        cmd.ExecuteNonQuery();
        Label1.Text = "Record Inserted Successfuly.";
        con.Close();
    }
```

```
protected void Button2_Click(object sender, EventArgs e)
{
        stringconnStr =
        ConfigurationManager.ConnectionStrings["connStr"].ConnectionStrin
        g;      SqlConnection con = new SqlConnection(connStr);
        stringInsertQuery = "delete from branch where NAME=@NAME";
        SqlCommandcmd = new SqlCommand(InsertQuery, con);
        cmd.Parameters.AddWithValue("@NAME", TextBox1.Text);
        con.Open( );
        cmd.ExecuteNonQuery( );
        Label1.Text = "Record Deleted Successfuly.";
        con.Close( ); } }
```

8)b) Create a web application To demonstrate data binding using DetailsView and FormView control.

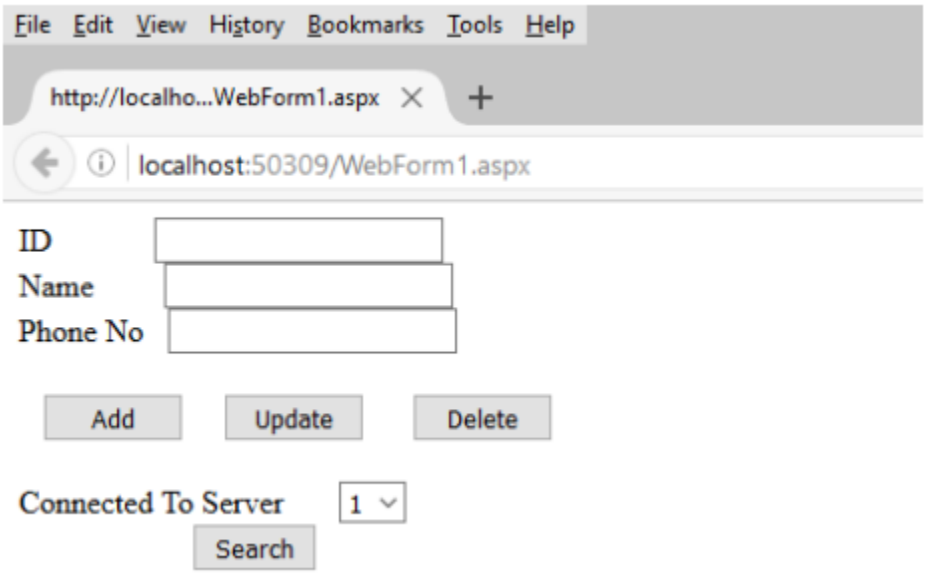
```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
```

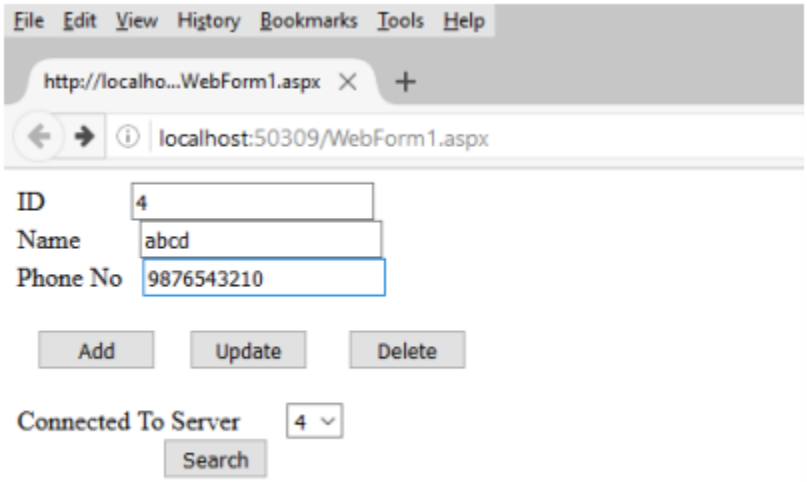
```
using System.Web.UI.WebControls;
using System.Collections.Generic;
namespace WebApplication1
{
public partial class WebForm1 : System.Web.UI.Page
{
SqlDataAdapter da = new SqlDataAdapter();
SqlConnection con = new SqlConnection();
SqlCommand cmd = new SqlCommand();
DataSet ds = new DataSet();
string str;
protected void Page_Load(object sender, EventArgs e)
{
con.ConnectionString = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=C:\\Users\\SAHIL\\Documents\\Stud
ents.mdf;Inte
grated Security=True;Connect Timeout=30";
con.Open();
Label4.Text = "Connected To Server";
con.Close();
}
protected void Button1_Click(object sender, EventArgs e)
{
str = "insert into stud_mast values(" + TextBox1.Text + " , ' " + TextBox2.Text + " ' ,
" +
TextBox3.Text + ")";
```

