

## 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) OUTPUT:-**

Enter number 1: 6

Enter number 2: 5

Enter number 3: 4

Enter number 4: 3

$6*5*4*3=360$

## B) Create an application to demonstrate string operations.

```
using System;
namespace cmdLineArgs
{
    class Program
    {
        static void Main(string[] args)
        {
            string str = args[0];
            int n = Convert.ToInt32(args[1]);
            Console.WriteLine("String:" + str);
            Console.WriteLine("Number:" + n);
        }
    }
}
```

### **OUTPUT:**

String : Roman

Number : 10

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[5];
            int i;
            for (i = 0; i < 5; 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 < 5; 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);
} } }
```

1)C) OUTPUT:

Enter Student Id:0001

Enter Student name : Prachit

Enter Course name : MSCit

Enter date of birth

Enter day(1-31):29

Enter month(1-12):9

Enter year:1995

Enter Student Id:0002

Enter Student name : Aniket

Enter Course name : Bscit

Enter date of birth

Enter day(1-31):4

Enter month(1-12):3

Enter year:1996

Enter Student Id:0003

Enter Student name : Prathamesh

Enter Course name : BMS

Enter date of birth

Enter day(1-31):9

Enter month(1-12):8

Enter year:2000

Enter Student Id:0004

Enter Student name : Sumit

Enter Course name :MScet

Enter date of birth

Enter day(1-31):25

Enter month(1-12):5

Enter year:1994

Enter Student Id : 0005

Enter Student name : Zaid

Enter Course name : BCOM

Enter date of birth

Enter day(1-31):6

Enter month(1-12):7

Enter year:1993

Student's List

Student ID : 0001

Student name : Prachit

Course name : MSCit

Date of birth(dd-mm-yy) : 29-9-1995

Student ID : 0002

Student name : Aniket

Course name : Bscit

Date of birth(dd-mm-yy) : 4-3-1996

Student ID : 0003

Student name : Prathamesh

Course name : BMS

Date of birth(dd-mm-yy) : 9-8-2000

Student ID : 0004

Student name : Sumit

Course name : MScet

Date of birth(dd-mm-yy) : 25-5-1994

Student ID : 0005

Student name : Zaid

Course name : BCOM

Date of birth(dd-mm-yy) : 6-7-1993

## 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++;

}



```
}  
}  
}
```

### **1)D)[i]OUTPUT:**

**Upto how many number you want fibonacci series:5**

**0 1 1 2 3**

---

### **[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)

Enter number:3

3 is prime number

(2nd)

Enter number:1

1 is neither prime nor composite

(3rd)

Enter number:4

4 is not prime number

### **[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:**

Enter a character : a

a is vowel

Enter a character : p

p is not a vowel

#### **[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:15

Reverse of 15=51

Sum of its digits:6

-----

## 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:- 6

Factorial of given number is:- 720

### [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("&quot;Enter your Choice :\n 1- Dollar to Rupee \n 2-
            Euro to Rupee \n 3- Malaysian
            Ringgit to Rupee &quot;");
            choice = int.Parse(Console.ReadLine());
            switch (choice)
            {
                case 1:
                    Double dollar, rupee, val;
                    Console.WriteLine("&quot;Enter the Dollar Amount :&quot;");
                    dollar = Double.Parse(Console.ReadLine());
                    Console.WriteLine("&quot;Enter the Dollar Value :&quot;");
                    val = double.Parse(Console.ReadLine());
                    rupee = dollar * val;

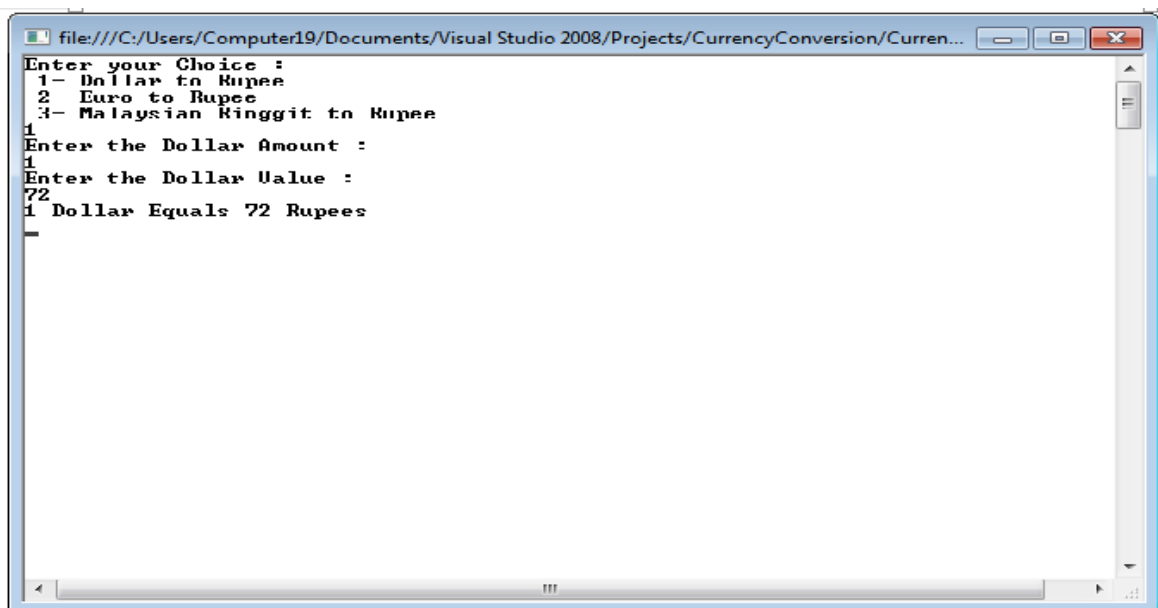
                    Console.WriteLine("&quot;{0} Dollar Equals {1} Rupees&quot;;", dollar,
                    rupee);
                    break;
                case 2:
                    Double Euro, rupe, valu;
                    Console.WriteLine("&quot;Enter the Euro Amount :&quot;");
                    Euro = Double.Parse(Console.ReadLine());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



