VVP SEM6 .NET

RADHIKA ODEDRA

lab manual

.net assignment

Contents

[**AIM-**INTRODUCTION TO C#. 1](#_Toc1317677)

[AIM-GTU QUESTIONS. 8](#_Toc1317678)

[Program 1 8](#_Toc1317679)

[Program 2 9](#_Toc1317680)

[Program 3 10](#_Toc1317681)

[Program 4 12](#_Toc1317682)

[AIM-METHOD AND CONSTRUCTOR OVERLOADING. 16](#_Toc1317683)

[Program 1 16](#_Toc1317684)

[Program 2 19](#_Toc1317685)

[AIM: REFLECTION 22](#_Toc1317686)

**PRACTICAL-1**

# **AIM-**INTRODUCTION TO C#.

using System;

namespace P1

{

class MyFirstClass

{

public static void Main()

{

Console.WriteLine("HiAll");

Console.ReadKey();

return;

}

}

}

2)

using System;

namespace Cant

{

public class Cant

{

public static void Main()

{

int a;

a = 99; //without this following line will raise an error

Console.WriteLine("Value is: {0}",a);

Console.ReadKey();

}

}

}

3)

using System;

namespace P1

{

class Scope1

{

public static void Main()

{

for(int i=0;i<5;i++)

{

Console.WriteLine(i);

}

//i goes out of Scope here

for(int i=4;i>=0;i--)

{

Console.WriteLine(i);

}

}

}

}

4)

using System;

namespace P1

{

class Scope2

{

public static void Main()

{

int j;

for(int i=0;i<15;i++)

{

int j;

Console.WriteLine(i);

}

}

}

}

5)

using System;

namespace P1

{

public class Scope{

static int j = 430;

public static void Main()

{

int j =900;

Console.WriteLine(Scope.j);

}

}

}

6)

using System;

namespace P1

{

public class Const

{

public static void Main()

{

const double bonusPercent = 0.51;

int sal = 3000;

int bonus = (int)(sal \* bonusPercent);

Console.WriteLine(bonus);

}

}

}

7)

using System;

namespace P1

{

public class Vector

{

public int value;

}

public class DataTypes

{

public static void Main()

{

int i;

int j;

i = 77;

j = i;

Console.WriteLine("i is {0} and j is {1}", i, j);

j = 20;

Console.WriteLine("i is {0} and j is {1}", i, j);

Vector x,y;

x = new Vector();

x.value = 33;

y = x;

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

y.value = 24;

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

}

}

}

8)

using System;

namespace P1

{

class IntType

{

public static void Main()

{

//Signed Variables

sbyte sb = 33;

short s =33 ;

int i = 33;

long l = 33L;

//Unsigned Variables

byte b = 33;

ushort us = 33;

uint ui = 33U;

ulong ul = 33UL;

us = (ushort)ul;

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

}

}

}

9)

using System;

namespace P1

{

public class Floatting

{

public static void Main()

{

float f = 0.123456789F;

double d = 0.112233445566778899;

decimal dec = 11223344.1112223334445556667778889999M;

f = (float)d;

Console.WriteLine("f is {0} and d is {1} and dec is {2}", f, d, dec);

}

}

}

10)

using System;

1

namespace P1

{

public class Boolean

{

public static void Main()

{

bool status = true;

Console.WriteLine(status);

}

}

}

11)

using System;

namespace P1

{

public class Char

{

public static void Main()

{

char c = 'a';

Console.WriteLine(\a);

}

}

}

**PRACTICAL-2**

# AIM-GTU QUESTIONS.

## Program 1

Perform following programs in c#.

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;

using System.Threading.Tasks;

namespace practical2

{

class Program

{

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

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

1

1 2

1 2 3

1 2 3 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace practical2.\_1

{

class Program

{

static void Main(string[] args)

{

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

{

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

{

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

}

Console.WriteLine("");

}

Console.ReadKey();

}

}

}

## Program 3

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;

using System.Threading.Tasks;

namespace practical2.\_2

{

class Program

{

static void Main(string[] args)

{

string name;

string country;

Console.WriteLine("enter your name:");

name=Console.ReadLine();

Console.WriteLine("enter your country:");

country = Console.ReadLine();

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

Console.ReadKey();

}

}

}

## Program 4

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 practical2.\_3

{

class car

{

public void Method1()

{

Console.WriteLine("this is the method of car class");

}

}

class maruti:car

{

public void method2()

