|  |
| --- |
|  |
| **.NET** |
| **Lab Manual** |

Shreyas Palan

160470107041

Table of Contents

[**Introduction to C#** 3](#_Toc4802460)

[Program 1: 3](#_Toc4802461)

[**GTU Programs** 10](#_Toc4802462)

[Program 1. 10](#_Toc4802463)

[Program 2. 11](#_Toc4802464)

[Program 3. 12](#_Toc4802465)

[Program 4. 12](#_Toc4802466)

[**Overloading** 15](#_Toc4802467)

[Program 1: 15](#_Toc4802468)

[Program 2: 17](#_Toc4802469)

[**Reflection** 19](#_Toc4802470)

[Program 1: 19](#_Toc4802471)

[**File Handling** 21](#_Toc4802472)

[Program 1: 21](#_Toc4802473)

[Program 2: 22](#_Toc4802474)

[Program 3: 23](#_Toc4802475)

[**Windows Form** 25](#_Toc4802476)

[Program 1: 25](#_Toc4802477)

[**Validator controls** 27](#_Toc4802478)

[Program 1: 27](#_Toc4802479)

[**Introduction to master page** 30](#_Toc4802480)

[Program 1: 30](#_Toc4802481)

**Practical-1**

# **Introduction to C#**

## Program 1:

Variables:

Initialization

Scope

Constant

Predefined Data Types

Value Types

Reference TYpes

Flow Control

Conditional Statements(if, switch)

Loop(for, while, dowhile, foreach)

Jump(goto, break, continue, return)

Eumerations

Passing Arguments

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace aim

{

class Program

{

static int newint=100;

public enum TimeOfDay

{

Morning = 0,

Afternoon = 1,

Evening = 2

}

public static void Main(string[] args)

{

Console.WriteLine("\n integer types");

sbyte sb = 10;

short s = 33;

int i = 10;

long l = 33L;

byte b = 22;

ushort us = 33;

uint ul = 33u;

ulong ulo = 33ul;

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

float f = 1.122345656767f;

double d = 12.1234455657878797;

Console.Write("\nFloat and Double:\n");

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

decimal dec=111.666666666666666666666M;

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

Console.WriteLine("\nBoolean:");

bool boolean =true;

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

// Console.ReadLine();

char character ='d';

Console.WriteLine(character);

character = '\0';

Console.WriteLine("Now null: " + character);

object o1 = "Hi, I am ALICE";

object o2 = 15.3454365;

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 = "this is string";

s2 = s1;

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

s2 = "other string";

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

s1 = "c:C:\\Users\\Dell\\source\\repos\\aim";

Console.WriteLine(s1);

s1 = @"c:C:\Users\Dell\source\repos\aim\aim";

Console.WriteLine(s1);

s1 = @"We can also write

like this";

Console.WriteLine(s1);

bool isZero;

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

if (i == 10)

{

isZero = true;

Console.WriteLine("i is Zero {0}",isZero);

}

else

{

isZero = false;

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

}

int integerA = 1;

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]);

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;

}}

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

int newint=0;

int j;

for (/\*int\*/ j = 0; j < 2; j++) //removing comment from for loop will raise error

{

//int j;

//uncomment above line to error "A local variable named 'j' cannot be declared in this

//scope because it would give a different meaning to 'j', which is already

//used in a 'parent or current' scope to denote something else"

Console.Write("{0} {1}\n", newint, Program.newint);

}

Console.WriteLine("2:");

for (int k = 0; k < 3; k++)

{

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

}//Scope of k ends here

Console.Write("\n");

//Console.Write(k);

//uncomment above line to see error "The name 'k' does not exist in the current context"

for (int k = 3; k > 0; k--)

{

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

}//scope of k ends here again

Console.WriteLine("Constants");

const int valConst = 100; // This value cannot be changed.

Console.WriteLine("{0} is constant value", valConst);

//valConst = 45;

//uncomment above line to see error "The left-hand side of an assignment must be a variable, property or indexer"

//const only allow constant variables into the expression

const int valConst2 = valConst + 9 /\* + j\*/;

//remove comments from the above line to see error "The expression being assigned to 'valConst2' must be constant"

Console.WriteLine("Another Constant: {0}", valConst2);

Console.WriteLine("\nPredefined Data Types\n\nValue Types and Reference Types");

//Value Types

int vali = 2, valj = vali;

Console.WriteLine("vali is: {0} and valj is: {1}", vali, valj);

valj = 90;

Console.WriteLine("vali is: {0} and valj is: {1}", vali, valj);

//Referece Types

Vector x, y;

x = new Vector();

x.value = 3;

y = x;

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

y.value = 234;

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

//If a variable is a reference, it is possible to indicate that it does not refer to any object by setting its value to null:

y = null;

//Console.Write("Value for y is: " + y.value);

//uncomment above line to see runtime exception "System.NullReferenceException: Object reference not set to an instance of an object."

