Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
(Deemed to be University)
Maduravoyal, Chennai - 600 095. Tamilnadu. India.
(An ISO 9001 : 2015 Certified Institution)

SUBJECT NAME: DOT NET LAB
SUBJECT CODE: BCS17L12
PROGRAMME NAME: B. TECH- CSE
REGULATION: 2017

STAFF MANUAL

LIST OF EXPERIMENTS

Experiment Number: 1(i)

FIBONACCI SERIES

AIM:

To write a C# program to generate the Fibonacci series.

ALGORITHM:

1. Start the program.
2. Enter the limit.
3. Assign the value for the limit.
4. Run loop from 0 to limit.
5. Print the value & swap the value.
6. Stop the program.

PROGRAM: (FIBONACCI SERIES)

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
namespacefibonacci
{
class Program
    {
static void Main(string[] args)
        {
int f1 = -1, f2 = 2, f3, i;
Console.Write("Enter the fibonacci series:");
int n = int.Parse(Console.ReadLine());
Console.WriteLine("The Series is:");
for (i = 0; i< n; i++)
```

```
            {
                f3 = f1 + f2;
                f1 = f2;
                f2 = f3;
Console.Write(f3 + " ");
            }
Console.ReadLine();
        }
    }
}
```

OUTPUT:

Enter the Fibonacci Series: 5

The Series is:

1 2 3 5 8

RESULT:

Hence, the program Fibonacci series is executed & output is verified.

Experiment Number: 1 (ii)

FACTORIAL NUMBER

AIM:

To write a C# program to find factorial of a given number.

ALGORITHM:

1. Start the program.
2. Enter the number whose factorial has to be found.
3. Assign to a variable.
4. Declare a fact variable and assign 1.
5. Run for loop from 1 to number and multiply fact=fact*i.
6. Print the value as per the variable assign.
7. Stop the program.

PROGRAM:(FACTORIAL)

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
namespace Factorial
{
class Program
    {
static void Main(string[] args)
        {
long fact = 1;
Console.Write("Enter the no to find factorial:");
int n = int.Parse(Console.ReadLine());
for (inti = 2; i<= n; i++)
```

```
            {
fact = fact * i;
            }
Console.WriteLine("₩n The factorial is:");
Console.WriteLine(n+"!="+fact);
Console.ReadLine();
        }
    }
}
```

OUTPUT:

Enter the Number to get Factorial: 4

The Factorial of: 4! =24

RESULT:

Hence, the program factorial number is executed & output is verified.

Experiment Number: 2 (i)

COMPLEX NUMBER

AIM:

To write a C# program for complex number.

ALGORITHM:

1. Start the program.
2. Declare the class complex.
3. Write a function to overload +operator.
4. Declare two objects of the complex class.
5. Assign real & imaginary value to both objects.
6. Add both objects using +operator.
7. Display the value.
8. Stop the program.

PROGRAM:(COMPLEX NUMBER)

```
using System;
class Complex
{
double x;
double y;
   public Complex()
    {   }
    public Complex(double real, double img)
```

```
    {
        x = real;
        y = img;
    }
public static Complex operator +(Complex c1, Complex c2)
    {
        Complex c3 = new Complex();
        c3.x = c1.x + c2.x;
        c3.y = c1.y + c2.y;
return (c3);
    }
public void display()
    {
Console.Write(x);
Console.Write("+j" + y);
Console.WriteLine();
    }
}
classComplextest
{
public static void Main()
    {
        Complex a, b, c;
```

```
        a =new Complex (2.5, 3.5);
        b =new Complex (1.6, 2.7);
c = a + b;
Console.Write("a=");
a.display();
Console.Write("b=");
b.display();
Console.Write("c=");
c.display();
Console.ReadLine();
    }
}
```

OUTPUT:

```
First Complex Number: 2.5 + j3.5
Second Complex Number: 1.6 + j2.7
The Sum Of The Two Complex Numbers: 4.1 + j6.2
```

RESULT:

Hence, the program complex number is executed & output is verified.

Experiment Number: 2 (ii)

MATRIX ADDITION

AIM:

To write a C# program for matrix addition.

