|  |
| --- |
| .NET TECHNOLOGY Lab Practical |
| EnrollmentNo: 160470107003 |
| Param Ardeshna |

Contents

[Practical 1 1](#_Toc4828996)

[Introduction to C# 1](#_Toc4828997)

[Practical 2 8](#_Toc4828998)

[Print given Pattern 8](#_Toc4828999)

[Print given Pattern. 9](#_Toc4829000)

[Prompt a user to input his/her name and country name and print on console 11](#_Toc4829001)

[Practical 3 14](#_Toc4829002)

[Method Overloading 14](#_Toc4829003)

[Constructor Overloading 17](#_Toc4829004)

[Practical 4 20](#_Toc4829005)

[Reflection API 20](#_Toc4829006)

[Practical 5 23](#_Toc4829007)

[Program to copy data from one file to another using StreamReader and StreamWriter class. 23](#_Toc4829008)

[Program to Read Lines from a File until the End of File is Reached. 24](#_Toc4829009)

[Program to List Files in a Directory. 25](#_Toc4829010)

[Practical 6 27](#_Toc4829011)

[WindowsFromApplication 27](#_Toc4829012)

[Practical 7 31](#_Toc4829013)

[ASP.NET Validation Control 31](#_Toc4829014)

[Practical 8 35](#_Toc4829015)

[Introduction to Master Pages 35](#_Toc4829016)

# Practical 1

AIM:

## Introduction to C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Program1

{

class vector

{

public int value;

}

class Program1

{

static int i = 25;

public enum TimeOfDay

{

Morning = 0,

Afternoon = 1,

Evening = 2

}

static void Main(string[] args)

{

Console.WriteLine("This is first program");

//Scope of variables

int i=5;

Console.WriteLine("Scope of the variable {0}",i);

for (i = 0; i < 2; i++)

{

Console.WriteLine("{0} {1}",i,Program1.i);

}

for (int k = 0; k < 2; k++)

{

Console.WriteLine("{0}",k);

}

//Constant

const int valueConst=25;

Console.WriteLine("{0}",valueConst);

//valueConst = 15;

const int valueConst2 = 15;

Console.WriteLine("{0}", valueConst2);

//valueConst = valueConst2;

Console.WriteLine("{0}",valueConst);

//Value Type DataTypes

Console.WriteLine("Value Type");

int val1, val2;

val1 = 50;

Console.WriteLine("val1= {0}",val1);

val2 = val1;

Console.WriteLine("val1= {0} val2= {1}", val1,val2);

//Reference Type

Console.WriteLine("Reference Type");

vector x, y;

x = new vector();

x.value = 15;

y = x;

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

y.value = 151;

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

Console.WriteLine("\n Interger Types");

sbyte sb = 22;

short s = 22;

int i1 = 22;

long l = 22L;

Console.WriteLine("{0} sbtye\n{1} short\n{2} int\n{3} long\n",sb,s,i1,l);

Console.WriteLine("Unsigned Integers");

byte b = 21;

ushort us = 21;

uint ui = 21;

ulong ul = 21;

Console.WriteLine("{0} btye\n{1} ushort\n{2} uint\n{3} ulong\n", b, us, ui, ul);

Console.WriteLine("Floating Point");

float f = 11.22334455F;

double d = 11.2233445566778899;

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

decimal dec = 111.222333444555666777888999M;

Console.WriteLine("Decimal:\n{0}", dec);

Console.WriteLine("\nBoolean:");

bool valBoolean = true;

Console.WriteLine("Status: " + valBoolean);

Console.WriteLine("\nCharacter:\nSingle Quote \'");

Console.WriteLine("Double Quote \"");

Console.WriteLine("Back Slash \\");

char charA = 'A';

Console.WriteLine(charA);

int integerA = 2;

Console.WriteLine("Predefined Reference Type");

Object o1 = "This is object 1";

Object o2 = 34;

String strObj = o1 as string;

Console.WriteLine(strObj);

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

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

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

string s1, s2;

s1 = "String 1";

s2 = s1;

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

s2 = "New String 1";

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

s1 = "c:\\NewFolder\\Hello\\P1.cs";

Console.WriteLine(s1);

s1 = @"c:\NewFolder\Hello\P1.cs";

Console.WriteLine(s1);

s1 = @"We can also write

like this";

Console.WriteLine(s1);

Console.WriteLine("Flow control if statement");

bool isZero;

Console.WriteLine("\nFlow Control: (if)\ni is " + i);

if (i == 0)

{

isZero = true;

Console.WriteLine("i is Zero");

}

else

{

isZero = false;

Console.WriteLine("i is Non - zero");