```

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

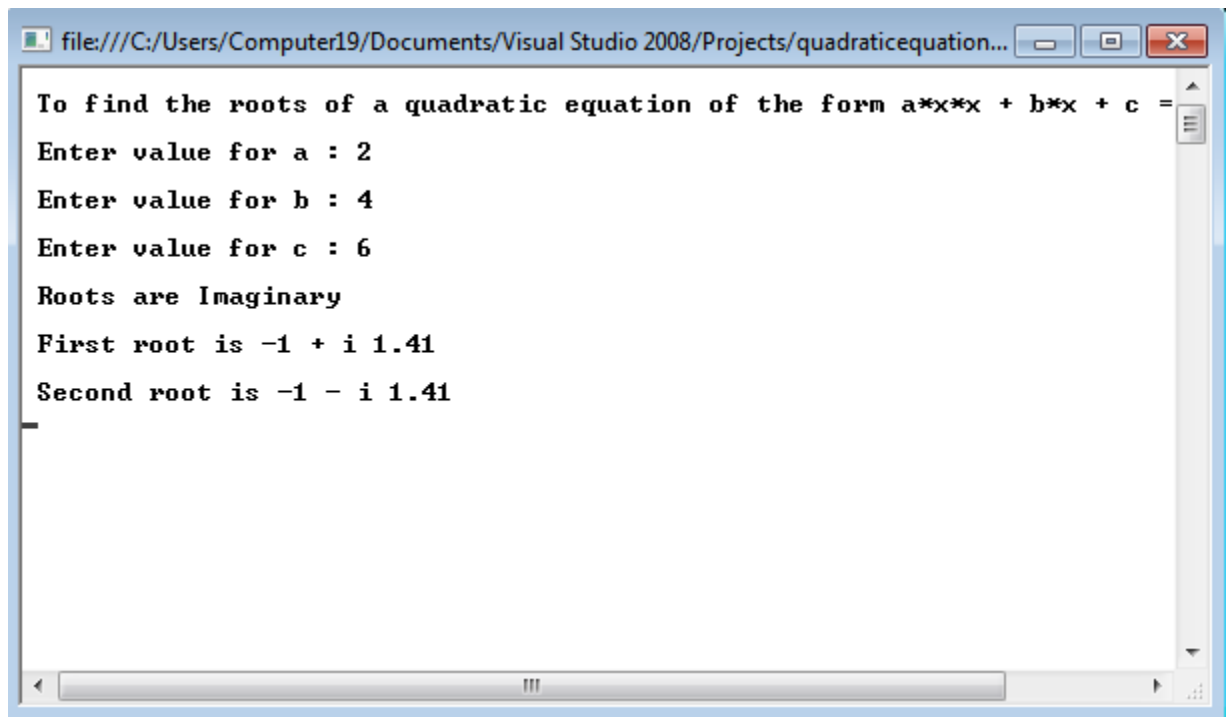
                Console.ReadLine();
                break;
        }
    }
}

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

```

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

2)A)[iii]

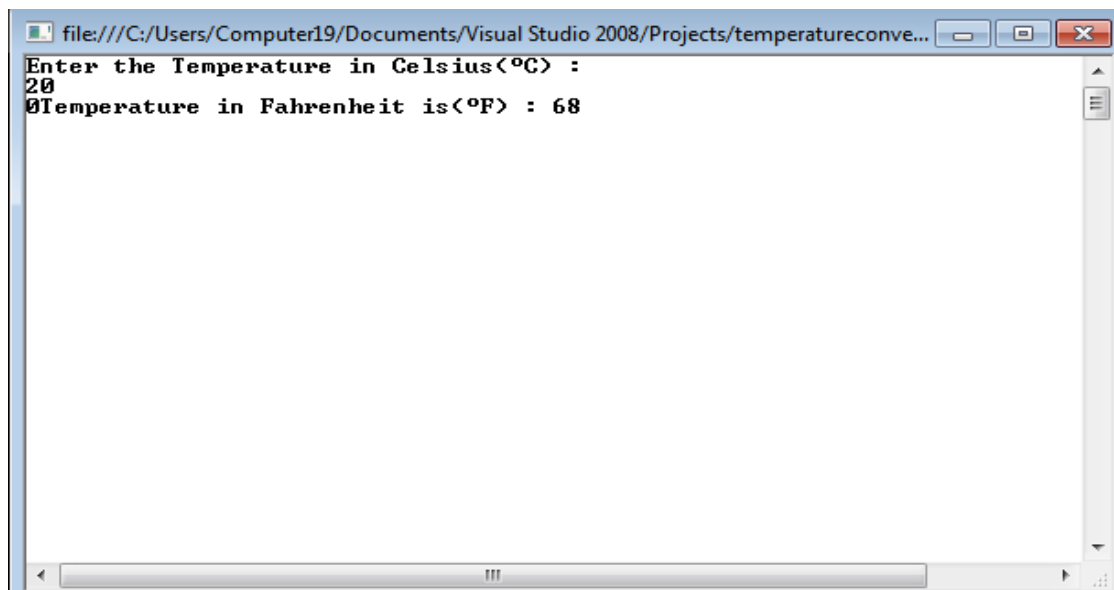


## [iv] Temperature Conversion

```
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("Temperature 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:

N=20 M=10

F1=20.6 F2=10.5

---

## [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 display()
{
    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 allowance : " + da);
    Console.WriteLine("Housing rent allowance : " + hra);
    Console.WriteLine("Travelling allowance : " + 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 allowance : 9000  
Housing rent allowance : 6000  
Travelling allowance : 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

A screenshot of a console window with a black background and white text. The text is organized into two groups. The first group contains four lines: 'This is ODD', 'This is ONE', 'This is THERE', and 'This is FIVE'. There is a blank line after this group. The second group contains three lines: 'This is EVEN', 'This is TWO', and 'This is FOUR'.

```
This is ODD  
This is ONE  
This is THERE  
This is FIVE  
  
This is EVEN  
This is TWO  
This is FOUR
```

(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 following operations.