ALGORITHM:

1. Start the program.
2. Create a class matrix.
3. Write a function to overload + operator.
4. Create two object of class matrix.
5. Take the matrix value that is randomly generated.
6. Add the two matrix using + operator.
7. Display the matrix.
8. Stop the program.

PROGRAM: (MATRIX ADDITION)

```
using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

namespace matrix

{

   class Program

    {

      static void Main(string[] args)
```

```
    {
     int i, j;
     int[,] a = new int[2, 2];
     int[,] b = new int[2, 2];
     int[,] c = new int[2, 2];
     Console.WriteLine("Enter the first matrix:");
     for (i = 0; i< 2; i++)
        {
         for (j = 0; j < 2; j++)
            {
         a[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
  Console.WriteLine("Enter the Second matrix:");
   for (i = 0; i< 2; i++)
   {
     for (j = 0; j < 2; j++)
      {
         b[i, j] = Convert.ToInt32(Console.ReadLine());
      }
   }
Console.WriteLine("Resultant of two matrix is:");
   for (i = 0; i< 2; i++)
```

```
    {
       for (j = 0; j < 2; j++)
      {
         c[i, j] = a[i, j] + b[i, j];
         Console.WriteLine(+c[i, j]);
        }
   }
     Console.ReadLine();
        }
    }
}
```

OUTPUT:

Enter the First Matrix:

1

3

5

7

Enter the Second Matrix:

2

4

6

8

Resultant of Two Matrix is:

3

7

11

15

RESULT:

Hence, the program matrix addition is executed & output is verified.

Experiment Number: 3(i)

STUDENT STATUS
(MULTIPLE INHERITANCE)

AIM:

To write a C# program to find the student information.

ALGORITHM:

1. Start the program.
2. Create a structure with student name, Reno, dept.
3. Ask the information from the user.
4. Display the information given by the user.
5. Stop the program.

PROGRAM

```
using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

namespacestudentlist

{

struct student

    {

public string name;

publicintrollnumber;

public string dept;

publicint mark;

publicint total, avg;
```

```
    }
class Program
    {
static void Main(string[] args)
        {
student s;
int total = 0;
Console.Write("Enter the Name:");
            s.name = Console.ReadLine();
Console.Write("Enter the Register Number:");
s.rollnumber = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the Dept.:");
s.dept = Console.ReadLine();
int[] mark = new int[5];
Console.WriteLine("Enter the 5 Subject Marks");
for (inti = 0; i< 5; i++)
            {
mark[i] = Convert.ToInt32(Console.ReadLine());
total = total + mark[i];
            }
Console.WriteLine("Name:" + s.name);
Console.WriteLine("Register Number:" + s.rollnumber);
Console.WriteLine("Dept.:" + s.dept);
```

```
Console.WriteLine("Total Marks:" + total);

Console.WriteLine("Average:" + total / 5);

Console.ReadLine();

        }

    }

}
```

OUTPUT:

```
Enter the Name:Vicky

Enter the Register Number: 91061101117

Enter the Dept.: CSE

Enter the 5 Subject Marks:

88
86
92
91
85
Name: Vicky
Register Number:91061101117

Dept.: CSE

Total Marks: 442

Average: 88
```

RESULT:

Hence, the program student status is executed & output is verified.

Experiment Number: 3 (ii)

AREA OF AN OBJECT (MULTIPLE INHERITANCE)

AIM:

To write a C# program to find area of an object using multiple inheritance.

ALGORITHM:

1. Start the program.
2. Create an interface Area and declare a method compute.
3. Define two class Square and Circle implementing Area.
4. In classes define the method of interface.
5. Perform the necessary calculation.
6. Display the result as per the given value.
7. Stop the program.