}

//else if

Console.WriteLine("\nType in a string:");

string input;

input = Console.ReadLine();

if (input == "")

{

Console.WriteLine("You typed in an empty string");

}

else if (input.Length < 5)

{

Console.WriteLine("The string had less than 5 characters");

}

else if (input.Length < 10)

{

Console.WriteLine("The string had at least 5 but less than 10 characters");

}

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

Console.WriteLine("\nSwitch:");

switch (integerA)

{

case 1:

Console.WriteLine("integerA = 1");

break;

case 2:

Console.WriteLine("integerA = 2");

//goto case 3;

break;

case 3:

Console.WriteLine("integerA = 3");

break;

default:

Console.WriteLine("integerA is not 1, 2, or 3");

break;

}

WriteGreeting(TimeOfDay.Morning);

Console.WriteLine("Argument is: {0}", args[1]);

Console.ReadLine();

}

static void WriteGreeting(TimeOfDay timeOfDay)

{

switch (timeOfDay)

{

case TimeOfDay.Morning:

Console.WriteLine("Good morning!");

break;

case TimeOfDay.Afternoon:

Console.WriteLine("Good afternoon!");

break;

case TimeOfDay.Evening:

Console.WriteLine("Good evening!");

break;

default:

Console.WriteLine("Hello!");

break;

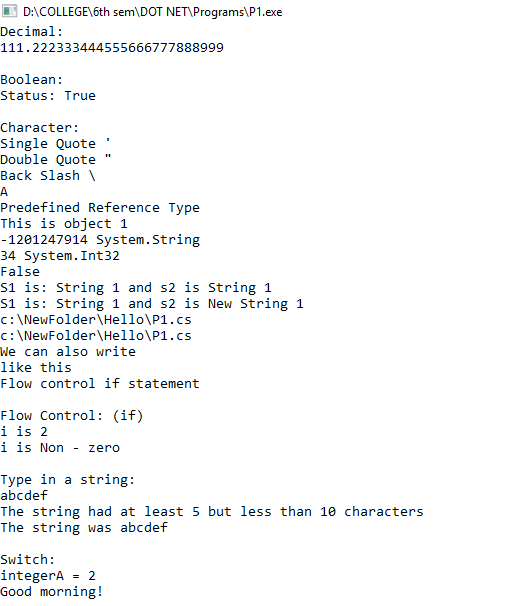
}

}

}

}

**Output:**



# Practical 2

Program 1:

AIM:

Print given Pattern

@ @ @ @ @

@ @ @ @

@ @ @

@ @

@

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Pattern1

{

class Program

{

static void Main(string[] args)

{

for (int i = 5; i > 0; i--)

{

for (int j = 0; j <i; j++)

{

Console.Write("@");

}

Console.WriteLine();

}

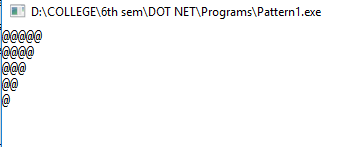
Console.ReadKey();

}

}

}

**Output:**



Program 2

AIM:

## Print given Pattern.

1

1 2

1 2 3

1 2 3 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Pattern2

{

class Program

{

static void Main(string[] args)

{

String s = Console.ReadLine();

int value = int.Parse(s);

for (int i = 1; i <= value; i++)

{

for (int j = 1; j <=i; j++)

{

Console.Write("{0} ",j);

}

Console.WriteLine();

}

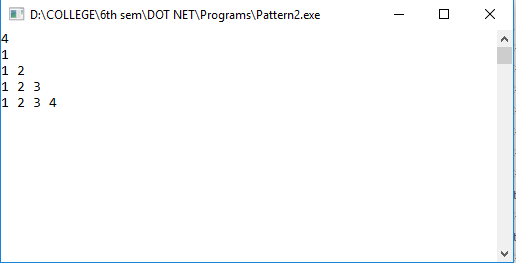
Console.ReadKey();

}

}

}

**Output:-**



Program 3

AIM:

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

Hello Ram from country India

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace PrintNameCountry

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter name");

String name = Console.ReadLine();

Console.WriteLine("Enter Country");

String country = Console.ReadLine();

Console.WriteLine("Hello {0} from country {1}", name, country);

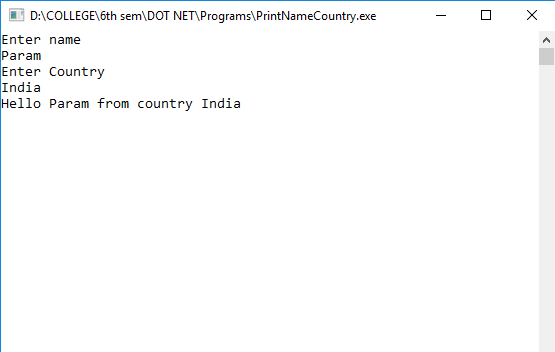
Console.ReadKey();