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

Calendar properties set for this example:

```
<asp:Calendar ID="Calendar1" runat="server" BackColor="#FFFFCC"
BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399"
Height="200px"
NextPrevFormat="ShortMonth" OnDayRender="Calendar1_DayRender"
ShowGridLines="True" Width="300px"
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>
```

```

calndrCtrl.aspx.cs
protected void btnResult_Click(object sender, EventArgs e)
{
    Calendar1.Caption = "SAMBARE";
    Calendar1.FirstDayOfWeek = FirstDayOfWeek.Sunday;
    Calendar1.NextPrevFormat = NextPrevFormat.ShortMonth;
    Calendar1.TitleFormat = TitleFormat.Month;

    Label2.Text = "Todays Date"+Calendar1.TodaysDate.ToShortDateString();
    Label3.Text = "Ganpati Vacation Start: 9-13-2018";
    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.ToShortDateString() == "9-13-2018")
        Label3.Text = "<b>Ganpati Festival Start</b>";
    if (Calendar1.SelectedDate.ToShortDateString() == "9-23-2018")
        Label3.Text = "<b>Ganpati Festival End</b>";
}

protected void Calendar1_DayRender(object sender,
System.Web.UI.WebControls.DayRenderEventArgs e)
{
    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);
    }
}

```

```

if (e.Day.Date.Day == 13 && e.Day.Date.Month == 9)
{
    Calendar1.SelectedDate = new DateTime(2018, 9, 12);
    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.Date.ToString();
}

```

### 3)A) OUTPUT:-

SAMBARE

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

Your selected date : 27-08-2018 00:00:00

Today's Date : 15-07-2018

Ganpati Vacation Start: 9-13-2018

Days Remaining For Ganpati Vacation : 59

Days remeaning for new year : 168

Result
Reset

- 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 version="1.0" encoding="utf-8" ?>
```

```
<studentdetail>
```

```
<student>
```

```
<sid>1</sid>
```

```
<sname>Tushar</sname>
```

```
<scsname>TYIT</scsname>
```

```
</student>
```

```
<student>
```

```
<sid>2</sid>
```

```
<sname>Sonali</sname>
```

```
<scsname>TYCS</scsname>
```

```
</student>
```

```
<student>
```

```
<sid>3</sid>
```

```
<sname>Yashashree</sname>
```

```
<scsname>TYIT</scsname>
```

```
</student>
```

```
<student>
```

```
<sid>4</sid>
```

```
<sname>Vedshree</sname>
```

```
<scsname>TYCS</scsname>
```

```
</student>
```

```
</studentdetail>
```

Default2.aspx

```
<form id="form1" runat="server">
```

```
<div>
```

Treeview control navigation:<asp:TreeView ID = "TreeView1" runat =

"server" Width =

"150px" ImageSet="Arrows">

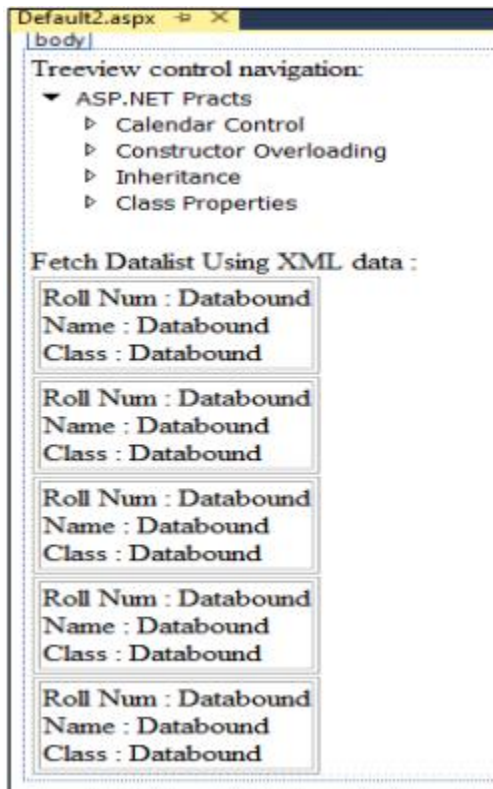
```

<HoverNodeStyle Font-Underline="True" ForeColor="#5555DD" />
<Nodes>
<asp:TreeNode Text = "ASP.NET Practs" Value = "New Node">
<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 NavigateUrl="~/singleInh.aspx" Text="Inheritance"
Value="BLUE"></asp:TreeNode>
<asp:TreeNode NavigateUrl="~/clsProp.aspx" Text="Class Properties"
Value="Class
Properties"></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>

```

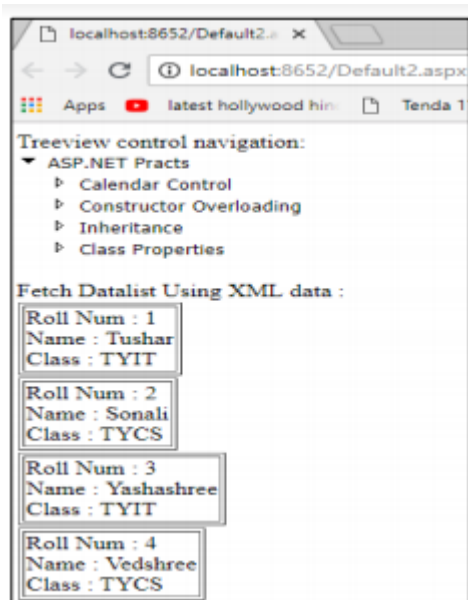




**Default2.aspx.cs**