//CTS

}

public class Vector

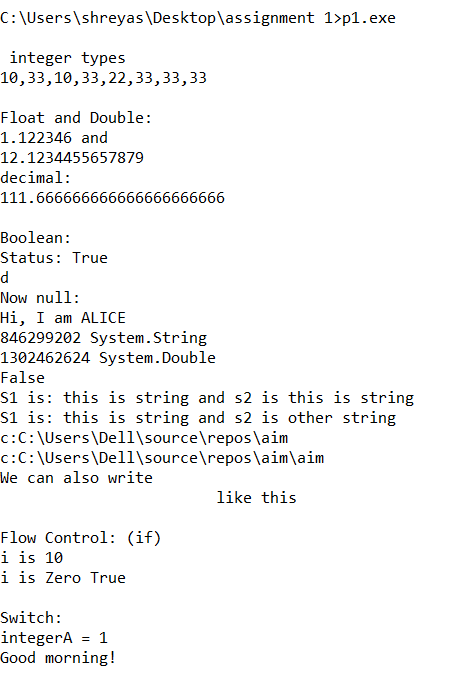
{

public int value;

}

}

}



**Practical-2**

# **GTU Programs**

Program 1.

Write console based program in code behind language VB or C# to print following pattern.

@@@@@

@@@@

@@@

@@

@

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace p2

{

class Pattern1

{

static void Main(string[] args)

{

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

for (int j = i; j > 0; j--) { Console.Write('@');

} Console.WriteLine();

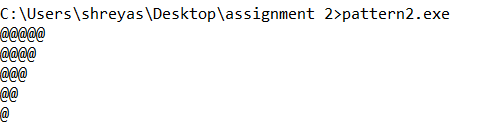
}

Console.ReadKey();

}

}

}



## Program 2.

Write console based program in code behind language VB or C# to print following pattern.

1

12

123

1234

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace p2

{

class Pattern2

{

static void Main(String[] ar){

for(int i=1;i<5;i++){

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

Console.Write(j);

}

Console.WriteLine();

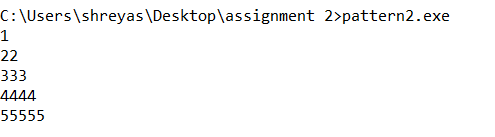
}

Console.ReadKey();

}

}

}



Program 3.

Write C# code to prompt a user to input his/her name and country name and then the output will be shown as an example below:

Hello Ram from country India

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace p2

{

class Read

{

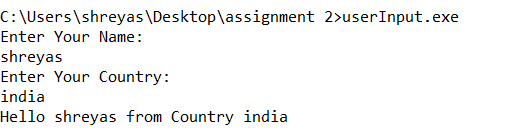
static void Main(String[] ar) { Console.WriteLine("Enter your name:"); string name = Console.ReadLine(); Console.WriteLine("Enter your City:"); string city = Console.ReadLine();

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

}

}

}



# 

Program 4.

What is inheritance? 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;

using System.Threading.Tasks;

namespace ConsoleApp3

{

class Car

{

public String name;

public Car(String name)

{

this.name = name;

}

public Car()

{

}

}

class Maruti : Car

{

public Maruti(String name): base(name) { }

public void display()

{

Console.WriteLine(name);

}

}

class Mahindra : Car

{

public Mahindra(String name) : base(name) { }

public void display()

{

Console.WriteLine(name);

}

}

class MainClass

{

public static void Main()

{

Maruti maruti = new Maruti("maruti 800");

Mahindra mahindra = new Mahindra("XUV 500");

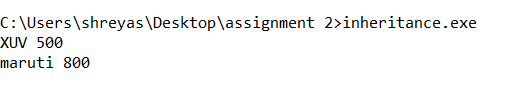
mahindra.display();

maruti.display();

}

}

}



**Practical – 3**

# **Overloading**

## Program 1:

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;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class MethodOverloding

{

public void add(int a, int b)

{

Console.WriteLine(a+b);

}

public void add(Vector v1, Vector v2)

{

Vector v3 = new Vector();

v3.X = v1.X + v2.X;

v3.Y = v1.Y + v2.Y;

Console.WriteLine("x= {0} and y={1}",v3.X,v2.Y);

}

}

class MainClass

{

static void Main(string[] args)

{

MethodOverloding m = new MethodOverloding();

m.add(2,5);

Vector v1 = new Vector();

Vector v2 = new Vector();

v1.X = 2;

v1.Y = 3;

v2.X = 4;

v2.Y = 5;

m.add(v1, v2);

}

}

class Vector

{

int x, y;

public int X{

get{

return x;

}

set{

this.x = value;

}

}

public int Y{

get{

return y;