}

}

}

**Output:**



Program 4

AIM: Create C# console application to define Car class and derive Maruti and Mahindra from it to Demonstrate Inheritance.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Inheritance

{

class Car

{

protected String name, fuel,id;

}

class Maruti: Car

{

internal Maruti(String name, String fuel, String id)

{

this.name = name;

this.fuel = fuel;

this.id = id;

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

}

}

class Mahindra : Car

{

internal Mahindra(String name, String fuel, String id)

{

this.name = name;

this.fuel = fuel;

this.id = id;

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

}

}

class Program

{

static void Main(string[] args)

{

Maruti obj1= new Maruti("abc","petrol","123");

Mahindra obj2 =new Mahindra("pqr","disel","456");

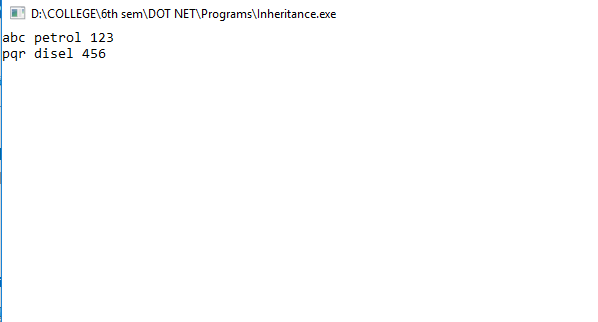
Console.ReadKey();

}

}

}

**Output:**



# Practical 3

Program 1

## Method Overloading

AIM: Write a c# program to add two integers, two vectors and two metric using Method Overloading.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace MethodOverloading

{

class Vector

{

internal int x, y, z;

internal Vector(int x, int y, int z)

{

this.x = x;

this.y = y;

this.z = z;

}

internal Vector() { }

}

class Matrix

{

internal int [,] m = new int[2,2];

internal Matrix(){}

}

class Program

{

static void add(int a, int b)

{

int temp = a + b;

Console.WriteLine(temp);

}

static void add(Vector a, Vector b)

{

Vector temp = new Vector();

temp.x = a.x + b.x;

temp.y = a.y + b.y;

temp.z = a.z + b.z;

Console.WriteLine("{0}x {1}y {2}z", temp.x, temp.y, temp.z);

}

static void add(Matrix a, Matrix b)

{

Matrix temp = new Matrix();

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

temp.m[i, j]=a.m[i,j]+b.m[i,j];

Console.Write(temp.m[i, j]);

}

Console.WriteLine();

}

}

static void Main(string[] args)

{

Console.WriteLine("Enter Vector");

Vector a = new Vector(int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine()));

Vector b = new Vector(int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine()), int.Parse(Console.ReadLine()));

add(a, b);

Console.WriteLine("Enter integer");

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

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

add(x, y);

Matrix m1 = new Matrix();

Matrix m2 = new Matrix();

m1.m[0, 0] = 2;

m1.m[0, 1] = 2;

m1.m[1, 0] = 2;

m1.m[1, 1] = 2;

m2.m[0, 0] = 3;

m2.m[0, 1] = 3;

m2.m[1, 0] = 3;

m2.m[1, 1] = 3;

add(m1, m2);

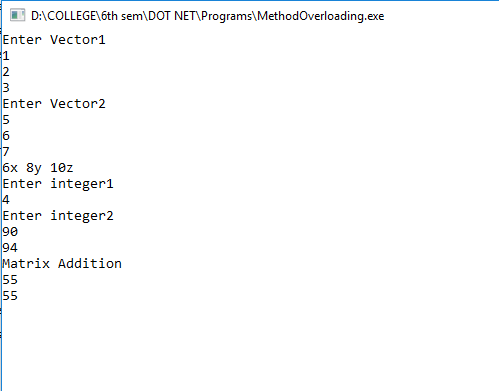
Console.ReadKey();

}

}

}

**Output:**



Program 2

## Constructor Overloading

AIM: Write a c# program that create student object. Overload Constructor to create new instant with following details.