```
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();
        }
    }
}
```

### 3)B)b) OUTPUT:-



## 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>
</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 :**



**This is User Control1**

Name

City

Your Name is Vithal Wadje and you are from Latur

## Practical No.: 5

AIM:- Working with Navigation, Beautification and Master page

- A) Create a web application to demonstrate use of Master Page with applying 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
}

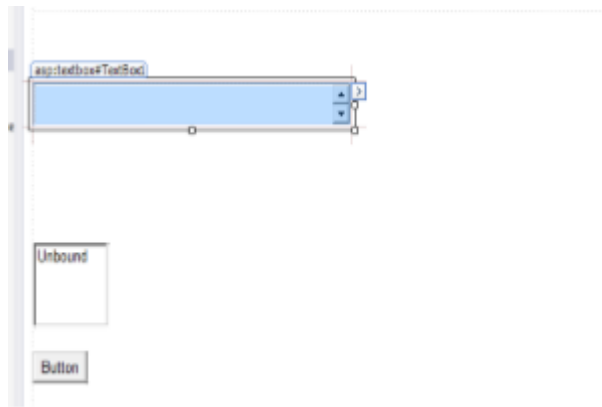
li a:hover{
background-color: aqua;
color:coral ;
padding:1%;
}
#side{
text-align: center;
float: right;
width: 15%;
padding-bottom: 79%;
background-color: #F1FAEE;
}
#article{
background-color: burlywood;
padding: 10px;
padding-bottom: 75%;
}
```

```
#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" />
```

```
<httpRuntime targetFramework="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;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.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)
    {
        string connStr =
            ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
        SqlConnection con = new SqlConnection(connStr);
        con.Open();
        SqlCommand cmd = new SqlCommand(textBox1.Text, con);
        SqlDataReader reader = cmd.ExecuteReader();
        listBox1.Items.Clear();
    }
}
```

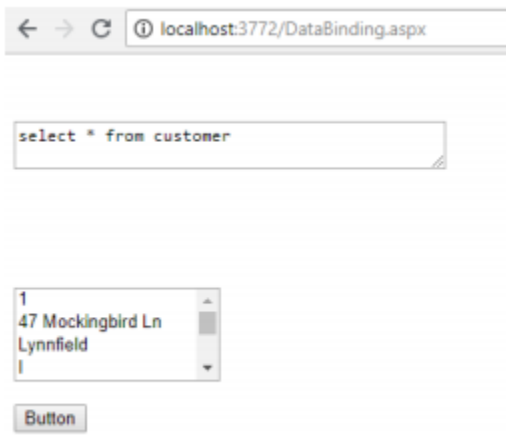
```

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();
}
}

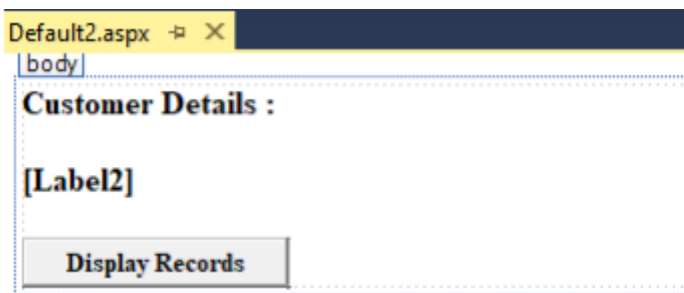
```

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)
```

```
{
```

```
    string connStr =
```

```
    ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
```

```
    SqlConnection con = new SqlConnection(connStr);
```

```
    SqlCommand cmd = 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**



**6 c) output:-**

TXN\_ID: 1  
AMOUNT: 100  
FUNDS\_AVAIL\_DATE: 1/15/2000 12:00:00 AM  
TXN\_DATE: 1/15/2000 12:00:00 AM  
TXN\_TYPE\_CD: CDT  
ACCOUNT\_ID: 1  
EXECUTION\_BRANCH\_ID:  
TELLER\_EMP\_ID:

TXN\_ID: 2  
AMOUNT: 100  
FUNDS\_AVAIL\_DATE: 1/15/2000 12:00:00 AM  
TXN\_DATE: 1/15/2000 12:00:00 AM  
TXN\_TYPE\_CD: CDT  
ACCOUNT\_ID: 2  
EXECUTION\_BRANCH\_ID:  
TELLER\_EMP\_ID:

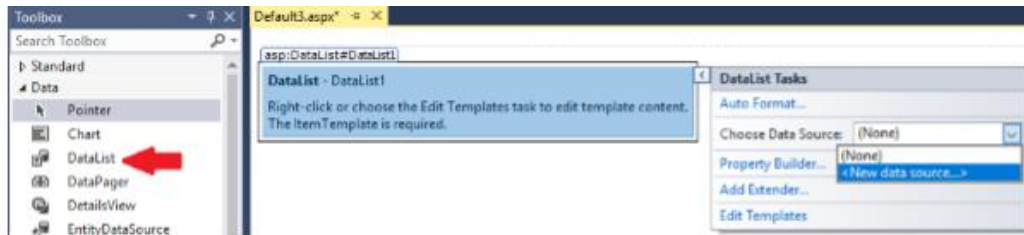
TXN\_ID: 3  
AMOUNT: 100  
FUNDS\_AVAIL\_DATE: 6/30/2004 12:00:00 AM  
TXN\_DATE: 6/30/2004 12:00:00 AM  
TXN\_TYPE\_CD: CDT  
ACCOUNT\_ID: 3  
EXECUTION\_BRANCH\_ID:  
TELLER\_EMP\_ID:

6 (c): Demonstrate the use of Datalist link control.