PROGRAM: (AREA OF AN OBJECT)

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
namespaceareaofanobject
{
   interface Area
    {
      double Compute(double x);
     }
  classSquare:Area
  {
    public double Compute(double x)
    {
       return(x*x);
     }
```

```
  }
classCircle:Area
{
    public double Compute(double x)
    {
      return(Math.PI *x*x);
     }
 }
class Program
    {
     static void Main(string[] args)
        {
            Square sqr=new Square();
            Circle cr=new Circle();
            Area area=(Area)sqr;
            Console.WriteLine("Area of Square="+area.Compute(10));
            area=(Area)cr;
            Console.WriteLine("Area of Circle="+ area.Compute(10));
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

Area of Square = 100

Area of Circle = 314.159265358979

RESULT:

Hence, the program student status is executed & output is verified.

Experiment Number: 4(i)

ENUMERATOR

AIM:

To write a C# program to implement enum type.

ALGORITHM:

1. Start the program.
2. Declare the enum as shape.
3. Use the SWITCH CASE method for getting area of square & circle.
4. Default be the invalid input.
5. Declare the class enumtest.
6. Display the output.
7. Stop the program.

PROGRAM: (ENUMERATOR)

```
using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

namespace enumerator

{

class Area

    {

public enum Shape

        {
```

```
circle,
square
        }
public void areashape(int x, Shape shape)
        {
double area;
switch (shape)
            {
caseShape.circle:
area = Math.PI * x * x;
Console.WriteLine("circle area=" + area);
break;
caseShape.square:
area = x * x;
Console.WriteLine("square area=" + area);
break;
            }
        }
    }
class enumtest
    {
static void Main(string[] args)
        {
```

```
            Area area = new Area();
area.areashape(15, Area.Shape.circle);
area.areashape(15, Area.Shape.square);
area.areashape(15, (Area.Shape)1);
area.areashape(15, (Area.Shape)10);
Console.ReadLine();
        }
    }
}
```

OUTPUT:

Circle area = 706.858347057703

Square area = 225

RESULT:

Hence, the program enumerator is executed & output is verified.

Experiment: 4 (ii)

STRUCTURE

AIM:

To write a C# program to implement structure.

ALGORITHM:

1. Start the program.
2. Construct a structure employee.
3. Give the id & salary for the output.
4. Get the void display for the result.
5. Use the structure as new employee.
6. Stop the program.

PROGRAM: (STRUCTURE)

```
using System;
namespace @struct
{
class Program
   {
     publicstruct employee
       {
         int id;
         double salary;
          publicemployee(int id, double salary)
            {
                this.id = id;
                 this.salary = salary;
            }
publicemployee(int id, int salary)
 {
   this.id = id;
    this.salary = 3400.00;
 }
publicemployee(employee x)
 {
```

```
 this.id = x.id;
this.salary = x.salary;
  }
public void DisplayValues()
 {
Console.WriteLine("Structure");
Console.WriteLine("ID: " + this.id.ToString());
Console.WriteLine("Salary : " + this.salary.ToString());
  }
   }
static void Main(string[] args)
{
employeeemp = new employee(12,4560.00);
emp.DisplayValues();
Console.ReadLine();
  }
    }
}
```

OUTPUT:

Structure
ID: 12
Salary: 4560

RESULT:

Hence, the program structureis executed & output is verified.

Experiment Number: 5

DESIGN A CALCULATOR

AIM:

To write a VB.net program to create a calculator.

ALGORITHM:

1. Start the program.
2. Create the GUI for the user.
3. Let the user enter two values in a Textbox.
4. Calculate the value based on the function selected by the user by clicking button.
5. Store the calculated value in a variable.
6. Display the resulted value in the result textbox when user click on = button.
7. Stop the program.

PROGRAM: (CALCULATOR)