1. Name

2. Name, Enrollment

3. Name, Enrollment, Branch

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConstructorOverloading

{

class Student

{

String name,enroll\_no,branch;

public Student(String name)

{

this.name = name;

}

public Student(String name, String enroll\_no)

{

this.name = name;

this.enroll\_no = enroll\_no;

}

public Student(String name, String enroll\_no, String branch)

{

this.name = name;

this.enroll\_no = enroll\_no;

this.branch = branch;

}

internal String getName()

{

return this.name;

}

internal String getEnroll()

{

return this.enroll\_no;

}

internal String getBranch()

{

return this.branch;

}

}

class Program

{

static void Main(string[] args)

{

Student s1 = new Student("abc");

Console.WriteLine(s1.getName());

Student s2 = new Student("pqr", "16047010459");

Console.WriteLine(s2.getName());

Console.WriteLine(s2.getEnroll());

Student s3 = new Student("xyz", "1604710236", "computer");

Console.WriteLine(s3.getName());

Console.WriteLine(s3.getEnroll());

Console.WriteLine(s3.getBranch());

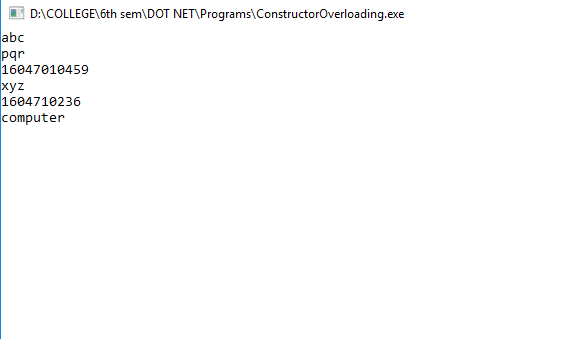
Console.ReadKey();

}

}

}

**Output:**



# Practical 4

## Reflection API

AIM: Create a c# program to find Methods, Properties and Constructors from class of running program using Reflection API.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Reflection;

namespace Reflection

{

class Student

{

String name, enroll\_no, branch;

public Student(String name)

{

this.name = name;

}

public Student(String name, String enroll\_no)

{

this.name = name;

this.enroll\_no = enroll\_no;

}

public Student(String name, String enroll\_no, String branch)

{

this.name = name;

this.enroll\_no = enroll\_no;

this.branch = branch;

}

public String getName()

{

return this.name;

}

public String getEnroll()

{

return this.enroll\_no;

}

public String getBranch()

{

return this.branch;

}

}

class Program

{

static void Main(string[] args)

{

Type t = Type.GetType("Reflection.Student");

ConstructorInfo[] ci = t.GetConstructors();

MethodInfo[] mi = t.GetMethods();

foreach (ConstructorInfo c in ci)

{

Console.WriteLine(c.ToString());

}

foreach (MethodInfo m in mi)

{

Console.WriteLine(m.ToString());

}

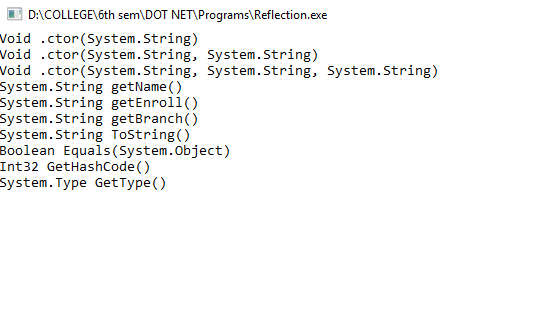
Console.ReadLine();

}

}

}

**Output:**



# Practical 5

AIM:

## Program to copy data from one file to another using StreamReader and StreamWriter class.

Program 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace CopyFile1

{

class Program

{

static void Main(string[] args)

{

String file1 = @"F:\16ce043\source.txt";

String file2 = @"F:\16ce043\destination1.txt";

using (StreamReader reader = new StreamReader(file1))

{

using (StreamWriter writer = new StreamWriter(file2))

{

writer.Write(reader.ReadToEnd());

}

}

}

}

}

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

Program 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace CopyFile2

{

class Program

{

static void Main(string[] args)

{

String file1 = @"F:\16ce043\source.txt";

String file2 = @"F:\16ce043\destination2.txt";

String content = null;

using (StreamReader reader = new StreamReader(file1))

{

using (StreamWriter writer = new StreamWriter(file2))

{

while ((content = reader.ReadLine())!= null)

{

writer.WriteLine(content);

}

}

}

}

}

}

## Program to List Files in a Directory.

Program 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace DirectoryStructure

{

class Program

{

static void Main(string[] args)

{

String[] directories = Directory.GetDirectories(@"F:\16ce043");

String[] files = Directory.GetFiles(@"F:\16ce043");

Console.WriteLine("Directories are");

foreach (String dir in directories)

Console.WriteLine(dir);

Console.WriteLine("Files are");

foreach (String file in files)

Console.WriteLine(file);

