

## Practical No.: 1 a

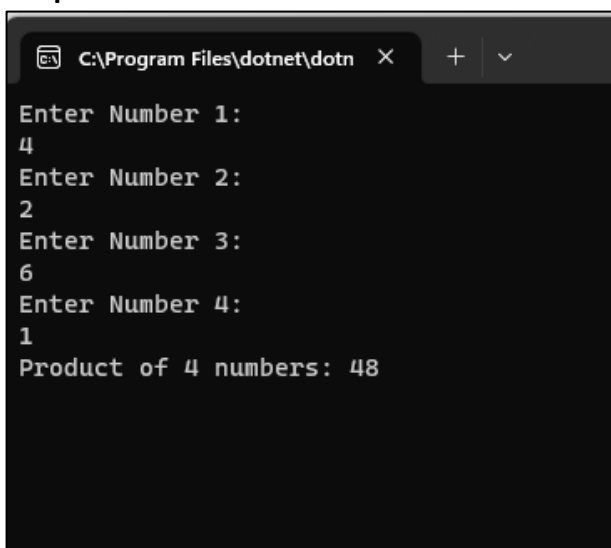
**Aim:** Create an application that obtains four int values from the user and displays the product.

**Code:**

Program.cs:

```
using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            int n1, n2, n3, n4, ans;
            Console.WriteLine("Enter Number 1: ");
            n1 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Number 2: ");
            n2 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Number 3: ");
            n3 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Number 4: ");
            n4 = Convert.ToInt32(Console.ReadLine());
            ans = n1 * n2 * n3 * n4;
            Console.WriteLine("Product of 4 numbers: " + ans);
            Console.ReadKey();
        }
    }
}
```

**Output:**



```
Enter Number 1:
4
Enter Number 2:
2
Enter Number 3:
6
Enter Number 4:
1
Product of 4 numbers: 48
```

## **Practical No.: 1 b**

**Aim:** Create an application to demonstrate string operations.

**Code:**

Program.cs:

```
using System;

namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            string s1 = "hello";
            Console.WriteLine("s1:" + s1);
            Console.WriteLine();

            Console.WriteLine("enter string:");
            string s2 = Console.ReadLine();
            Console.WriteLine("s2:" + s2);
            Console.WriteLine();

            string s3 = string.Copy(s1);
            Console.WriteLine("Copied String s3:" + s3);
            Console.WriteLine();

            string s4 = string.Concat(s1, s2);
            Console.WriteLine("Concatenated String s4:" + s4);
            Console.WriteLine();

            int nos = 789;
            string s5 = nos.ToString();
            Console.WriteLine("Converting num to String s5:" + s5);
            Console.WriteLine();

            string s6 = "lean";
            Console.WriteLine("O.G. String :" + s6);
            string s7 = s6.Insert(3, "r");
            Console.WriteLine("Insertion s7:" + s7);
            Console.WriteLine();

            int n = string.Compare(s1, s2);
            Console.WriteLine("Comparison n:" + n);
            Console.WriteLine();

            bool b1 = s1.Equals(s2);
            Console.WriteLine("Equals b1:" + b1);
```

```
Console.WriteLine();

bool b2 = string.Equals(s1, s2);
Console.WriteLine("Equals other way b2:" + b2);
Console.WriteLine();

bool b3 = (s1 == s2);
Console.WriteLine(" == b3:" + b3);
Console.WriteLine();

string x = "xyz";
Console.WriteLine("O.G. string :" + x);
int i = x.IndexOf('x');
Console.WriteLine("Index of x=" + i);
Console.WriteLine();

string y = "madam";
Console.WriteLine("O.G. string :" + y);
int j = y.LastIndexOf('a');
Console.WriteLine("Last index of a:" + j);
Console.WriteLine();

string z = "bye bye";
Console.WriteLine("O.G. string :" + z);
string p = z.Replace('e', 'y');
Console.WriteLine("Replaced String:" + p);
Console.WriteLine();

string q = "hello welcome";
Console.WriteLine("O.G. string :" + q);
string str1 = q.Remove(2);
string str2 = q.Remove(2, 4);
Console.WriteLine("Remove with single parameter:" + str1);
Console.WriteLine();
Console.WriteLine("Remove with two parameter:" + str2);
Console.WriteLine();

string s8 = "new york";
Console.WriteLine("O.G. string :" + s8);
string s9 = s8.Substring(5);
Console.WriteLine("Substring with single parameter i.e. starting point:" + s9);
Console.WriteLine();
string s10 = s8.Substring(1, 5);
Console.WriteLine("Substring with two parameter i.e. starting point and end point" + s10);
Console.WriteLine();

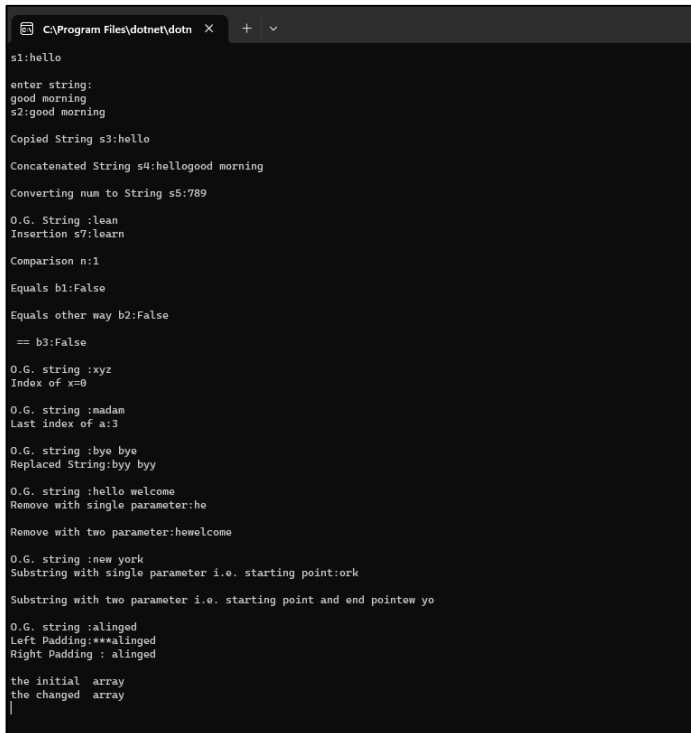
string s17 = "alinged";
Console.WriteLine("O.G. string :" + s17);
string s18 = s17.PadLeft(10, '*');
Console.WriteLine("Left Padding:" + s18);
string s19 = s17.PadRight(3, '*');
Console.WriteLine("Right Padding : " + s19);
Console.WriteLine();
```

```

        string strsrc = "changed";
        char[] dest = { 't', 'h', 'e', ' ', 'i', 'n', 'i', 't', 'i', 'a', 'l', ' ', ' ', ' ', 'a', 'r', 'r', 'a', 'y' };
        Console.WriteLine(dest);
        strsrc.CopyTo(0, dest, 4, strsrc.Length);
        Console.WriteLine(dest);
        Console.ReadKey();
    }
}
}

```

### Output:



```

C:\Program Files\dotnet\dotn X + v
s1:hello
enter string:
good morning
s2:good morning
Copied String s3:hello
Concatenated String s4:hellogood morning
Converting num to String s5:789
O.G. String :lean
Insertion s7:learn
Comparison n:1
Equals b1:False
Equals other way b2:False
== b3:False
O.G. string :xyz
Index of x=0
O.G. string :madam
Last index of a:3
O.G. string :bye bye
Replaced String:byy byy
O.G. string :hello welcome
Remove with single parameter:he
Remove with two parameter:hewelcome
O.G. string :new york
Substring with single parameter i.e. starting point:ork
Substring with two parameter i.e. starting point and end pointow yo
O.G. string :alinged
Left Padding:***alinged
Right Padding : alinged
the initial array
the changed array
|