## **steps**

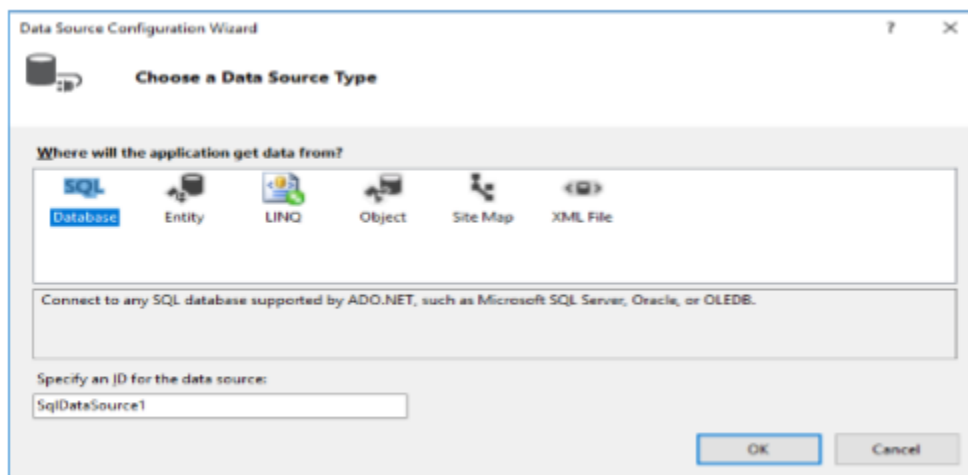
1. Drag the Datalist control to our web page form toolbox->Data->Datalist.

2. Then select Choose Data Source Option and select <New Data Source>.



3.

Now Select SQL Database from options and Click Ok button.



4. In next window click on New Connection button.

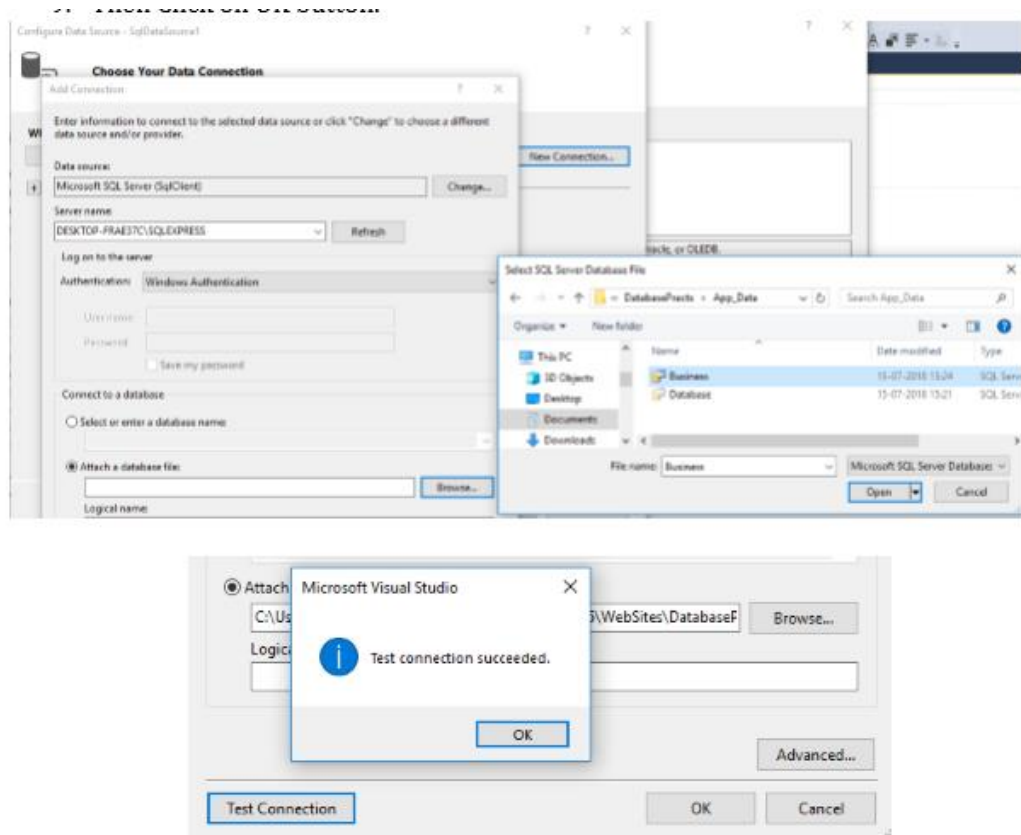
5. In add connection window Select the available SQL Server Name

6. Keep the Authentication as Windows Authentication.

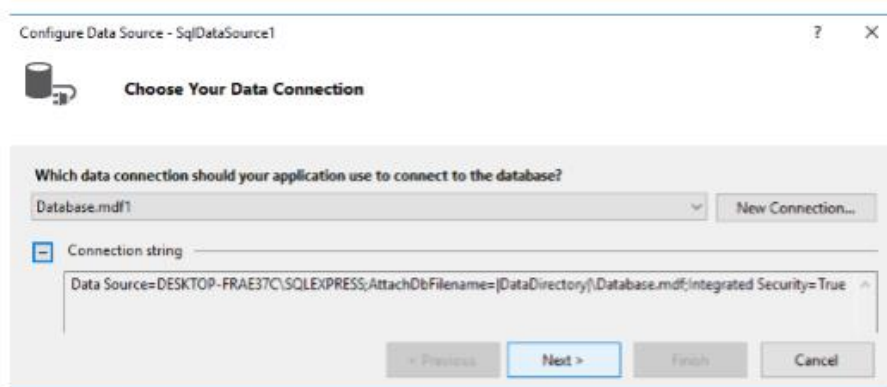
7. After that select Attach a Database file radio button. Here we have to select the database that we have created in our application. (Usually it will be in Documents folder under Visual Studio 2015/ Websites).