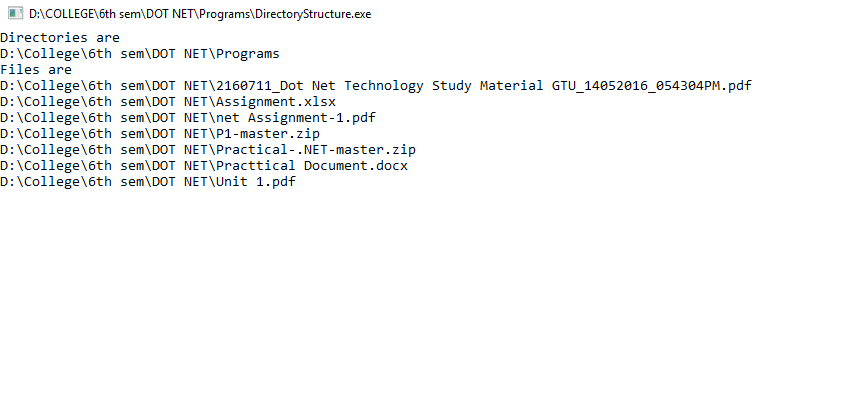
Console.ReadKey();

}

}

}

**Output:**



# Practical 6

## WindowsFromApplication

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

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace StudentRegistrationForm

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Gender\_Click(object sender, EventArgs e)

{

}

private void Submit\_Click(object sender, EventArgs e)

{

String gender;

if (rdoMale.Checked == true)

{

gender = "Male";

}

else

{

gender = "Female";

}

String source = @"Data Source=mycomputer\sqlexpress;Initial Catalog=DBstudent;Integrated Security=True;Pooling=False";

String select = "select count(\*) from tblStudent;";

SqlConnection conn=new SqlConnection(source);

SqlCommand cmd=new SqlCommand(select,conn);

conn.Open();

int i=Convert.ToInt32(cmd.ExecuteScalar());

int pkStudent=i+1;

string insert="insert into tblStudent(pkStudent,fname,mname,lname,dob,Address,email,Father\_name,Mother\_name,Mobile\_Number,Gender) values("+pkStudent+",'"+txtFirstName.Text+"','"+txtMiddleName.Text+"','"+txtLastName.Text+"','"+datepickDOB.Value.Date+"','"+txtAddress.Text+"','"+txtEmail.Text+"','"+txtFatherName.Text+"','"+txtMotherName.Text+"',"+txtMobNo.Text+",'"+gender+"')";

cmd= new SqlCommand(insert,conn);

int a= cmd.ExecuteNonQuery();

MessageBox.Show("You are Done");

InitializeComponent();

}

private void openFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

}

private void btnUpload\_Click(object sender, EventArgs e)

{

openFileDialog1.Filter = "Jpg|\*.jpg";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

String imgPath;

imgPath = @"F:\16ce043\StudentRegistrationForm"+ openFileDialog1.SafeFileName;

imgStudent.Image = Image.FromFile(openFileDialog1.FileName);

}

}

private void Form1\_Load(object sender, EventArgs e)