```

## **Practical No.: 1 c**

**Aim:** 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.

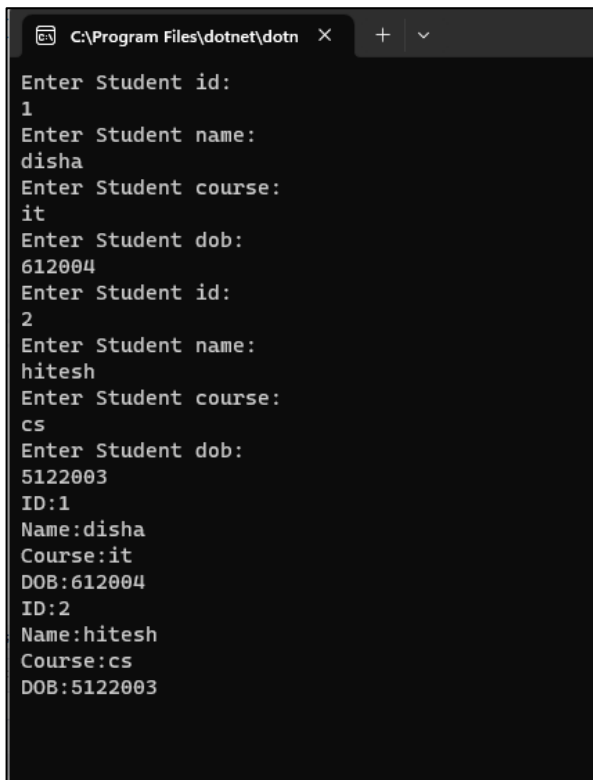
**Code:**

Program.cs:

```
using System;

namespace p1
{
    public struct stud
    {
        public int id;
        public string name;
        public string course;
        public int dob;
    };
    class Program
    {
        static void Main(string[] args)
        {
            stud[] s = new stud[2];
            for (int i = 0; i < 2; i++)
            {
                Console.WriteLine("Enter Student id:");
                s[i].id = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter Student name:");
                s[i].name = Console.ReadLine();
                Console.WriteLine("Enter Student course:");
                s[i].course = Console.ReadLine();
                Console.WriteLine("Enter Student dob:");
                s[i].dob = Convert.ToInt32(Console.ReadLine());
            }
            for (int i = 0; i < 2; i++)
            {
                Console.WriteLine("ID:" + s[i].id);
                Console.WriteLine("Name:" + s[i].name);
                Console.WriteLine("Course:" + s[i].course);
                Console.WriteLine("DOB:" + s[i].dob);
            }
            Console.ReadKey();
        }
    }
}
```

**Output:**



```
C:\Program Files\dotnet\dotn X + v
Enter Student id:
1
Enter Student name:
disha
Enter Student course:
it
Enter Student dob:
612004
Enter Student id:
2
Enter Student name:
hitesh
Enter Student course:
cs
Enter Student dob:
5122003
ID:1
Name:disha
Course:it
DOB:612004
ID:2
Name:hitesh
Course:cs
DOB:5122003
```

## **Practical No.: 1 d**

**Aim:** Create an application to demonstrate following operations.

- i) Generate Fibonacci series.
- ii) Test for prime numbers.
- iii) Test for vowels.
- iv) Use of foreach loop with arrays.
- v) Reverse a number and find sum of digits of a number.

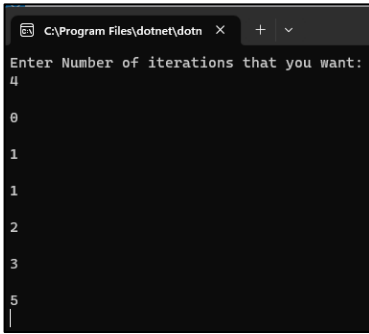
### **i. Fibonacci Series:**

**Code:**

Program.cs:

```
using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            int ni, f1, f2, fib;
            Console.WriteLine("Enter Number of iterations that you want: ");
            ni = int.Parse(Console.ReadLine());
            f1 = 0;
            f2 = 1;
            Console.WriteLine("\n" + f1);
            Console.WriteLine("\n" + f2);
            for(int i=1;i<=ni;i++)
            {
                fib = f1 + f2;
                f1 = f2;
                f2 = fib;
                Console.WriteLine("\n" + fib);
            }
            Console.ReadKey();
        }
    }
}
```

**Output:**



```

C:\Program Files\dotnet\dotn x + v
Enter Number of iterations that you want:
4
0
1
1
2
3
5
|

```

## ii. Prime Numbers:

### Code:

#### Program.cs:

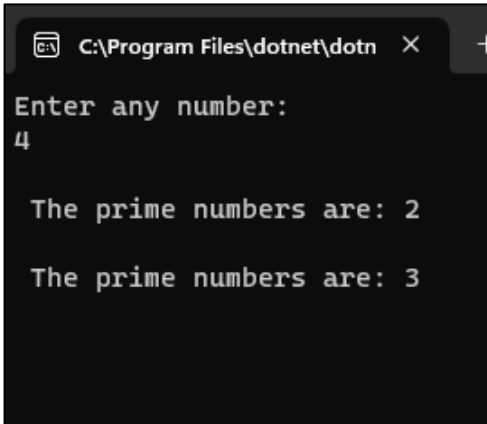
```

using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, no, j;
            bool flag = true;
            Console.WriteLine("Enter any number: ");
            no = int.Parse(Console.ReadLine());
            for (i = 2; i < no; i++)
            {
                for (j = 2; j < no; j++)
                {
                    if (i != j && i % j == 0)
                    {
                        flag = false;
                        break;
                    }
                }
                if (flag == true)
                {
                    Console.WriteLine("\n The prime numbers are: " + i);
                }
                flag = true;
            }
            Console.ReadKey();
        }
    }
}

```

### Output:





A screenshot of a console application window titled 'C:\Program Files\dotnet\dotn'. The window shows the following text: 'Enter any number:', '4', 'The prime numbers are: 2', and 'The prime numbers are: 3'.

### iii. Vowels:

#### Code:

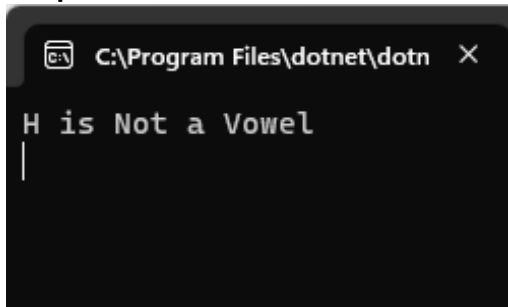
#### Program.cs:

```
using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            string s = "Hi GOOD Morning";
            Char[] letter = s.ToCharArray();
            foreach (char c in letter)
            {
                string s1 = Vowel(c);
                Console.WriteLine("{0} is {1}", c, s1);
                Console.ReadKey();
            }
        }
        public static string Vowel(char a)
        {
            switch (a)
            {
                case 'a':
                case 'A':
                case 'e':
                case 'E':
                case 'i':
                case 'I':
                case 'o':
                case 'O':
                case 'u':
                case 'U': return ("Vowel");
                default:
                    return ("Not a Vowel");
            }
        }
    }
}
```

```

    }
}
}
}

```

**Output:**


A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Program Files\dotnet\dotn' and a close button. The console output displays the text 'H is Not a Vowel' on a single line, with a cursor positioned at the end of the line.