8. After selection of Database file. We can also Test the connection.

9. Then Click on OK button.



10. Once the Connection is made then click on Next button from Data Source Wizard.



11. Then wizard ask for saving the connection string in configuration file. If you already stored it web.config file then uncheck check box, if you haven't, then select the checkbook. Then click on next button.

12. The next screen gives option to configure the select statement. Here we can choose the table as well as configure the select statement as we need to display the data on web page.



## Configure the Select Statement

How would you like to retrieve data from your database?

☐ Specify a custom SQL statement or stored procedure

☒ Specify columns from a table or view

Name:  
ACC\_TRANSACTION

Columns:

<input type="checkbox"/> *	<input type="checkbox"/> Return only unique rows
<input checked="" type="checkbox"/> TXN_ID	WHERE...
<input checked="" type="checkbox"/> AMOUNT	ORDER BY...
<input checked="" type="checkbox"/> FUNDS_AVAIL_DATE	Advanced...
<input checked="" type="checkbox"/> TXN_DATE	
<input checked="" type="checkbox"/> TXN_TYPE_CD	
<input checked="" type="checkbox"/> ACCOUNT_ID	
<input checked="" type="checkbox"/> EXECUTION_BRANCH_ID	
<input checked="" type="checkbox"/> TELLER_EMP_ID	

SELECT statement:

```
SELECT [TXN_ID], [AMOUNT], [FUNDS_AVAIL_DATE], [TXN_DATE], [TXN_TYPE_CD], [ACCOUNT_ID],  
[EXECUTION_BRANCH_ID], [TELLER_EMP_ID] FROM [ACC_TRANSACTION]
```

< Previous   Next >   Finish   Cancel

13. In next screen we can test our query to check the output. Then Click on finish.

## 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;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

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

```

    {
        if (IsPostBack == false)
        {
            string connStr =
                ConfigurationManager.ConnectionStrings["connStr"].Connection
                onString;

            SqlConnection con = new SqlConnection(connStr);

            SqlCommand cmd = 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();

        }

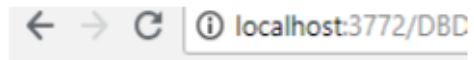
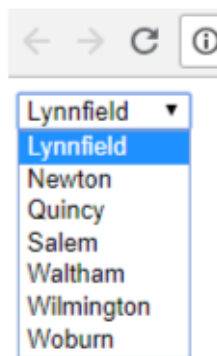
    }

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

```



**7 a) output:-**



Salem ▼

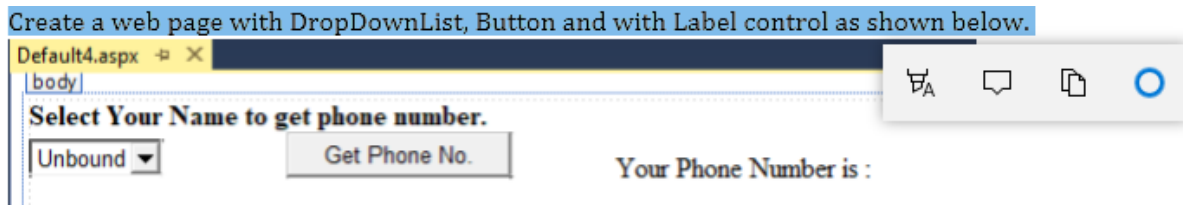
Button

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;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;
```

```
using System.Web.UI;
```

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

```
using System.Data;
```

```
using System.Data.SqlClient;
```

```
using System.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)
        {
            string connStr =
                ConfigurationManager.ConnectionStrings["connStr"].Connection
                String;

            SqlConnection con = new SqlConnection(connStr);
            SqlCommand cmd = 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();

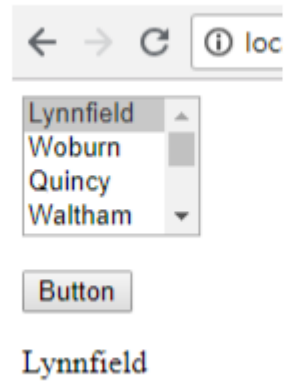
            reader.Close();

            con.Close();

        }
    }
}

```

**7b) output :-**



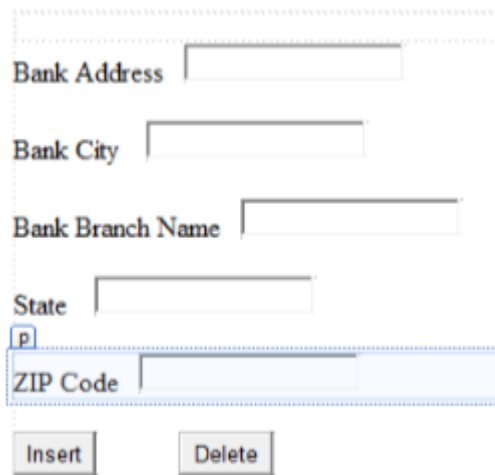
The screenshot shows a web application interface. At the top, there is a navigation bar with four icons: a left arrow, a right arrow, a circular refresh icon, and an information icon labeled 'loc'. Below the navigation bar is a dropdown menu with four options: 'Lynnfield', 'Woburn', 'Quincy', and 'Waltham'. The 'Lynnfield' option is currently selected. Below the dropdown menu is a button labeled 'Button'. Below the button, the text 'Lynnfield' is displayed.

---

-----

**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):-**



The screenshot shows a web application form. It contains five text input fields with labels: 'Bank Address', 'Bank City', 'Bank Branch Name', 'State', and 'ZIP Code'. The 'ZIP Code' field is highlighted with a blue border. Below the input fields are two buttons: 'Insert' and 'Delete'.

---

**Default.aspx.cs:-**

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class ExecuteNonQuery : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        string connStr =
        ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
        SqlConnection con = new SqlConnection(connStr);

        string InsertQuery = "insert into BRANCH values(@ADDRESS, @CITY,
        @NAME, @STATE, @ZIP_CODE)";

        SqlCommand cmd = 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 Successfully.";

        con.Close();
    }
}