}

set{

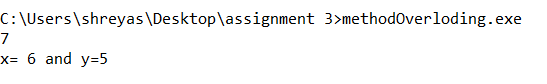
this.y = value;

}

}

}

}



## Program 2:

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;

using System.Threading.Tasks;

namespace ConsoleApp3

{

class Student

{

String name;

String enrollment;

String branch;

public Student(string name, string enrollment) : this(name)

{

this.enrollment = enrollment;

}

public Student(string name)

{

this.name = name;

}

public Student(string name, string enrollment, string branch)

{

this.name = name;

this.enrollment = enrollment;

this.branch = branch;

}

public override string ToString()

{

return name + " " +enrollment ;

}

}

class MainClass

{

public static void Main()

{

Student s1 = new Student("rockey");

Student s2 = new Student("perfect", "164895004");

Student s3 = new Student("nick", "13324443", "ce");

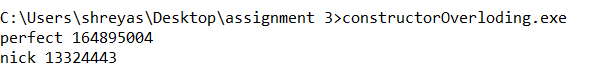
Console.WriteLine(s2.ToString());

Console.WriteLine(s3.ToString());

}

}

}



**Practical-4**

# **Reflection**

## Program 1:

Create a c# program to find Methods, Properties and Constructors from class of running program.(Use Class from previous practical)

using System;

using System.Reflection;

namespace ReflectionExample

{

class Customer

{

public int ID { get; set; }

public string Name { get; set; }

public Customer(int ID, string Name)

{

this.ID = ID;

this.Name = Name;

}

public Customer()

{

this.ID = -1;

this.Name = string.Empty;

}

public void printID()

{

Console.WriteLine("ID is: {0}", this.ID);

}

public void printName()

{

Console.WriteLine("Name is: {0}", this.Name);

}

public static void Main()

{

Type T = Type.GetType("ReflectionExample.Customer");

MethodInfo[] methods = T.GetMethods();

foreach (MethodInfo method in methods)

{

Console.WriteLine(method.ReturnType + " " + method.Name);

}

PropertyInfo[] properties = T.GetProperties();

Console.WriteLine("\nProperties");

foreach (PropertyInfo property in properties)

{

Console.WriteLine(property.PropertyType+" "+ property.Name);

}

Console.WriteLine("\nConstructors");

ConstructorInfo[] constructors = T.GetConstructors();

foreach (ConstructorInfo constructor in constructors)

{

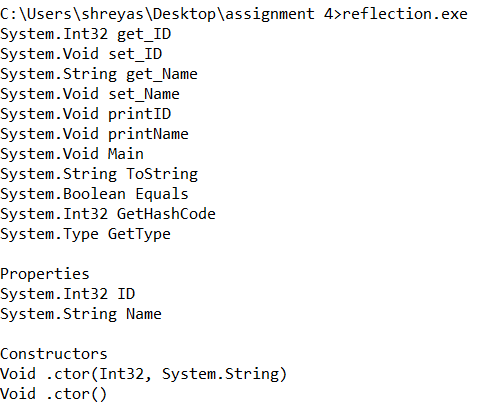
Console.WriteLine(constructor.ToString());

}

}

}

}



**Practical-5**

# **File Handling**

## Program 1:

Write a C# program to copy data from one file to another using

Stream Reader and Stream Writer class.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace p2

{

class P4\_1

{

public static void Main(){

string f1 = @"f1.txt";

string f2 = @"f2.txt";

using (StreamReader reader = new StreamReader(f1))

using (StreamWriter writer = new StreamWriter(f2))

writer.Write(reader.ReadToEnd());

}

}

}

F1.txt : Hello World..

F2.txt: Hello World..

## Program 2:

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

Reached.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace p2

{

class ListFile

{

public static void Main()

{

using (StreamReader SR = new StreamReader (@"C:\Users\shreyas\Desktop\assignment 5\f2.txt")) {

string line = null;

while ((line = SR.ReadLine()) != null) {

Console.WriteLine(line);

}

}

}

}

}



## Program 3:

Write a C# Program to List Files in a Directory.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

namespace p2

{

class ListFile

{

public static void Main() {

string[] Directories = Directory.GetDirectories(@"E:\Sem-6\VS");

foreach (string dir in Directories)

Console.WriteLine(dir);

string[] files = Directory.GetFiles(@"E:\Sem-6\VS");

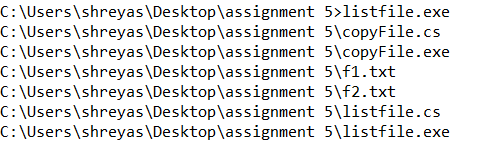
foreach (string file in files) Console.WriteLine(file);

Console.ReadKey();

}

}

}



**Practical 6**

# **Windows Form**

## Program 1:

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

SqlConnection con;

SqlCommand cmd;