**iv. Foreach loop with arrays:****Code:**Program.cs:

```

using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {

            String[] item =
            {
                "TEJAS","SUSHANT","RAHUL","HARIKESH","PRADEEP","MAG" };
            foreach(string i in item)
            {
                Console.WriteLine("\n" + i);
            }
            Console.ReadKey();
        }
    }
}

```

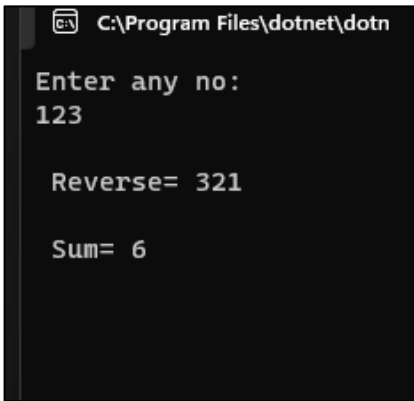
**Output:**

**v. Reverse:****Code:**Program.cs:

```
using System;
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {

            int no, r, rev, sum;
            Console.WriteLine("Enter any no: ");
            no = int.Parse(Console.ReadLine());
            rev = 0;
            sum = 0;
            while (no != 0)
            {
                r = no % 10;
                rev = (rev * 10) + r;
                sum = sum + r;
                no = no / 10;
            }
            Console.WriteLine("\n Reverse= " + rev);
            Console.WriteLine("\n Sum= " + sum);
            Console.ReadKey();
        }
    }
}
```

**Output:**

A screenshot of a Windows command prompt window. The title bar at the top shows the file path "C:\Program Files\dotnet\dotn". The command prompt displays the following text: "Enter any no:", followed by the user input "123". Below the input, it shows "Reverse= 321" and "Sum= 6".

```
C:\Program Files\dotnet\dotn  
Enter any no:  
123  
  
Reverse= 321  
  
Sum= 6
```

## Practical No.: 2 a

**Aim:** Create simple application to perform following operations.

- i) Finding factorial Value.
- ii) Money Conversion.
- iii) Quadratic Equation.
- iv) Temperature Conversion.

### **i. Factorial Value:**

**Code:**

Program.cs:

```
using System;

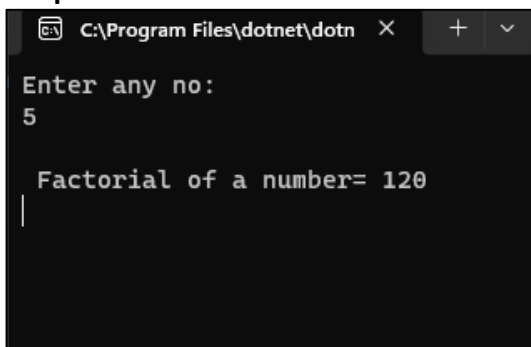
namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {

            int no, fact = 1;
            Console.WriteLine("Enter any no: ");
            no = int.Parse(Console.ReadLine());

            for(int i=1;i<=no;i++)
            {
                fact = fact * i;
            }
            Console.WriteLine("\n Factorial of a number= " + fact);
            Console.ReadKey();

        }
    }
}
```

**Output:**

A screenshot of a Windows console application window. The title bar shows the file path 'C:\Program Files\dotnet\dotn' and standard window controls. The console output shows the prompt 'Enter any no:' followed by the user input '5'. Below that, it displays 'Factorial of a number= 120' with a cursor on the next line.

```
C:\Program Files\dotnet\dotn X + v
Enter any no:
5
Factorial of a number= 120
|
```

**ii. Money Conversion:****Code:**Program.cs:

```
using System;

namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            string ans = " ";
            do
            {
                Console.WriteLine("Enter Currency in Rs: ");
                int r = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Select your choice to convert into 1.Dollars 2.Euro 3.Pounds");
                int n = Convert.ToInt32(Console.ReadLine());
                double d = 0;
                Console.WriteLine("Currency Converter");
                switch (n)
                {
                    case 1:
                        d = r / 68.5;
                        Console.WriteLine(r + "Rs.= " + d + "$");
                        break;
                    case 2:
                        d = r / 89.48;
                        Console.WriteLine(r + "Rs.= " + d + "pounds");
                        break;
                    case 3:
                        d = r / 79.64;
                        Console.WriteLine(r + "Rs.= " + d + "Euro");
                        break;
                    default:
                        Console.WriteLine("Invalid choice");
                        break;
                }
                Console.WriteLine("Do you want to continue?");
                ans = Console.ReadLine();
            }
            while (ans != "n");
            Console.ReadKey();
        }
    }
}
```

**Output:**

```

C:\Program Files\dotnet\dotn X + v
Enter Currency in Rs:
69
Select your choice to convert into 1.Dollars 2.Euro 3.Pounds
1
Currency Converter
69Rs.= 1.00729927007299$
Do you want to continue?
y
Enter Currency in Rs:
90
Select your choice to convert into 1.Dollars 2.Euro 3.Pounds
2
Currency Converter
90Rs.= 1.00581135449262pounds
Do you want to continue?
y
Enter Currency in Rs:
80
Select your choice to convert into 1.Dollars 2.Euro 3.Pounds
3
Currency Converter
80Rs.= 1.00452034153692Euro
Do you want to continue?

```

**iii. Quadratic Equations:****Code:**Program.cs:

```
using System;
```

```
namespace p1
```

```
{
```

```
class Program
```

```
{
```

```
    public static void SolveQuadratic(int a, int b, int c)
```

```
    {
```

```
        double sqrtpart = b * b - 4 * a * c;
```

```
        double x, x1, x2, img;
```

```
        if (sqrtpart > 0)
```

```
        {
```

```
            x1 = (-b + System.Math.Sqrt(sqrtpart)) / (2 * a);
```

```
            x2 = (-b - System.Math.Sqrt(sqrtpart)) / (2 * a);
```

```
            Console.WriteLine("Two Real Solutions: {0,8:f4} or {1,8:f4}", x1, x2);
```

```
        }
```

```
        else if (sqrtpart < 0)
```

```
        {
```

```
            sqrtpart = -sqrtpart;
```

```
            x = -b / (2 * a);
```

```
            img = System.Math.Sqrt(sqrtpart) / (2 * a);
```

```
            Console.WriteLine("Two Imaginary Solutions: {0,8:f4} + {1,8:f4} i or {2,8:f4} + {3,8:f4} i", x, img, x, img);
```

```
        }
```

```
        else
```

```
        {
```

```

        x = (-b + System.Math.Sqrt(sqrtpart)) / (2 * a);
        Console.WriteLine("One Real Solution: {0,8:f4}", x);
    }

}

static void Main(string[] args)
{

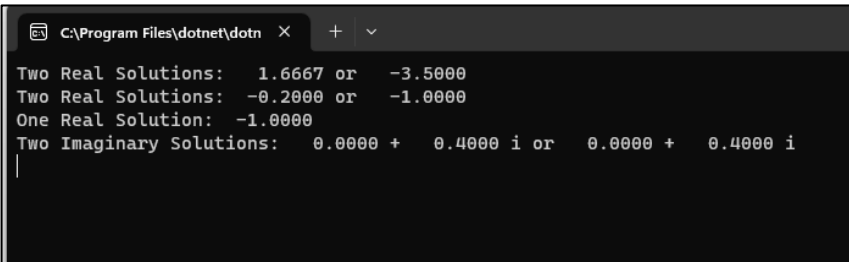
    // 6x^2 + 11x - 35 = 0
    SolveQuadratic(6, 11, -35);

    // 5x^2 + 6x + 1 = 0
    SolveQuadratic(5, 6, 1);

    // 2x^2 + 4x + 2 = 0
    SolveQuadratic(2, 4, 2);

    // 5x^2 + 2x + 1 = 0
    SolveQuadratic(5, 2, 1);
    Console.ReadKey();
}
}
}

```

**Output:**


```

C:\Program Files\dotnet\dotn  X  +  v
Two Real Solutions:  1.6667 or  -3.5000
Two Real Solutions: -0.2000 or  -1.0000
One Real Solution:  -1.0000
Two Imaginary Solutions:  0.0000 +  0.4000 i or  0.0000 +  0.4000 i
|