{

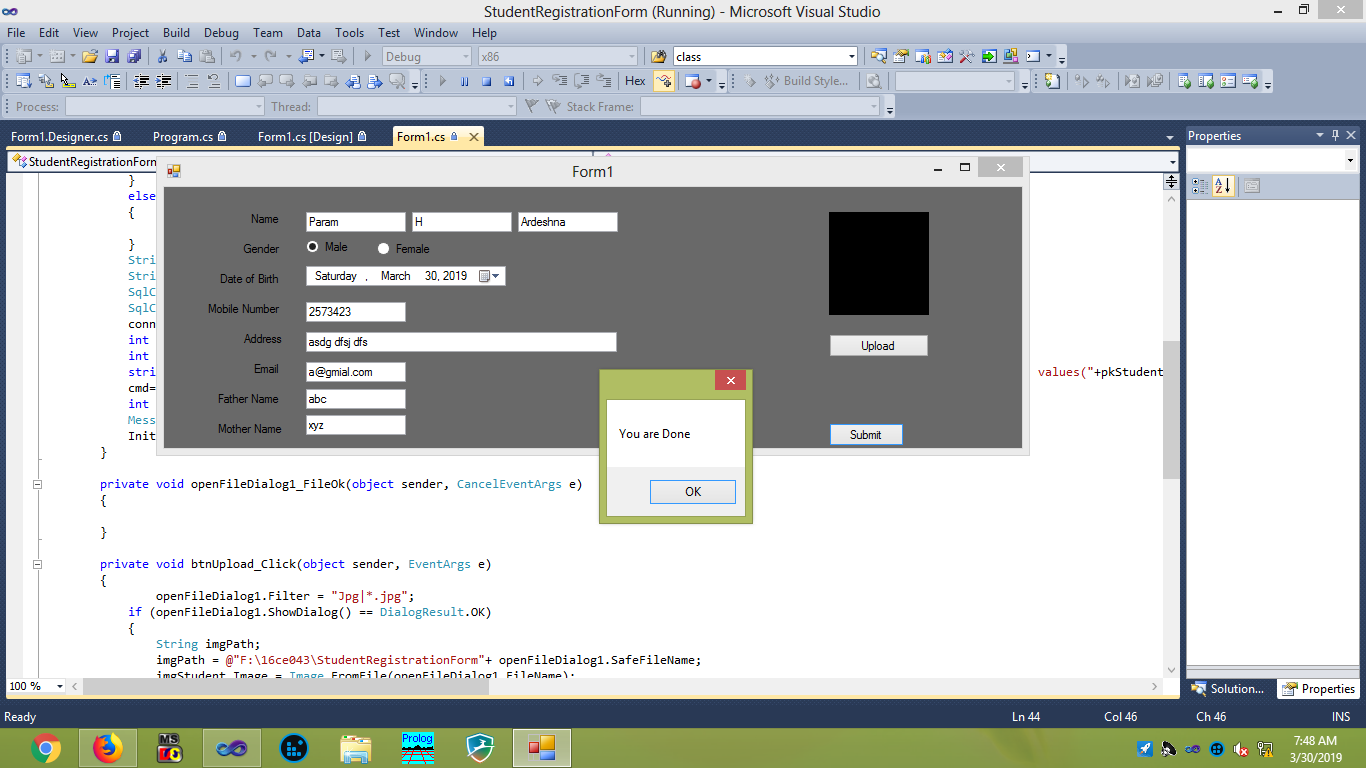
rdoMale.Checked = true;