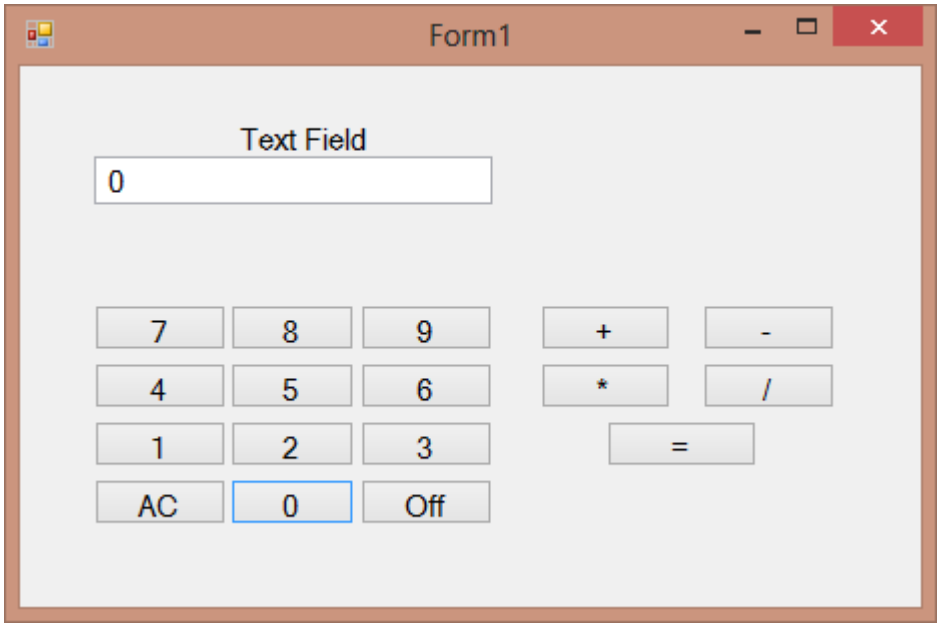
```
Public Class Calculator
    Inherits System.Windows.Forms.Form
    Dim num1 As Double
    Dim num2 As Double
    Dim result As Double
    Dim add As Boolean
    Dim sb As Boolean
    Dim mul As Boolean
    Dim div As Boolean
    Private Sub Button13_Click(sender As Object, e As EventArgs) Handles Button13.Click
mul = True
        num2 = num1
        num1 = 0
        TextBox1.Text = " "
    End Sub
    Private Sub Button17_Click(sender As Object, e As EventArgs) Handles Button17.Click
        End
    End Sub
    Private Sub Calculator_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        TextBox1.Text = " "
```

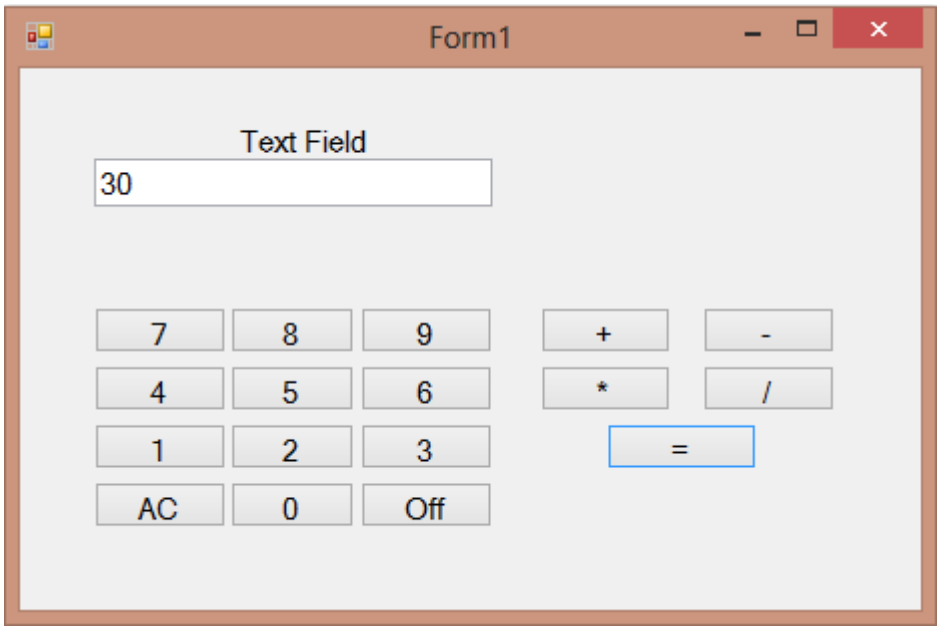
```
add = sb = mul = div = False
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        TextBox1.Text = TextBox1.Text + Button1.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        TextBox1.Text = TextBox1.Text + Button2.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        TextBox1.Text = TextBox1.Text + Button3.Text
        num1 = TextBox1.Text
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles

Button4.Click
        TextBox1.Text = TextBox1.Text + Button4.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        TextBox1.Text = TextBox1.Text + Button5.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        TextBox1.Text = TextBox1.Text + Button6.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
        TextBox1.Text = TextBox1.Text + Button7.Text
        num1 = TextBox1.Text
    End Sub
```

```
    Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
        TextBox1.Text = TextBox1.Text + Button8.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
        TextBox1.Text = TextBox1.Text + Button9.Text
        num1 = TextBox1.Text
    End Sub
    Private Sub Button14_Click(sender As Object, e As EventArgs) Handles Button14.Click
div = True
        num2 = num1
        num1 = 0
        TextBox1.Text = " "
    End Sub
    Private Sub Button12_Click(sender As Object, e As EventArgs) Handles Button12.Click
sb = True
        num2 = num1
        num1 = 0
        TextBox1.Text = " "
    End Sub
    Private Sub Button11_Click(sender As Object, e As EventArgs) Handles

Button11.Click
add = True
        num2 = num1
        TextBox1.Text = " "
    End Sub
    Private Sub Button15_Click(sender As Object, e As EventArgs) Handles Button15.Click
        If add Then
result = num1 + num2
        End If
        If sb Then
result = num2 - num1
```

```
        End If
        If mul Then
result = num1 * num2
        End If
        If div Then
result = num2 / num1
        End If
        TextBox1.Text = result
        num1 = result
    End Sub
    Private Sub Button16_Click(sender As Object, e As EventArgs) Handles Button16.Click
        TextBox1.Text = " "
        num1 = 0
result = 0
add = False
sb = False
mul = False
div = False
        num2 = 0
    End Sub
    Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
        TextBox1.Text = TextBox1.Text + Button10.Text
        num1 = TextBox1.Text
    End Sub
End Class
```