```

**iv. Temperature Conversion:****Code:**Program.cs:

```

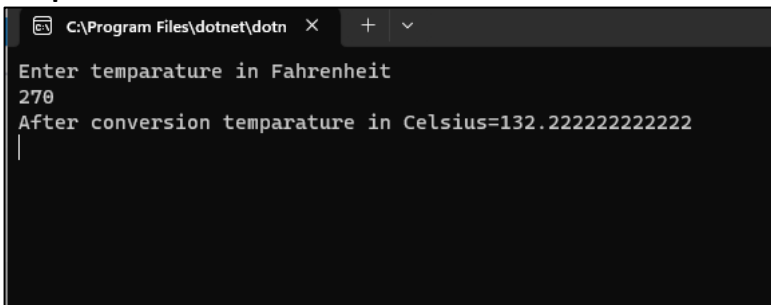
using System;

namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter temperature in Fahrenheit ");
            double F, C;
            F = Convert.ToDouble(Console.ReadLine());
            C = (F - 32) * 5 / 9;
            Console.WriteLine("After conversion temperature in Celsius={0}", C);
        }
    }
}

```



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

**Output:**A screenshot of a Windows console application window. The title bar shows the file path 'C:\Program Files\dotnet\dotn' and standard window controls. The console output is as follows:  
Enter temparature in Fahrenheit  
270  
After conversion temparature in Celsius=132.222222222222  
The cursor is positioned on the line following the output.

```
C:\Program Files\dotnet\dotn X + v  
Enter temparature in Fahrenheit  
270  
After conversion temparature in Celsius=132.222222222222  
|
```

## **Practical No.: 2 b**

**Aim:** Create simple application to perform following operations.

- i) Function Overloading.
- ii) Inheritance (all types).
- iii) Constructor Overloading.
- iv) Interfaces.

### **i. Function Overloading:**

**Code:**

Program.cs:

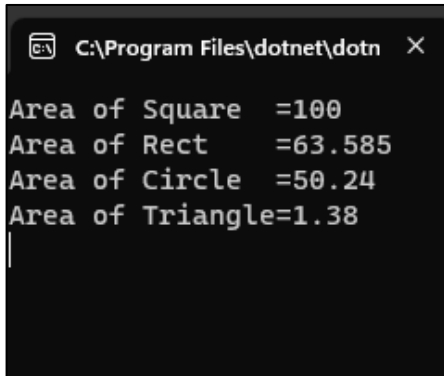
```
using System;

namespace p1
{
    class Program
    {
        public float Area(float x)
        {
            return (x * x);
        }
        public int Area(int l, int h)
        {
            return (l * h);
        }
        public double Area(double r)
        {
            return (3.14 * r * r);
        }
        public float Area(float len, float hyt)
        {
            return (0.5F * len * hyt);
        }
        static void Main(string[] args)
        {
            Program p = new Program();
            Console.WriteLine("Area of Square =" + p.Area(10));
            Console.WriteLine("Area of Rect  =" + p.Area(4.5));
            Console.WriteLine("Area of Circle =" + p.Area(4.0));
            Console.WriteLine("Area of Triangle=" + p.Area(1.2F, 2.3F));
            Console.ReadKey();
        }
    }
}
```

```

    }
}

```

**Output:**


```

C:\Program Files\dotnet\dotn
Area of Square =100
Area of Rect =63.585
Area of Circle =50.24
Area of Triangle=1.38

```

**ii. Inheritance & iii. Constructor Overloading:****Code:**Program.cs:

```

using System;

namespace p1
{
    class A
    {
        public int a;
    public A(int x)
        {
            a = x;
        }

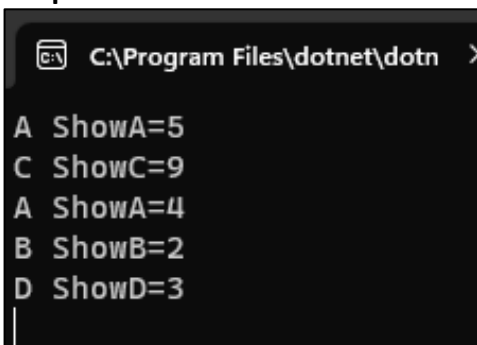
        public void ShowA()
        {
            Console.WriteLine("A ShowA=" + a);
        }
    }
    class B : A
    {
    int v;
    public B(int z, int y)
        : base(z)
        {
            v = ;y
        }
    public void ShowB()
    {

        Console.WriteLine("B ShowB=" + v);
    }
}

```

```
class C : A
{
    int n;
    public C(int x, int m)
        : base(x)
    {
        n = m;
    }
    public void ShowC()
    {
        Console.WriteLine("C ShowC=" + n);
    }
}
class D : B
{
    int d;
    public D(int x, int y, int z)
        : base(x, y)
    {
        d = z;
    }
    public void ShowD()
    {
        Console.WriteLine("D ShowD=" + d);
    }
}
class Program
{
    static void Main(string[] args)
    {
        C c1 = new C(5, 9);
        D d1 = new D(4, 2, 3);
        c1.ShowA();
        c1.ShowC();
        d1.ShowA();
        d1.ShowB();
        d1.ShowD();
        Console.ReadKey();
    }
}
```

**Output:**



```
C:\Program Files\dotnet\dotn >
A ShowA=5
C ShowC=9
A ShowA=4
B ShowB=2
D ShowD=3
|
```

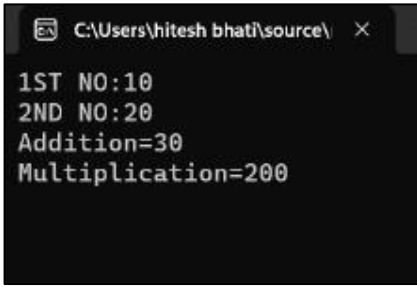
**iv. Interfaces:****Code:**Program.cs:

```

interface Addition
{
    int Add();
}
interface Multiplication
{
    int Mul();
}
class A : Addition, Multiplication
{
    public int x, y;
    public A(int a, int b)
    {
        x = a;
        y = b;
    }
    Console.WriteLine("1ST NO:" + x);
    Console.WriteLine("2ND NO:" + y);
}
public int Add()
{
    return (x + y);
}
public int Mul()
{
    return (x * y);
}
}
class MultipleInterface
{
    static void Main(string[] args)
    {
        A a1 = new A(10, 20);
        Addition obj = (Addition)a1;
        Multiplication obj1 = (Multiplication)a1;
        Console.WriteLine("Addition=" + obj.Add());
        Console.WriteLine("Multiplication=" + obj1.Mul());
        Console.ReadKey();
    }
}

```

**Output:**



```
C:\Users\hitesh bhat\source\
1ST NO:10
2ND NO:20
Addition=30
Multiplication=200
```

## Practical No.: 2 c

**Aim:** Create simple application to perform following operations.

- i) Using Delegates and Events.
- ii) Exception Handling.

### **i. Delegates and Events:**

**Code:**

Program.cs:

```
using System;

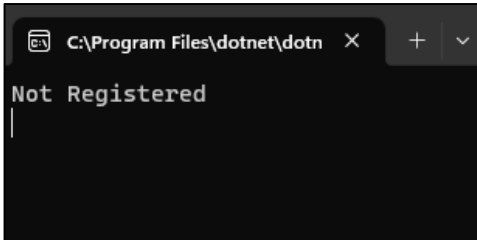
namespace p1
{
    public delegate void EventHandler(string a);
    public class Operation
    {
        public event EventHandler xyz;
        public void Action(string a)
        {
            if (xyz != null)
            {
                xyz(a);
                Console.WriteLine(a);
            }
            else
            {
                Console.WriteLine("Not Registered");
            }
        }
    }
}

class Program
{
    public static void CatchEvent(string s)
    {
        Console.WriteLine("Method Calling");
    }
    static void Main(string[] args)
    {
        Operation o = new Operation();
```

```

        o.Action("Event Calling");
        o.xyz += new EventHandler(CatchEvent);
        Console.ReadLine();
    }
}
}