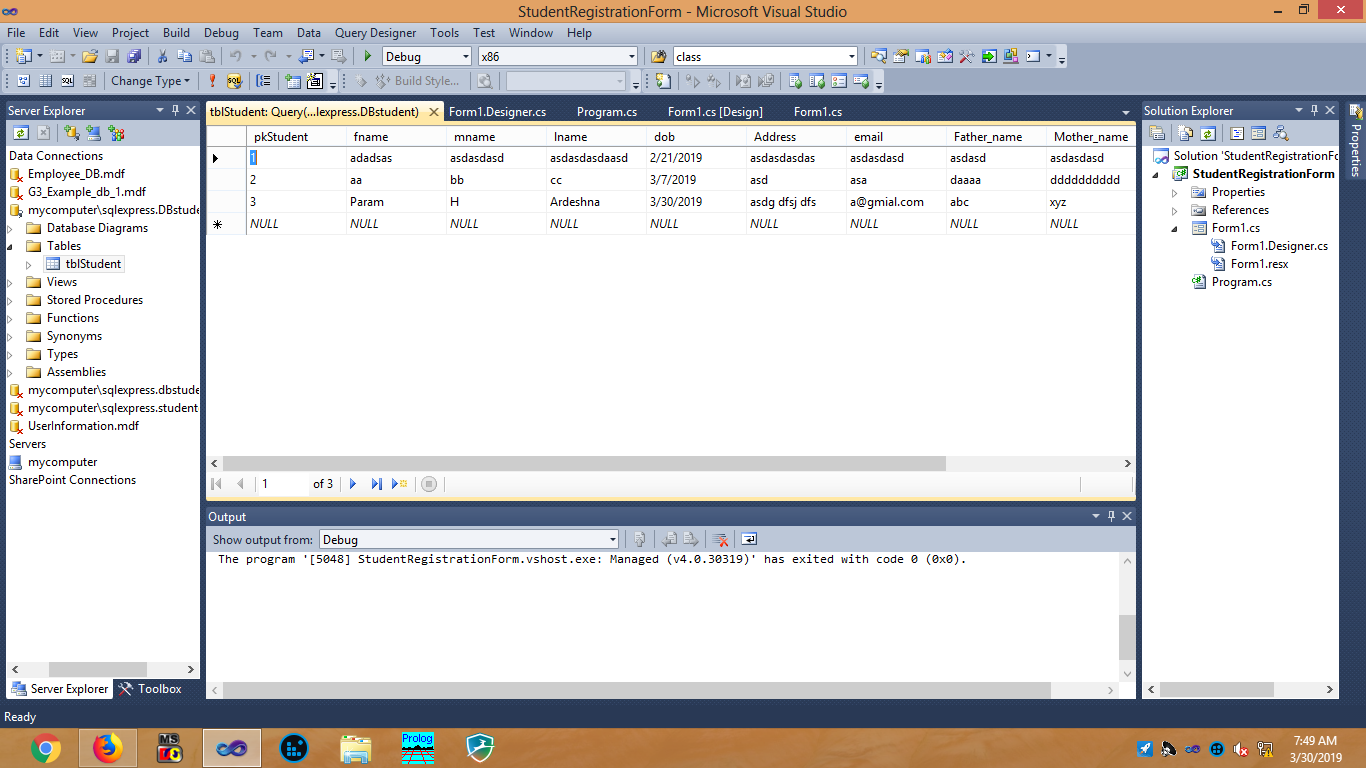
}

}

}

**Output:-**

****



# Practical 7

AIM:

## ASP.NET Validation Control

RequiredFieldValidator, CompareValidator, RegularExpressionValidator, CustomValidator,  
RangeValidator, ValidationSummary.

**WebForm1.aspx**

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="Label1" runat="server" Text="Name"></asp:Label>

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"

ControlToValidate="txtName" Display="Dynamic" ErrorMessage="Please Enter Name"

ForeColor="Red">\*</asp:RequiredFieldValidator>

</div>

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

<asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"

ControlToValidate="txtEmail" Display="Dynamic" ErrorMessage="Enter valid email"

ForeColor="Red"

ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*">\*</asp:RegularExpressionValidator>

<p>

<asp:Label ID="Label3" runat="server" Text="Mobile"></asp:Label>

<asp:TextBox ID="txtMobile" runat="server"></asp:TextBox>

<asp:RegularExpressionValidator ID="RegularExpressionValidator2" runat="server"

ControlToValidate="txtMobile" ErrorMessage="Enter valid Mobile number"

ForeColor="Red" ValidationExpression="[0-9]{10}"></asp:RegularExpressionValidator>

</p>

<p>

<asp:Label ID="Label4" runat="server" Text="Password"></asp:Label>

<asp:TextBox ID="txtPassword" runat="server"></asp:TextBox>

</p>

<p>

<asp:Label ID="Label5" runat="server" Text="Confirm Password"></asp:Label>

<asp:TextBox ID="txtConfirm" runat="server"></asp:TextBox>

<asp:CompareValidator ID="CompareValidator1" runat="server"

ControlToCompare="txtPassword" ControlToValidate="txtConfirm" Display="Dynamic"

ErrorMessage="CompareValidator" ForeColor="Red"></asp:CompareValidator>

</p>

<p>

<asp:Label ID="Label6" runat="server" Text="Semester"></asp:Label>

<asp:TextBox ID="txtSemester" runat="server"></asp:TextBox>

<asp:RangeValidator ID="RangeValidator1" runat="server"

ControlToValidate="txtSemester" ErrorMessage="RangeValidator" ForeColor="Red"

MaximumValue="8" MinimumValue="1" Type="Integer"></asp:RangeValidator>

</p>

<p>

<asp:Button ID="Button1" runat="server" onclick="Button1\_Click" Text="Submit" />

</p>

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

</form>

</body>

</html>

**WebForm1.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace ValidationControls

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

}

}

}

Output:-



# Practical 8

AIM:

## Introduction to Master Pages

**Site1.Master**

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title></title>

<asp:ContentPlaceHolder ID="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

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

<div>

<table style="height:400px;width:400px">

<tr>

<td>

<asp:Label ID="lblHeader" runat="server" Text="Header"></asp:Label>

</td>

</tr>

<tr>

<td><asp:Button ID="btnSearch" runat="server"

Text="Search" />

<asp:TextBox ID="txtSearch" runat="server"></asp:TextBox>

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

<p>

Web Content

</p>

</asp:ContentPlaceHolder>

</td>

</tr>

<tr>

<td>Footer

</td>

</tr>

</table>

</div>

</form>

</body>

</html>

**Site1.Master.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace MasterPages

{

public partial class Site1 : System.Web.UI.MasterPage

{

protected void Page\_Load(object sender, EventArgs e)

{

}

public Label LblHeader

{

get { return lblHeader; }

}

public Button BtnSearch

{

get { return btnSearch; }

}

public TextBox TxtSearch

{

get { return txtSearch; }

}

}

}

**WebForm1.aspx**

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

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<asp:TextBox ID="txtHeader" runat="server"></asp:TextBox>

<asp:Button ID="btnHeader" runat="server" onclick="btnHeader\_Click"

Text="Change Header" />

&nbsp;

</asp:Content>

**WebForm1.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace MasterPages

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnHeader\_Click(object sender, EventArgs e)