```

```

protected void Button2_Click(object sender, EventArgs e)
{
    string connStr =
    ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
    SqlConnection con = new SqlConnection(connStr);

    string InsertQuery = "delete from branch where NAME=@NAME";
    SqlCommand cmd = new SqlCommand(InsertQuery, con);
    cmd.Parameters.AddWithValue("@NAME", TextBox1.Text);

    con.Open( );

    cmd.ExecuteNonQuery( );

    Label1.Text = "Record Deleted Successfully.";

    con.Close( ); } }

```

## Practical no.8

### 8)a) Create a web application to demonstrate Various uses and properties of SQL Data Source.

1. File → new → website → empty website → name it → ok
2. Right click on website made → add new item → sql server database → name it → add → yes
3. Right click on table In server explorer → add new table → add columns → save the table
4. Right click on table made → show table data → add values
5. Right click on website → add new item → webform → name it
6. Go to design view → add form for login
7. Add sqldatasource → configure it
8. Write code

LoginModule.aspx.cs   LoginModule.aspx   dbo.Employee: T...MPLOYEEDEPT.MDF   dbo.Login: Table...TA\USERLOGIN.MDF

body

UserName   
 Password   
  
 SqlDataSource - SqlDataSource1  
 [lblResult]

Login: Query(vsit...TA\USERLOGIN.MDF)   LoginModule.aspx.cs   LoginModule.aspx

	UserName	Password
▶	Prachit	12345
*	NULL	NULL

**CODE:**

### LoginModule.aspx

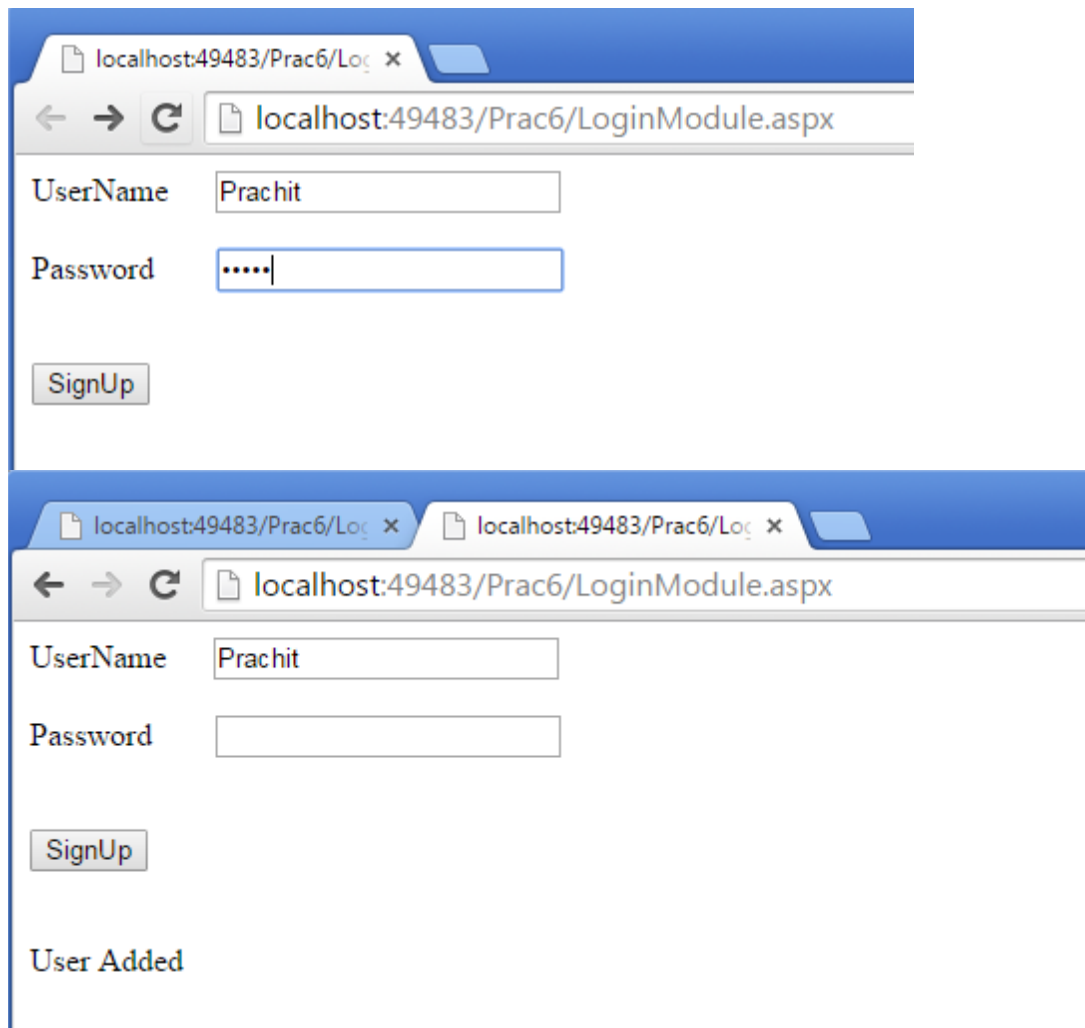
```

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

using System.Web.UI;
using System.Web.UI.WebControls;
public partial class LoginModule : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void btnSignUp_Click(object sender, EventArgs e)
    {
        SqlDataSource1.InsertParameters["Username"].DefaultValue = txtUserName.Text;
        SqlDataSource1.InsertParameters["Password"].DefaultValue = txtPassword.Text;
        SqlDataSource1.Insert();
        lblResult.Text = "User Added";
    }
}