```
con.Open();
cmd = new SqlCommand(str,con);
cmd.ExecuteNonQuery();
con.Close();
Label4.Text = " Save Successfull ";

TextBox1.Text = " ";
TextBox2.Text = " ";
TextBox3.Text = " ";
}
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs
e)
{
}
protected void Button4_Click(object sender, EventArgs e)
{
str = "select * from stud_mast where stud_id= " + DropDownList1.Text + " ";
da = new SqlDataAdapter(str, con);
ds = new DataSet();
da.Fill(ds,"stud_mast");
TextBox1.Text = ds.Tables["stud_mast"].Rows[0]["stud_id"].ToString();
TextBox2.Text = ds.Tables["stud_mast"].Rows[0]["stud_name"].ToString();
TextBox3.Text = ds.Tables["stud_mast"].Rows[0]["phn_no"].ToString();
}
protected void Button2_Click(object sender, EventArgs e)
```

```
{
str = "update stud_mast set stud_name= ' " + TextBox2.Text + " ', phn_no= "
+TextBox3.Text+" where stud_id= "+DropDownList1.Text+" ";
con.Open();
cmd = new SqlCommand(str, con);
cmd.ExecuteNonQuery();
con.Close();
Label4.Text = " Update Successfull ";
}
protected void Button3_Click(object sender, EventArgs e)
{
str = "delete from stud_mast where stud_id=" + DropDownList1.Text + " ";
con.Open();
cmd = new SqlCommand(str, con);
cmd.ExecuteNonQuery();
con.Close();
Label4.Text = " Update Successfull ";
}}}
```

Output:-

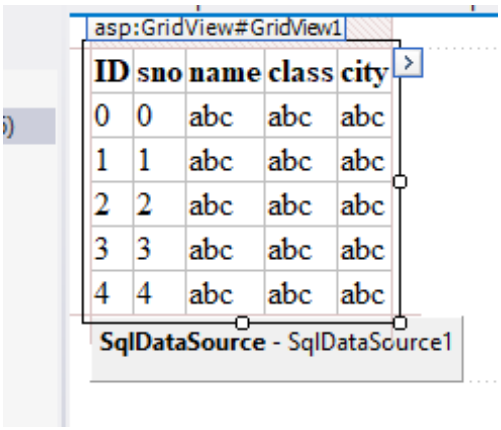

File Edit View History Bookmarks Tools Help
http://localho...WebForm1.aspx
localhost:50309/WebForm1.aspx
ID
Name
Phone No
Add
Update
Delete
Connected To Server
1
Search


File Edit View History Bookmarks Tools Help
http://localho...WebForm1.aspx
localhost:50309/WebForm1.aspx
ID
4
Name
abcd
Phone No
9876543210
Add
Update
Delete
Connected To Server
4
Search

Practical no.9

Create a web application to demonstrate use of GridView button column and GridView events.

Grid_view.aspx:-

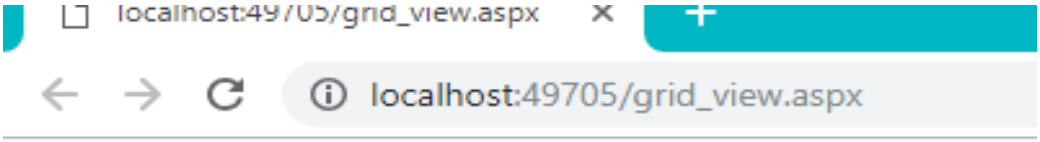


Grid_view.aspx.cs:-

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;
usingSystem.Drawing;
public partial class grid_view : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
   {

   }
protected void GridView1_RowCommand(object sender,
GridViewCommandEventArgs e)
   {
if (e.CommandName == "b1")
      {
Response.Write(e.CommandName);
         GridView1.SelectedRowStyle.BackColor=System.Drawing.Color.Brown;
```