OUTPUT:





RESULT:

Hence, the program calculator is executed & output is verified.

Experiment Number: 6(i)

EMPLOYEE DETAILS

AIM:

To write a VB.NET program to find the net salary of employee.

ALGORITHM:

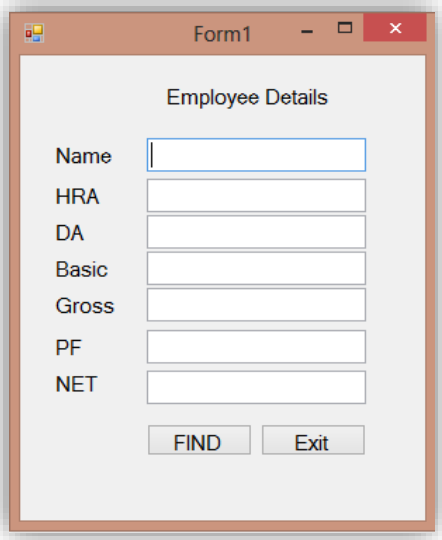
1. Start the program.
2. Create the GUI for the user.
3. Design some label, textbox & button.
4. After giving values by the user, system will find the gross &net salary of that employee.
5. It provides message-box which gives the salary details.
6. After clicking end button the GUI will be exit.
7. Stop the program.

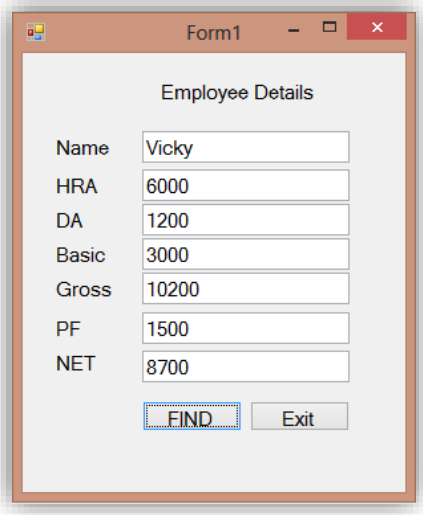
PROGRAM: (EMPLOYEE DETAILS)