{

((Site1)Master).LblHeader.Text = txtHeader.Text;

}

}

}

**WebForm2.aspx**

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="MasterPages.WebForm2" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<asp:GridView ID="grdStudent" runat="server">

</asp:GridView>

</asp:Content>

**WebForm2.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

namespace MasterPages

{

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

{

protected void Page\_Init(object sender, EventArgs e)

{

((Site1)Master).BtnSearch.Click += new EventHandler(BtnSearch\_Click);

}

void BtnSearch\_Click(object sender, EventArgs e)

{

getData();

}

protected void Page\_Load(object sender, EventArgs e)

{

}

void getData()

{

String source = @"Data Source=mycomputer\sqlexpress;Initial Catalog=DBstudent;Integrated Security=True;Pooling=False";

SqlConnection conn = new SqlConnection(source);

String search = ((Site1)Master).TxtSearch.Text;

String query = "select \* from tblStudent where fname like '%"+search+"%'";

SqlCommand cmd = new SqlCommand(query, conn);

conn.Open();

SqlDataReader reader = cmd.ExecuteReader();

grdStudent.DataSource = reader;

grdStudent.DataBind();

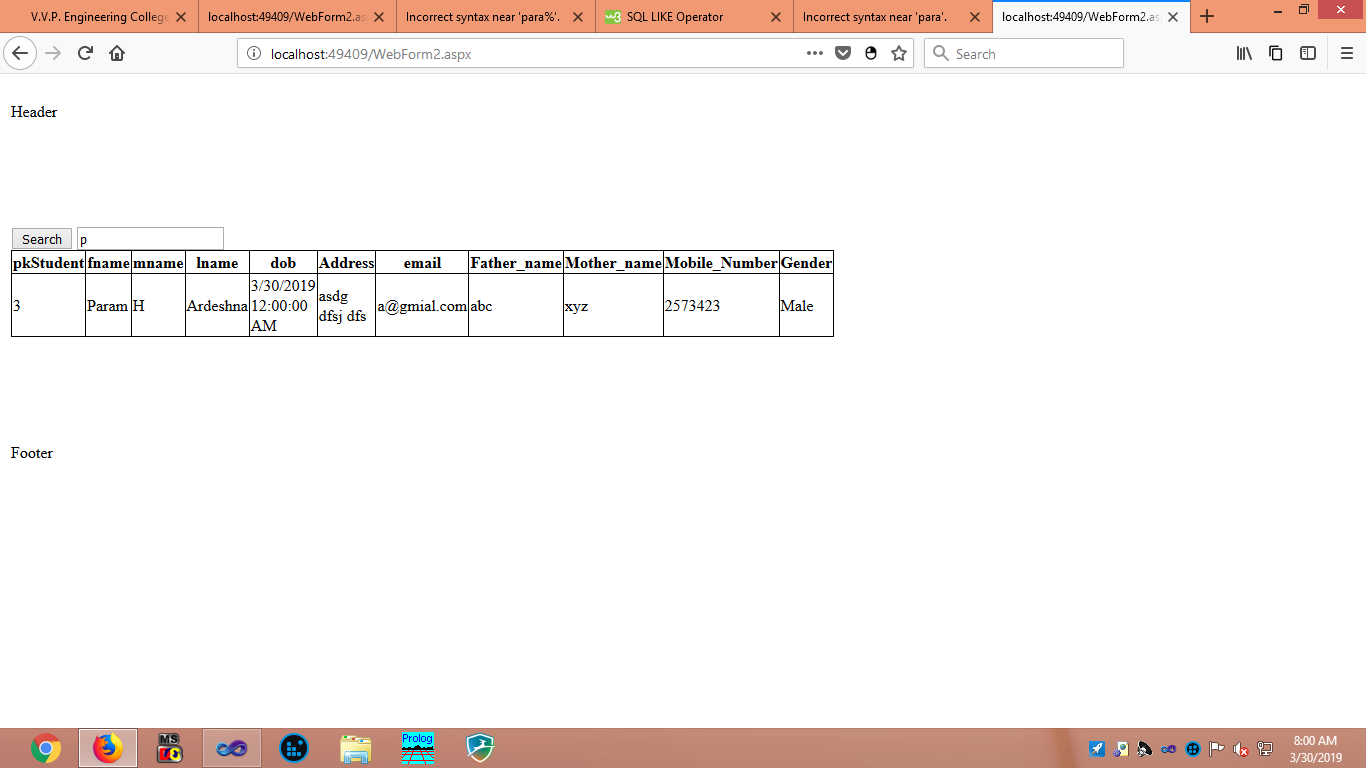
conn.Close();

}

}

}

**Output:-**

****