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

{

}

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

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

con = new SqlConnection(@”Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\shreyas\source\repos\WindowsFormsApp1\WindowsFormsApp1\Database1.mdf;Integrated Security=True”);

cmd = new SqlCommand(“insert into Students (Name,Enrollment,Email)values ( ‘”+textBox1.Text+”’,’”+textBox2.Text+”’,’”+textBox3.Text+”’)”, con);

con.Open();

int x = cmd.ExecuteNonQuery();

if(x ==1){

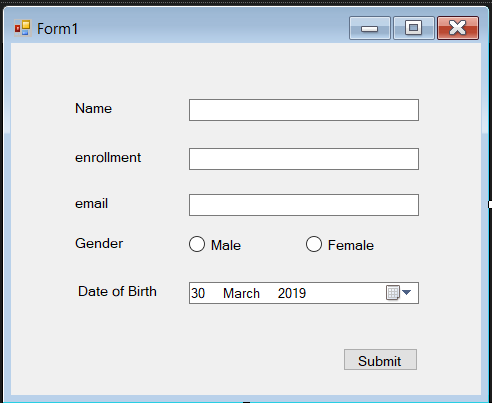
Console.WriteLine(“success”);

}

}

}

}



**Practical 7**

# **Validator controls**

## Program 1:

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

<div>

<table>

<tr>

<td>

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

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

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtname" ErrorMessage="RequiredFieldValidator"></asp:RequiredFieldValidat or>

<br />

</td>

</tr>

<tr>

<td>

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

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

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

ErrorMessage="RegularExpressionValidator" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-

.]\w+)\*" ControlToValidate="txtemail"></asp:RegularExpressionValidator>

<br />

</td>

</tr>

<tr>

<td>

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

<asp:TextBox ID="txtpass" runat="server" TextMode="Password"></asp:TextBox>

<br />

</td>

</tr>

<tr>

<td>

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

<asp:TextBox ID="txtcpass" runat="server" TextMode="Password"></asp:TextBox>

<asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="txtcpass" ControlToValidate="txtpass" ErrorMessage="CompareValidator"></asp:CompareValidator>

<br />

</td>

</tr>

<tr>

<td>

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

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

<asp:RangeValidator ID="RangeValidator1" runat="server" ControlToValidate="txtsem" ErrorMessage="RangeValidator" MaximumValue="8"

MinimumValue="1"></asp:RangeValidator>

<br />

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

</td>

</tr>

<tr>

<td>

<asp:Button ID="Button1" runat="server" Text="Save" />

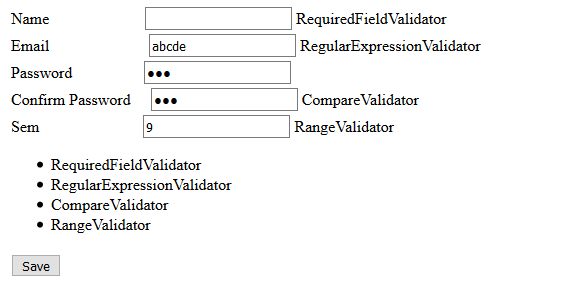
</td>

</tr>

</table>

</div>

</form>



**Practical 8**

# **Introduction to master page**

## Program 1:

Site1.master

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs" Inherits="WebApplication2.Site1" %>

<!DOCTYPE html>

<html>

<head runat="server">

<title></title>

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

</asp:ContentPlaceHolder>

</head>

<body>

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

<table>

<tr>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>

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

</td>

<td>

<div>

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

</asp:ContentPlaceHolder>

</div>

</td>

</tr>

<tr>

<td>

Footer

</td>

</tr>

</table>

</form>

</body>

</html>

Site1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication2

{

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

{

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

{

}

public Label Lblname

{

get

{

return lblname;

}

}

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="WebApplication2.WebForm1" %>

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

</asp:Content>

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

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

<asp:Button ID="Btnname" runat="server" Text="Set Header" onclick="Btnname\_Click" />

</asp:Content>

Webform1.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication2

{

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

{

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

{

}

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

{

((Site1)Master).Lblname.Text = txtname.Text;

}

}

}

Webform2.aspx

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="WebApplication2.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:Content>

Webform2.aspx.cs

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication2

{

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();

}

void getData()

{

string s = ((Site1)Master).Txtsearch.Text; Console.WriteLine(s);

string source = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\shreyas\source\repos\WebApplication2\WebApplication2\App\_Data\Database1.mdf;Integrated Security=True";

string select = "select \* from Students where Fname like '%" + ((Site1)Master).Txtsearch.Text + "%'";

SqlConnection con = new SqlConnection(source);

SqlCommand cmd = new SqlCommand(select, con);

con.Open();

SqlDataReader rdr = cmd.ExecuteReader();

grdstudent.DataSource = rdr; grdstudent.DataBind();

con.Close();

}

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

{

}

}

}