```

**Output:****iv. Exception Handling:****Code:**Program.cs:

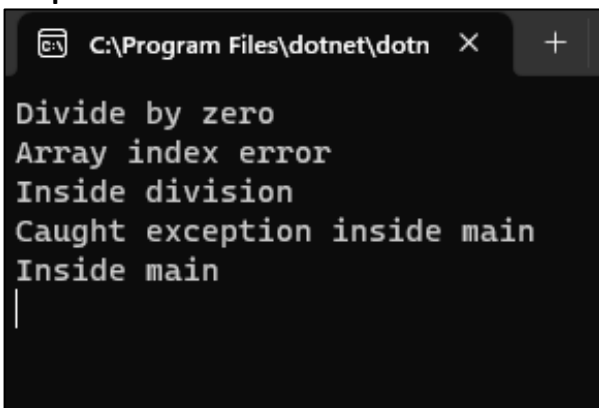
```

using System;

namespace p1
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 10, b = 5, c = 5;
            int x, y;
            try
            {
                x = a / (b - c);
            }
            catch (Exception e)
            {
                Console.WriteLine("Divide by zero");
            }
            y = a / (b + c);
            int[] d = { 5, 10 };
            int f = 5;
            try
            {
                int g = d[2] / f - d[1];
            }
            catch (ArithmeticException e)
            {
                Console.WriteLine("Divide by zero");
            }
            catch (IndexOutOfRangeException e)
            {
                Console.WriteLine("Array index error");
            }
        }
    }
}

```

```
    }  
    try  
    {  
        Division();  
    }  
    catch (DivideByZeroException e)  
    {  
        Console.WriteLine("Caught exception inside main");  
    }  
    finally  
    {  
        Console.WriteLine("Inside main");  
    }  
    Console.ReadKey();  
}  
static int m = 10;  
static int n = 0;  
static void Division()  
{  
    try  
    {  
        int k = m / n;  
    }  
    catch (ArgumentException e)  
    {  
        Console.WriteLine("Caught exception inside main");  
    }  
    finally  
    {  
        Console.WriteLine("Inside division");  
    }  
}  
}  
}
```

**Output:**

```
C:\Program Files\dotnet\dotn X +  
Divide by zero  
Array index error  
Inside division  
Caught exception inside main  
Inside main  
|
```



### **Practical No.: 3**

**Aim:** Create a simple web page with various sever controls to demonstrate setting and use of their properties and use of validation controls.

**Code:**

Login.aspx.cs:

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

namespace Pract_3a
{
    public partial class Login : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btn_login_Click(object sender, EventArgs e)
        {
            if (txt_name.Text == "Disha" && txt_pwd.Text == "123") Response.Redirect("Registration form.aspx");
            else
                Label1.Text = "Enter valid information";
        }

        protected void btn_reset_Click(object sender, EventArgs e)
        {
            txt_name.Text = "";
            txt_pwd.Text = "";
            txt_confpwd.Text = "";
        }
    }
}
```

RegistrationForm.aspx.cs:

```

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

namespace Pract_3a
{
    public partial class Registration_form : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        protected void btn_register_Click(object sender, EventArgs e)
        {
            string g, c = "";
            string[] a = new string[4];
            if (RadioButton1.Checked == true)
                g = RadioButton1.Text;
            else
                g = RadioButton2.Text;
            if (DropDownList1.SelectedValue == "IT")
                c = "25,500";
            else if (DropDownList1.SelectedValue == "CS")
                c = "29,000";
            else if (DropDownList1.SelectedValue == "DS")
                c = "22,200";
            Response.Redirect("Receipt.aspx?Name=" + text_name.Text + "&Gender=" + g + "&Email=" + txt_email.Text +
"&Course=" + DropDownList1.SelectedValue + "&Fees=" + c);
        }
    }
}

```

Receipt.aspx.cs:

```

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

namespace Pract_3a

```

```

{
    public partial class Receipt1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            lblName.Text = Request.QueryString["Name"];
            lblGender.Text = Request.QueryString["Gender"];
            lblEmail.Text = Request.QueryString["Email"];
            lblCourse.Text = Request.QueryString["Course"];
            lblFees.Text = Request.QueryString["Fees"];
        }
    }
}

```

### FeesDetails.cs:

```

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

namespace Pract_3a
{
    public partial class Fees_Details : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Home.aspx");
        }
    }
}

```

### Output:

The output shows two web pages from a local development environment. The first page, titled 'LOGIN', has a header 'LOGIN' and three input fields: 'Username' with the value 'Disha', 'Password' (masked with dots), and 'Confirm Password' (masked with dots). There are 'Login' and 'Reset' buttons at the bottom. The second page, titled 'Registration Form', has a header 'Registration Form' and several input fields: 'Name' (Disha), 'Age' (22), 'Gender' (radio buttons for Male and Female, with Female selected), 'Email' (disha12@gma.com), and 'Course' (a dropdown menu showing 'IT'). A 'Register' button is at the bottom right.



## Practical No.: 4

**Aim:** Demonstrate the use of Calendar control to perform following operations.

- Display messages in a calendar control.
- Display vacation in a calendar control.
- Selected day in a calendar control using style.
- Difference between two calendar dates.

**Code:**

Calendar.aspx.cs:

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

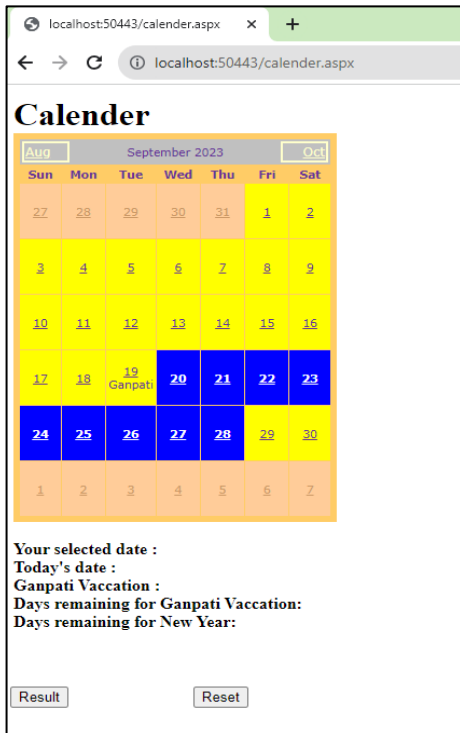
```
namespace Pract_3_B
{
    public partial class Calender : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Calendar1_SelectionChanged(object sender, EventArgs e)
        {
            Label1.Text = "Your selected date" + Calendar1.SelectedDate.Date.ToString();
        }
        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 == 19 && e.Day.Date.Month == 9)
    {
        Calendar1.SelectedDate = new DateTime(2023, 9, 18);
        Calendar1.SelectedDates.SelectRange(Calendar1.SelectedDate, Calendar1.SelectedDate.AddDays(10));
        Label lbl1 = new Label();
        lbl1.Text = "<br>Ganpati";
        e.Cell.Controls.Add(lbl1);
    }
}
protected void Button1_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: 9-19-2023";
    TimeSpan d = new DateTime(2023, 9, 19) - DateTime.Now;
    Label4.Text = "Days Remaining for Ganpati Vacation: " + d.Days.ToString();
    TimeSpan d1 = new DateTime(2023, 12, 31) - DateTime.Now;
    Label5.Text = "Days Remaining for new year:" + d1.Days.ToString();
    if (Calendar1.SelectedDate.ToShortDateString() == "9-19-2023")
        Label3.Text = "<b>Ganpati Festival Start</b>";
    else if (Calendar1.SelectedDate.ToShortDateString() == "9-24-2023")
        Label3.Text = "<b>Ganpati Festival End</b>";
}
protected void Button2_Click(object sender, EventArgs e)
{
    Label1.Text = "";
    Label2.Text = "";
    Label3.Text = "";
    Label4.Text = "";
    Label5.Text = "";
    Calendar1.SelectedDates.Clear();
}
}
}

```

**Output:**



## Practical No.: 5

**Aim:** Demonstrate the use of Treeview control perform following operations.

- Treeview control and datalist.
- Treeview operations.

**Code:**

Default.aspx.cs:

```
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 Pract_3c
{
    public partial class WebForm1 : 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();
    }
}

protected void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
{
}
}
}

```

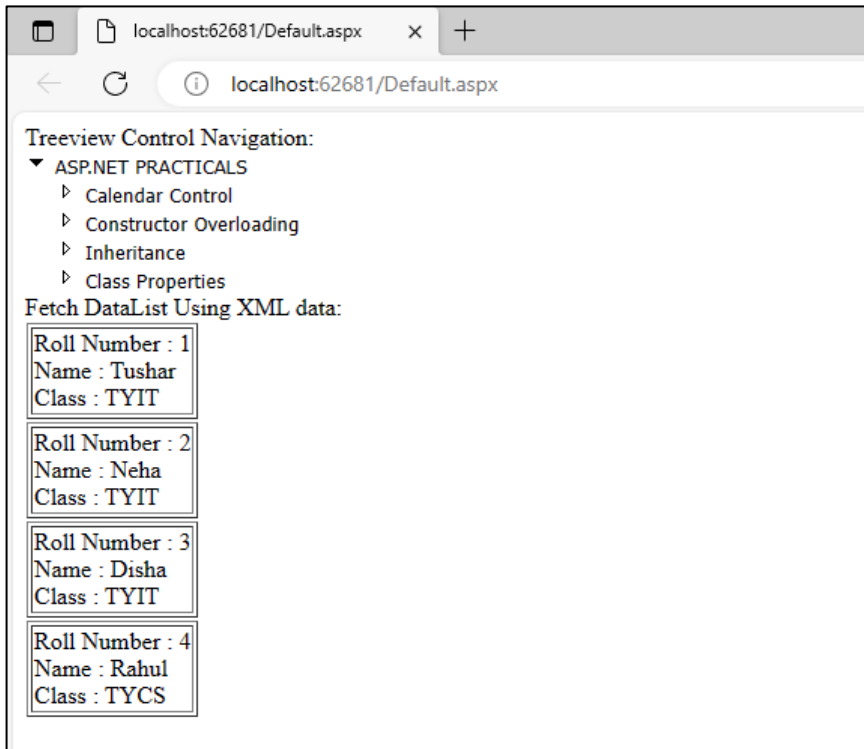
stdetail.xml:

```

<?xml version="1.0" encoding="utf-8" ?>
<Studentdetail>
  <student>
    <sid>1</sid>
    <sname>Tushar</sname>
    <sclass>TYIT</sclass>
  </student>
  <student>
    <sid>2</sid>
    <sname>Neha</sname>
    <sclass>TYIT</sclass>
  </student>
  <student>
    <sid>3</sid>
    <sname>Disha</sname>
    <sclass>TYIT</sclass>
  </student>
  <student>
    <sid>4</sid>
    <sname>Rahul</sname>
    <sclass>TYCS</sclass>
  </student>
</Studentdetail>

```

**Output:**





## Practical No.: 6

**Aim:** Create Web Form to demonstrate use of Adrotator Control.

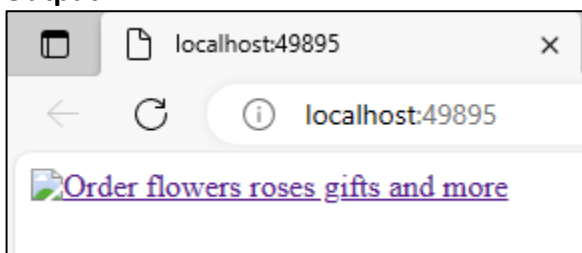
**Code:**

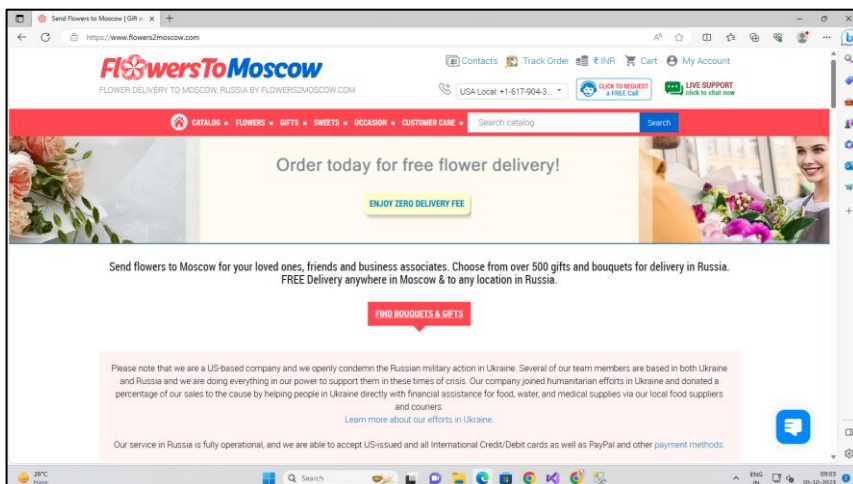
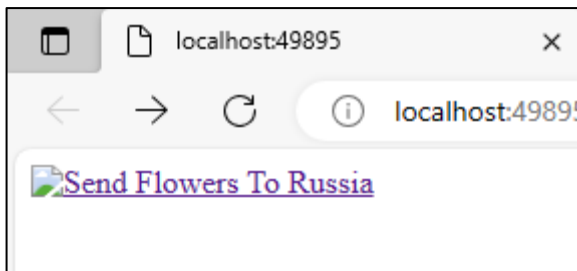
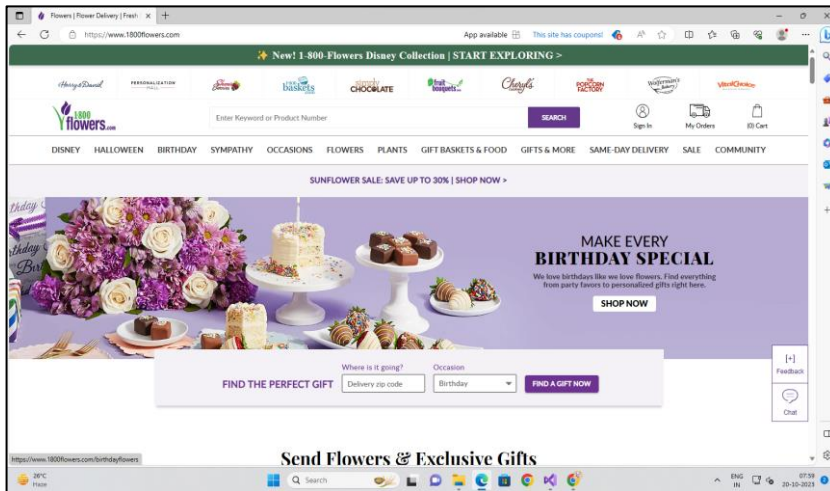
.cs file:

ADfile.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
  <Ad>
    <ImageUrl>D:\18\pract4A\images\rose1.jpg</ImageUrl>
    <NavigateUrl>http://www.1800Flowers.com</NavigateUrl>
    <AlternateText>Order flowers roses gifts and more</AlternateText>
    <Impressions>20</Impressions>
    <Keyword>Flower</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>D:\18\pract4A\images\rose2.jpg</ImageUrl>
    <NavigateUrl></NavigateUrl>
    <AlternateText>Order flowers roses and flowers</AlternateText>
    <Impressions>20</Impressions>
    <Keyword>Gifts</Keyword>
  </Ad>
  <Ad>
    <ImageUrl>D:\18\pract4A\images\rose3.jpg</ImageUrl>
    <NavigateUrl>http://www.flowers2moscow.com</NavigateUrl>
    <AlternateText>Send Flowers To Russia</AlternateText>
    <Impressions>20</Impressions>
    <Keyword>Russia</Keyword>
  </Ad>
</Advertisements>
```