{

Console.WriteLine("this is the method of maruti");

Console.ReadKey();

}

}

class mahindra:car

{

public void method3()

{

Console.WriteLine("this is the method of mahindra");

}

}

class Program

{

static void Main(string[] args)

{

mahindra m = new mahindra();

maruti m1 = new maruti();

m.Method1();

m1.Method1();

Console.ReadKey();

}

}

**PRACTICAL-3**

# AIM-METHOD AND CONSTRUCTOR 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 p3

{

public class Add

{

public void add()

{

int[,] m1 = new int[50, 50];

int[,] m2 = new int[50, 50];

int[,] m3 = new int[50, 50];

Console.WriteLine("enter size of array:");

int size = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter first array:");

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

{

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

{

m1[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine("enter second array:");

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

{

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

{

m2[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

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

{

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

{

m3[i, j] = m1[i, j] + m2[i, j];

}

}

Console.WriteLine("addition array:");

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

{

Console.Write("\n");

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

{

Console.Write("{0}\t", m3[i, j]);

}

Console.Write("\n");

}

}

public int add(int a, int b)

{

return (a + b);

}

}

public class Vector

{

public void add()

{

Console.WriteLine("enter first vector");

int x = Convert.ToInt32(Console.ReadLine());

int y = Convert.ToInt32(Console.ReadLine());

int z = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("enter second vector");

int x1 = Convert.ToInt32(Console.ReadLine());

int y1 = Convert.ToInt32(Console.ReadLine());

int z1 = Convert.ToInt32(Console.ReadLine());

int x2 = x + x1;

int y2 = y + y1;

int z2 = z + z1;

Console.WriteLine("<" + x2 + "," + y2 + "," + z2 + ">");

}

}

class Program

{

static void Main(string[] args)

{

Add a1 = new Add();

Vector v1 = new Vector();

v1.add();

a1.add();

int res=a1.add(1, 2);

Console.Write("method overloading for addtion{0}",res);

Console.ReadLine();

}

}

}

## Program 2

Write a c# program that create student object. Overload constror 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;

using System.Reflection;

namespace p3a1

{

class Program

{

public int ID { get; set; }

public string Name { get; set; }

String name, branch;

int enrol;

public Program(String name)

{

this.name = name;

Console.WriteLine("constructor 1:" + name);

}

public Program(String name, int enrol)

{

this.name = name;

this.enrol = enrol;

Console.WriteLine("constructor 2:" + name + " " + enrol);

}

public Program(String name, int enrol, String branch)

{

this.name = name;

this.enrol = enrol;

this.branch = branch;

Console.WriteLine("constructor 3:" + name + " " + enrol + " " + branch);

}

static void Main(string[] args)

{

Program p1 = new Program("bob");

Program p2 = new Program("bob", 1);

Program p3 = new Program("bob", 1, "computer");

Console.ReadLine();

}

}

}

**PRACTICAL-4**

# AIM: REFLECTION

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Reflection;

namespace p3a1

{

class Program

{

public int ID { get; set; }

public string Name { get; set; }

String name, branch;

int enrol;

public void printID()

{

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

}

public void printName()

{

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

}

public Program(String name)

{

this.name = name;

Console.WriteLine("constructor 1:" + name);

}

public Program(String name, int enrol)

{

this.name = name;

this.enrol = enrol;

Console.WriteLine("constructor 2:" + name + " " + enrol);

}

public Program(String name, int enrol, String branch)

{

this.name = name;

this.enrol = enrol;

this.branch = branch;

Console.WriteLine("constructor 3:" + name + " " + enrol + " " + branch);

}

static void Main(string[] args)

{

try

{

Type T = Type.GetType("p3a1.Program");

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());

}

Program p1 = new Program("bob");

Program p2 = new Program("bob", 1);

Program p3 = new Program("bob", 1, "computer");

Console.ReadLine();

catch (Exception e)

{

Console.WriteLine(e);

Console.ReadLine();

}

}

}

}

**PRACTICAL-5**

Aim:

File Handling

Program 1: Write a C# program to copy data from one file to another using StreamReader and StreamWriter class.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Practical5

{

class Program

{

static void Main(string[] args)

{

CopyFile cp = new CopyFile();

String file1= @"C:\dotNet\f1.txt";

String file2 = @"C:\dotNet\radhika\f2.txt";

cp.copyFile(f1, f2);

}

}

public class CopyFile

{

public void copyFile(String f1,String f2)

{

using (StreamReader reader = new StreamReader(f2))

{

using (StreamWriter writer = new StreamWriter(f2))

{

String line = null;

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

{

writer.WriteLine(line);

}

}

}

}

}

}

Output:

f1.txt: Hello world...

f2.txt: Hello vvp...