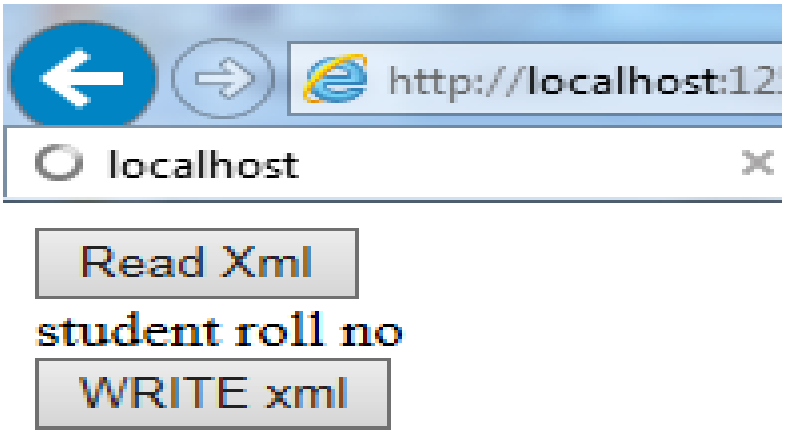
```
GridView1.Rows[Convert.ToInt16(e.CommandArgument)].BackColor =
System.Drawing.Color.Blue;
      }
   }
}
```

9) output:-



ID	sno	name	class	city
1	6	man	FYIT	nerul
2	3	nmj	SYIT	nerul
3	4	nmo	sycs	nerul
4	7	hj	hj	v

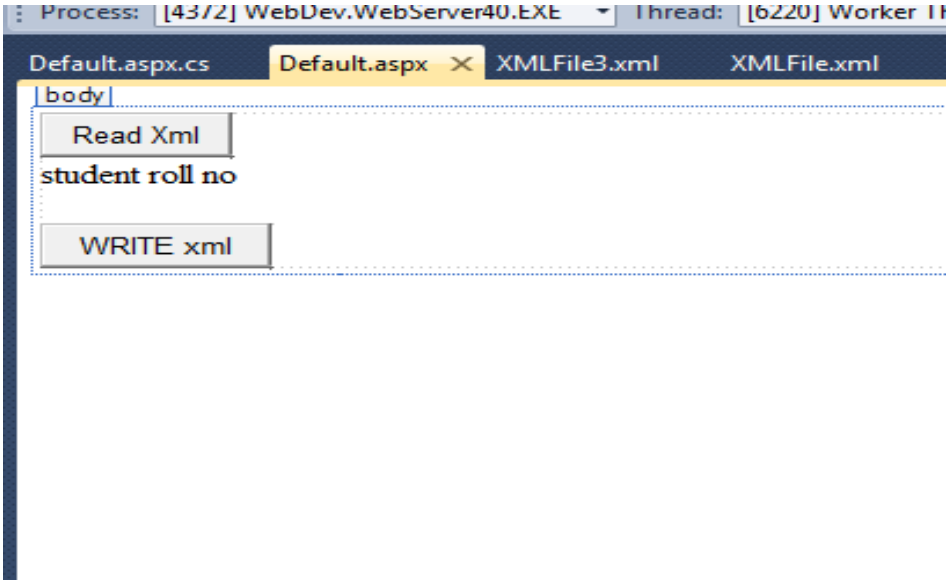
10 a) output:-


http://localhost:12
localhost
Read Xml
student roll no
WRITE xml

Practical no 10

10 a) Create a web application to demonstrate reading and writing operation with XML.