**Output:**





## Practical No.: 7

**Aim:** Create Web Form to demonstrate use User Controls.

**Code:**

MyUserControl.ascx.cs:

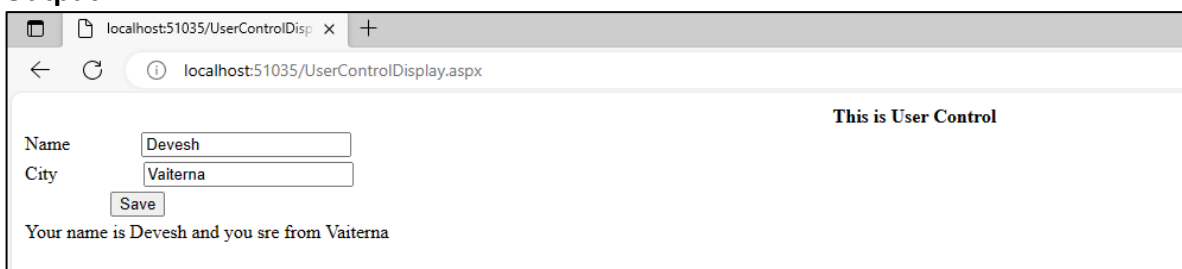
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace pract4B
{
    public partial class MyUserControl : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Text = "Your name is " + TextBox1.Text + " and you sre from " + TextBox2.Text;
        }
    }
}
```

**Output:**



The screenshot shows a web browser window with the address bar displaying 'localhost:51035/UserControlDisplay.aspx'. The page content includes a header 'This is User Control' on the right. On the left, there is a form with two text boxes: 'Name' containing 'Devesh' and 'City' containing 'Vaiterna'. Below these is a 'Save' button. At the bottom, a label displays the text 'Your name is Devesh and you sre from Vaiterna'.

## Practical No.: 8

**Aim:** Create a web application bind data in a multiline textbox by querying in another textbox.

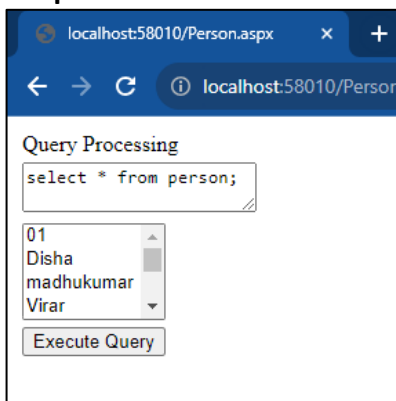
**Code:**

Person.aspx.cs:

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

namespace p6_7
{
    public partial class Person : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {}
        protected void Button1_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=tyit001;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand(textBox1.Text, conn);
            SqlDataReader dr = cmd.ExecuteReader();
            listBox1.Items.Clear();
            while (dr.Read())
            {
                for (int i = 0; i <= dr.FieldCount - 1; i++)
                {
                    listBox1.Items.Add(dr[i].ToString());
                }
            }
        }
    }
}
```

**Output:**



## Practical No.: 9

**Aim:** Create a web application to display records by using database.

**Code:**

Default.aspx.cs:

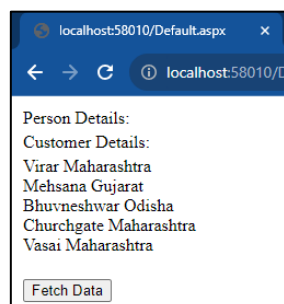
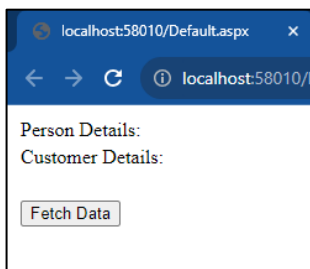
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

namespace p6_7
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=tyit001;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand("select city, state from person", conn);
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                Label1.Text += dr["city"].ToString() + " " + dr["state"].ToString() + "<br>";
            }
            dr.Close();
            conn.Close();
        }
    }
}
```

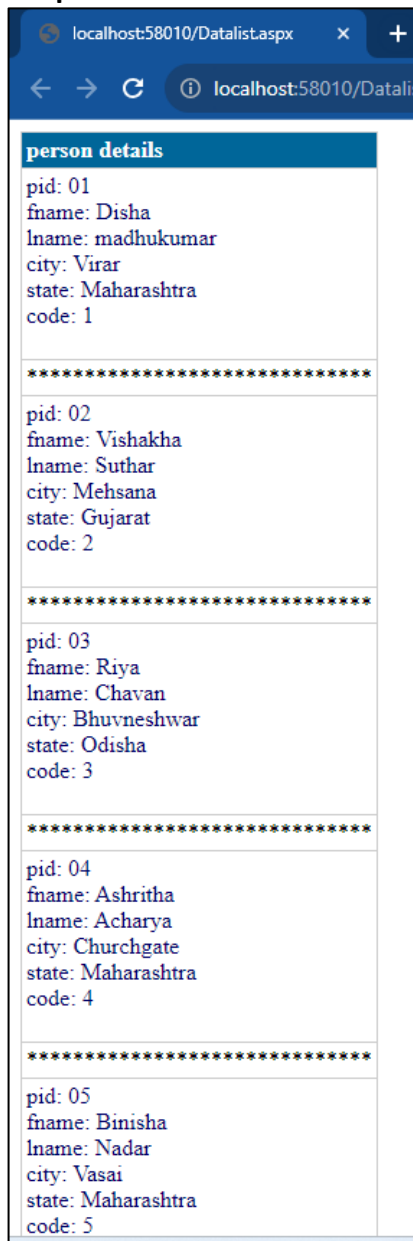
**Output:**



## Practical No.: 10

**Aim:** Demonstrate the use of Datalist link control.

**Output:**



The screenshot shows a web browser window with the address bar displaying 'localhost:58010/Datalist.aspx'. The page content is titled 'person details' in a blue header. Below the header, there are five rows of data, each separated by a line of asterisks. Each row contains the following fields: pid, fname, lname, city, state, and code.

person details					
pid: 01	fname: Disha	lname: madhukumar	city: Virar	state: Maharashtra	code: 1
*****					
pid: 02	fname: Vishakha	lname: Suthar	city: Mehsana	state: Gujarat	code: 2
*****					
pid: 03	fname: Riya	lname: Chavan	city: Bhuvneshwar	state: Odisha	code: 3
*****					
pid: 04	fname: Ashritha	lname: Acharya	city: Churchgate	state: Maharashtra	code: 4
*****					
pid: 05	fname: Binisha	lname: Nadar	city: Vasai	state: Maharashtra	code: 5

## Practical No.: 11

**Aim:** Create a web application to display the phone number of an author using database.

**Code:**

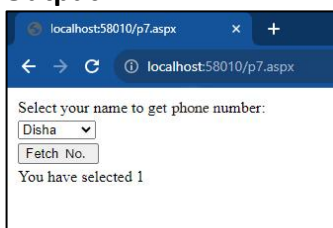
Pract.aspx.cs:

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

namespace p6_7
{
    public partial class p7 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if(IsPostBack==false)
            {
                SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=tyit001;Integrated Security=True");
                conn.Open();
                SqlCommand cmd = new SqlCommand("select fname, phone from person", conn);
                SqlDataReader dr = cmd.ExecuteReader();
                DropDownList1.DataSource = dr;
                DropDownList1.DataTextField = "fname";
                DropDownList1.DataValueField = "phone";
                DropDownList1.DataBind();
                dr.Close();
                conn.Close();
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Text = "You have selected " + DropDownList1.SelectedValue;
        }
    }
}
```

**Output:**



## Practical No.: 12

**Aim:** Create a web application for inserting and deleting and updating record from a database (Using Execute-Non Query) and display using GridView control in it show update delete sorting paging and selection.