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 {

public class CopyFile

{

public void copyFile(string f1, string f2)

{

using (StreamReader reader = new StreamReader(f1))

using (StreamWriter writer = new StreamWriter(f2))

{

string line = null;

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

writer.WriteLine(line);

}

}

}

public class main

{

public static void Main()

{

CopyFile cp = new CopyFile();

string f1 = @"E:\Sem-6\VS\p2\p2\f1.txt";

string f2 = @"E:\Sem-6\VS\p2\p2\f2.txt";

cp.copyFile(f1,f2);

}

}

}

Output:

F1.txt:

Hello vvp.....

How are you???

F2.txt: Hello World.....

hii

how are you ???

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();

}

}

}

Output:

E:\Sem-6\VS\p2\p2>P4.3.exe

E:\Sem-6\VS\P1-master E:\Sem-6\VS\p2

E:\Sem-6\VS\Assignment.docx E:\Sem-6\VS\C# word.txt

E:\Sem-6\VS\Doc1.docx E:\Sem-6\VS\P1-master.zip

E:\Sem-6\VS\p1.cs

E:\Sem-6\VS\p1.exe E:\Sem-6\VS\VS.docx

E:\Sem-6\VS\~$VS.docx

**PRACTICAL-6**

Aim:

Windows Form Application

Program: Create Windows Form Application for Student Registration and store student Details in Database.

**Form.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;

using System.IO;

namespace StudentForm {

public partial class Form1 : Form

{   
string imgPath;

public Form1()

{

InitializeComponent();

}

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

{

string gen = null;

string subject = null;

if (genMale.Checked == true)

{

gen = "m";

}

if (genFemale.Checked == true)

{

gen = "f";

}

if (ck1.Checked == true)

{

subject = subject + " s1";

}

if (ck2.Checked == true)

{

subject = subject + " s2";

}

string source = @"Data Source=radhika odedra\SQLExpress;

Initial Catalog=DemoDb;

Integrated Security=True;Pooling=False";

string insert = "insert into tblstudent (fname,lname,gender,subject,imgStudent) values ('" + txtfname.Text + "','" + txtlname.Text + "','" + gen + "','" + subject + "','" + (imgPath == null ? "" : imgPath) + "')"; //MessageBox.Show(insert);

SqlCommand cmd = new SqlCommand(insert,conn); conn.Open(); int i = cmd.ExecuteNonQuery(); conn.Close(); Console.WriteLine("Success....");

}

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

}

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

{

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

if (openFileDialog1.ShowDialog() == DialogResult.OK) { imgPath = openFileDialog1.SafeFileName; pictureBox.Image = Image.FromFile(openFileDialog1.FileName); //MessageBox.Show(imgPath); } }

}

}

**PRACTICAL-7**

**Aim:**

**ASP.NET Validation Control**

<%@ Page Title="Home Page" Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebApplication2.\_Default" %>

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

<div> <table> <tr> <td> <asp:Label runat="server" Text="Name"></asp:Label>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs p;

<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>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs p;&nbsp;

<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>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<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>

&nbsp;&nbsp;&nbsp;

<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>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs p;&nbsp;&nbsp;

<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**

Aim:

Introduction To Master Pages

Site1.Master:

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs" Inherits="WebApplication1.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"> <headrunat="server">

<title></title>

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

<style type="text/css">

.style1

{

width: 97px; height: 141px;

}

.style2

{

width: 97px;

height: 105px;

}

.style3  
 {

width: 97px;

height: 99px;

} .style4

{

width: 9px;

}

</style>

</head>

<body>

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

<table height="50%" width="50%">

<tr>

<td class="style2" colspan="2">

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

</asp:Label>

</td>

</tr>

<tr>

<td class="style4">

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

<asp:TextBox ID="txtsearch" runat="server"></asp:TextBox> </td> <td class="style3">

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server"> content page </asp:ContentPlaceHolder>

</td>

</tr>

<tr>

<td class="style1" colspan="2">

<asp:Label ID="lblfooter" runat="server" Text="Footer">

</asp:Label>

</td>

</tr>

</table>

</form>

</body>

</html>