Default.aspx:-



Default.aspx.cs:-

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

public partial class _Default : System.Web.UI.Page
{
protected void Button1_Click(object sender, EventArgs e)
   {
```

```
XmlReader red = XmlReader.Create(@"C:\Users\Admin\Documents\Visual Studio
2010\WebSites\WebSite24\XMLFile.xml");
while (red.Read())
      {
if (red.NodeType.Equals(XmlNodeType.Element))
         {
string s = Label1.Text + "";
            Label1.Text = s + red.Name;
         }
      }
red.Close();
   }
protected void Button2_Click(object sender, EventArgs e)
   {
XmlWriterSettings set = new XmlWriterSettings();
set.Indent = true;
XmlWriterwr = XmlWriter.Create(@"C:\Users\Admin\Documents\Visual Studio
2010\WebSites\WebSite24\XMLFile3.xml",set);
wr.WriteStartDocument();
wr.WriteComment("EXAMPLE OF WRITE A XML DOCUMENT");
wr.WriteStartElement("student");
wr.WriteEndElement();
   }
}
```

Practical 11:-

11) Programs to create and use DLL

Class1.cs:-

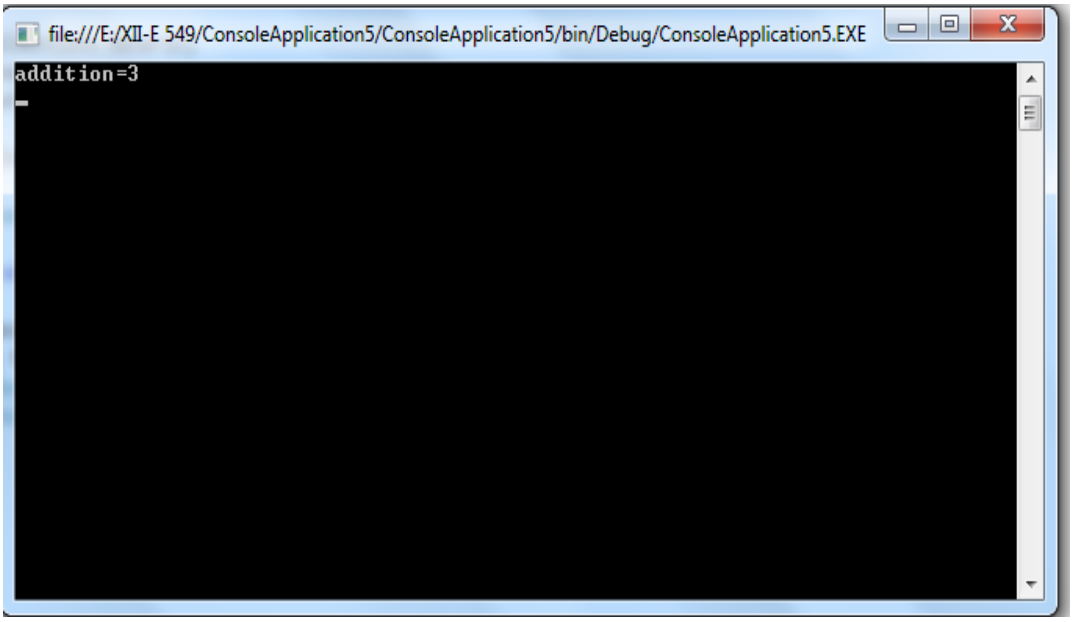
```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;

namespace ClassLibrary5
{
publicclassClass1
    {
publicint add(int a, int b)
        {
int c = a + b;
return c;
        }
    }
}
```

Consoleapplication5.cs:-

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

namespace ConsoleApplication5
{
classProgram
    {
staticvoid Main(string[] args)
        {
          ClassLibrary5.Class1 c = newClassLibrary5.Class1();
int t = c.add(1, 2);
Console.WriteLine("addition={0}", t);
Console.ReadKey();
```

```
            }
        }
}
```

11) output:-


file:///E:/XII-E 549/ConsoleApplication5/ConsoleApplication5/bin/Debug/ConsoleApplication5.EXE
addition=3