```

**OUTPUT:**



**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 X +

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 X +

localhost:50309/WebForm1.aspx

ID

Name

Phone No

Add Update Delete

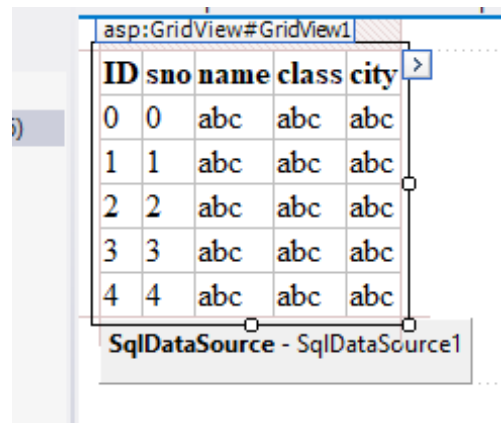
Connected To Server 4 ▾

Search

## Practical no.9

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

### **Grid\_view.aspx:-**



ID	sno	name	class	city
0	0	abc	abc	abc
1	1	abc	abc	abc
2	2	abc	abc	abc
3	3	abc	abc	abc
4	4	abc	abc	abc

SqlDataSource - SqlDataSource1

### **Grid\_view.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.Drawing;

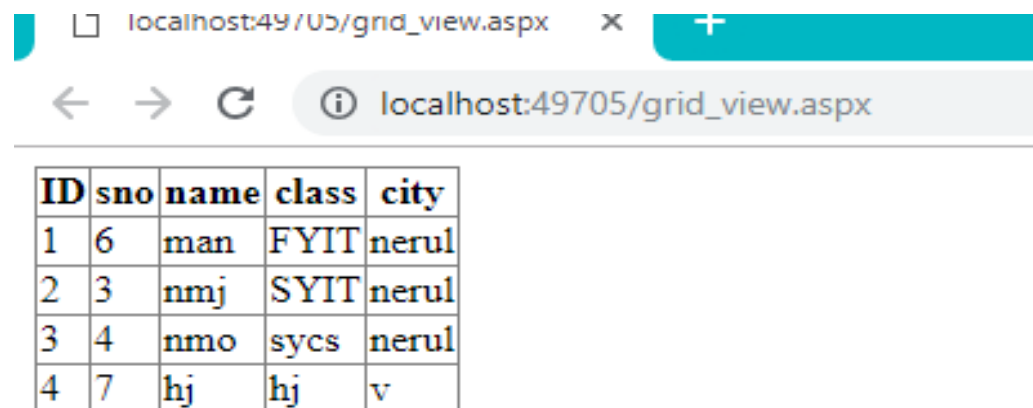
public partial class grid_view : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    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:-



The screenshot shows a web browser window with the address bar displaying 'localhost:49705/grid\_view.aspx'. Below the browser window, a table is displayed with the following data:

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

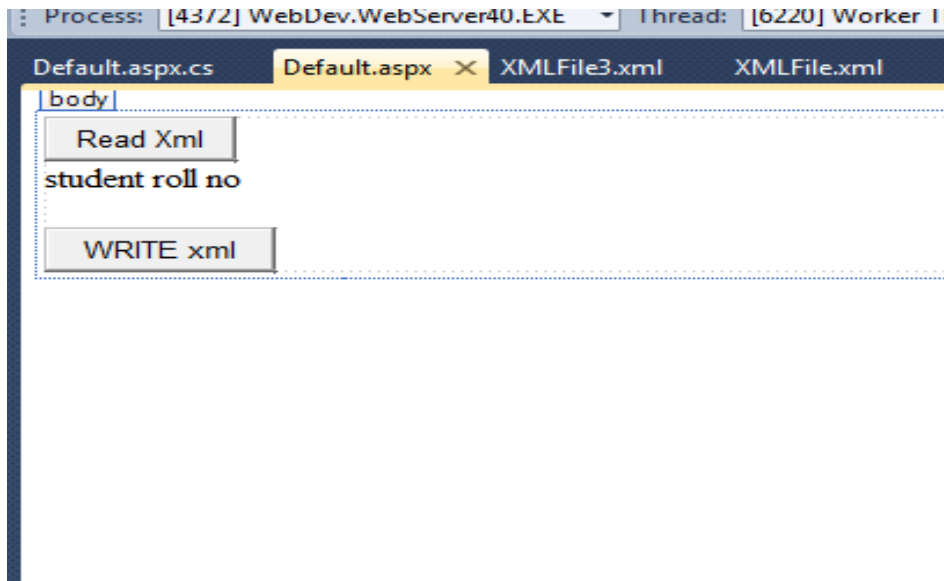
10 a) output:-



## Practical no 10

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

### Default.aspx:-



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

```
namespace ClassLibrary5
{
    public class Class1
    {
        public int add(int a, int b)
        {
            int c = a + b;
            return c;
        }
    }
}
```

#### **ConsoleApplication5.cs:-**

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

```
namespace ConsoleApplication5
{
    class Program
    {
        static void Main(string[] args)
        {
            ClassLibrary5.Class1 c = new ClassLibrary5.Class1();
            int t = c.add(1, 2);
            Console.WriteLine("addition={0}", t);
            Console.ReadKey();
        }
    }
}
```

```
}  
}  
}
```

**11) output:-**