**Code:**

Pract.aspx.cs:

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

namespace pract7
{
    public partial class _7c : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=tyit18;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand("update Person set Fname=@Fname,Lname=@Lname, City=@City,
Code=@Code where PID=@PID", conn);
            cmd.Parameters.AddWithValue("@PID", TextBox1.Text);
            cmd.Parameters.AddWithValue("@Fname", TextBox2.Text);
            cmd.Parameters.AddWithValue("@Lname", TextBox3.Text);
            cmd.Parameters.AddWithValue("@City", TextBox4.Text);
            cmd.Parameters.AddWithValue("@Code", TextBox5.Text);
            cmd.ExecuteNonQuery();
            Label1.Text = "record updated succesfully";
            conn.Close();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {

            SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=tyit18;Integrated Security=True");
            conn.Open();
            SqlCommand cmd = new SqlCommand("insert into Person values(@PID,@Fname,@Lname,@City,@Code)",
conn);
```



```

cmd.Parameters.AddWithValue("@PID", TextBox1.Text);
cmd.Parameters.AddWithValue("@Fname", TextBox2.Text);
cmd.Parameters.AddWithValue("@Lname", TextBox3.Text);
cmd.Parameters.AddWithValue("@City", TextBox4.Text);
cmd.Parameters.AddWithValue("@Code", TextBox5.Text);
cmd.ExecuteNonQuery();
Label1.Text = "record inserted succesfully";
conn.Close();
}

```

```

protected void Button3_Click(object sender, EventArgs e)
{
    TextBox1.Text = " ";
    TextBox2.Text = " ";
    TextBox3.Text = " ";
    TextBox4.Text = " ";
    TextBox5.Text = " ";
}

```

```

protected void Button4_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=tyit18;Integrated Security=True");
    conn.Open();
    SqlCommand cmd = new SqlCommand("delete from Person where PID=@PID", conn);
    cmd.Parameters.AddWithValue("@PID", TextBox1.Text);
    cmd.ExecuteNonQuery();
    Label1.Text = "record deleted succesfully";
    conn.Close();
}

```

```

protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    TextBox1.Text = GridView1.SelectedRow.Cells[4].Text;
    TextBox2.Text = GridView1.SelectedRow.Cells[5].Text;
    TextBox3.Text = GridView1.SelectedRow.Cells[6].Text;
    TextBox4.Text = GridView1.SelectedRow.Cells[7].Text;
    TextBox5.Text = GridView1.SelectedRow.Cells[8].Text;
}

```

```

protected void SqlDataSource1_Selecting(object sender, SqlDataSourceSelectingEventArgs e)
{
}
}
}

```

### Output:

localhost:55588/7c.aspx

← → ↻ ⓘ localhost:55588/7c.aspx

enter person id :  
 enter first name:  
 enter last name:  
 enter city:  
 enter code:

Label

			PID	Fname	Lname	City	Code
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>		dev	da	virar	P015
<a href="#">Update</a>	<a href="#">Cancel</a>		12	<input type="text" value="monkey"/>	<input type="text" value="luffy"/>	<input type="text" value="onepeice"/>	<input type="text" value="P020"/>
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	2	disha	madhukumar	virar	P002
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	3	hitesh	bhati	virar	P003
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	4	saloni	bhagat	virar	P004
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	5	rahul	agrahari	nalasopara	P005
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	6	pranav	gorate	nalasopara	P006
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	7	bhakti	gagre	virar	P007
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	8	jethalal	gada	gokuldham	P008
<a href="#">Edit</a>	<a href="#">Delete</a>	<a href="#">Select</a>	9	daya	gada	gokuldham	P009

## Practical No.: 13

**Aim:** Create a web application to demonstrate reading and writing operation with XML.

**Code:**

XMLFile.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<EmployeeDetails>
  <Details>
    <Name>Disha</Name>
    <Department>IT</Department>
    <Location>Virar</Location>
  </Details>
  <Details>
    <Name>Saloni</Name>
    <Department>CS</Department>
    <Location>Vasai</Location>
  </Details>
  <Details>
    <Name>Bhakti</Name>
    <Department>DS</Department>
    <Location>Palghar</Location>
  </Details>
  <Details>
    <Name>Devika</Name>
    <Department>IT</Department>
    <Location>virar</Location>
  </Details>
</EmployeeDetails>
```

readwritexml.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;
using System.Text;

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

        }
    }
}
```

```

protected void Button1_Click(object sender, EventArgs e)
{
    XmlDocument xmlEmployeeDoc = new XmlDocument();
    xmlEmployeeDoc.Load(Server.MapPath("~/XMLFile.xml"));
    XmlElement ParentElement = xmlEmployeeDoc.CreateElement("Details");
    XmlElement Name = xmlEmployeeDoc.CreateElement("Name");
    Name.InnerText = TextBox1.Text;
    XmlElement Department = xmlEmployeeDoc.CreateElement("Department");
    Department.InnerText = TextBox2.Text;
    XmlElement Location = xmlEmployeeDoc.CreateElement("Location");
    Location.InnerText = TextBox3.Text;
    ParentElement.AppendChild(Name);
    ParentElement.AppendChild(Department);
    ParentElement.AppendChild(Location);
    xmlEmployeeDoc.DocumentElement.AppendChild(ParentElement);
    xmlEmployeeDoc.Save(Server.MapPath("~/XMLFile.xml"));
}

protected void Button2_Click(object sender, EventArgs e)
{
    ReadXmlFile(Server.MapPath("XMLFile.xml"));
}

protected void ReadXmlFile(string xmlFile)
{
    string parentElementName = "";
    string childElementName = "";
    string childElementValue = "";
    bool element = false;
    Label1.Text = "";

    XmlTextReader xmlreader = new XmlTextReader(xmlFile);
    while (xmlreader.Read())
    {
        if(xmlreader.NodeType == XmlNodeType.Element)
        {
            if(element)
            {
                parentElementName = parentElementName + childElementName + "<br/>";
            }
            element = true;
            childElementName = xmlreader.Name;
        }
        else if(xmlreader.NodeType == XmlNodeType.Text | xmlreader.NodeType == XmlNodeType.CDATA)
        {
            element = false;
            childElementValue = xmlreader.Value;
            Label1.Text = Label1.Text + "<b>" + parentElementName + "<br/>" + childElementName + "<b><br/>" +
childElementValue;
            parentElementName = "";
            childElementName = "";
        }
    }
    xmlreader.Close();
}
}

```

**Output:**

The screenshot shows a web browser window with the address bar displaying 'localhost:64466/Readwritexml.aspx'. The page content includes a form titled 'Employee Details:' with three input fields for 'Name:', 'Department:', and 'Location:'. Below the form are two buttons: 'Write into XML file' and 'Read XML file'. The output section displays the XML data for three employees: EmployeeDetails, VirarDetails, and VasaiDetails. Each entry lists the Name, Department, and Location.

**Employee Details:**

Name:

Department:

Location:

[Write into XML file](#) [Read XML file](#)

**EmployeeDetails**  
**Details**

**Name**  
Disha  
**Department**  
IT  
**Location**  
**VirarDetails**

**Name**  
Saloni  
**Department**  
CS  
**Location**  
**VasaiDetails**

**Name**  
Bhakti  
**Department**  
DS  
**Location**  
**PalgharDetails**

**Name**  
Devika  
**Department**  
IT  
**Location**  
virar

## **Practical No.: 14**

**Aim:** Create a web application to demonstrate use of various Ajax controls.

**Code:**

ajax.aspx.cs:

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

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

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Text = System.DateTime.Now.ToString();
        }
    }
}
```

ajaxtimer.aspx.cs:

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

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

        }

        protected void Timer1_Tick(object sender, EventArgs e)
        {

```

```
        Label1.Text = System.DateTime.Now.ToString();  
    }  
}
```

**Output:**