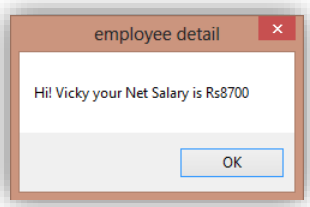
```
Public Class Form1
    Private Sub Label5_Click(sender As Object, e As EventArgs) Handles Label5.Click
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        TextBox7.Text = (Val(TextBox5.Text) - (Val(TextBox6.Text)))
MsgBox("Hi! " & TextBox1.Text &" your Net Salary is Rs" & TextBox7.Text)
    End Sub
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    End Sub
    Private Sub TextBox5_TextChanged(sender As Object, e As EventArgs) Handles
TextBox5.TextChanged
        TextBox5.Text = (Val(TextBox2.Text) + (Val(TextBox3.Text) + (Val(TextBox4.Text))))
    End Sub
```

```
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End
    End Sub
End Class
```

OUTPUT:







RESULT:

Hence, the program employee detail is executed & output is verified.

Experiment number: 6 (ii)

VOTERS
(EXCEPTION HANDLING)

AIM:

To write a C# program for voters through exception handling.

ALGORITHM:

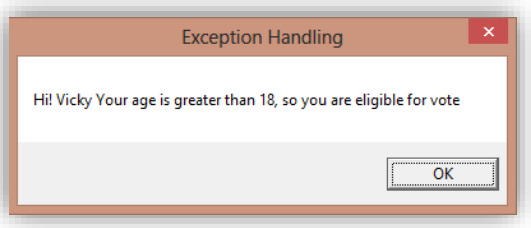
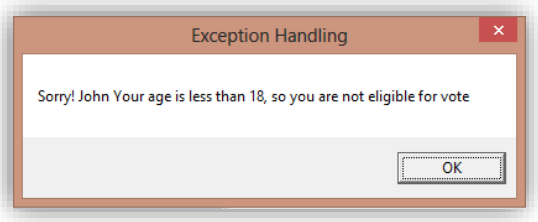
1. Start the program.
2. Declare the name & age its data-type.
3. Use the try-catch method to getting the result.
4. Use if method for checking the age.
5. Get the message-box for getting the output as the user are eligible for vote or not.
6. Stop the program.

PROGRAM: (VOTERS)

```
Module Module
  Sub Main()
      Dim vname As String
      Dim age As Integer
      Try
Console.Write("Enter your Name:")
vname = Console.ReadLine()
Console.Write("Enter your Age:")
age = Int32.Parse(Console.ReadLine())
          If (age >= 18) Then
MsgBox("Hi " &vname& " Your age is greater than 18, so you are eligible for vote")
          Else
MsgBox("Sorry " &vname& " Your age is less than 18, so you are not eligible for vote")
```

```
End If
 Catch ex As Exception
Console.WriteLine("Exception is:" &ex.Message)
Console.ReadLine()
End Try
 End Sub
End Module
```

OUTPUT:

Enter your Name: Vicky Enter your Age: 22 	Enter your Name: John Enter your Age: 16 

RESULT:

Hence, the program - voters is executed & output is verified.

Experiment Number: 7

ADO.NET

AIM:

To write a Vb.net program to connect with the SQL server and perform insertion, updating, deletion using ADO.Net.

ALGORITHM:

1. Start the program.
2. Open the server explorer using view menu IDE.
3. Right click data connection node and create new database menu option.
4. Use windows authentication and specify the database name.
5. Right click the table node and select add new table e.g. table inventory.
6. Using the table editor add columns e.g. name,no.
7. Right click inventory table and select show table data.
8. Declare the connection and command object.
9. Paste the database connection string and set the connection string.
10. Open the connection and run the needed query.
11. Stop the program.

PROGRAM: (STUDENT RECORD)

```
Imports System.Data.SqlClient
Public Class Form1
    Dim sqlconn As New SqlConnection()
    Dim sqlcomm As New SqlCommand()
    Dim sqldaat As New SqlDataAdapter()
    Dim ds As New DataSet()
    Private Sub StdBindingNavigatorSaveItem_Click(sender As System.Object, e As
System.EventArgs) Handles StdBindingNavigatorSaveItem.Click
Me.Validate()
Me.StdBindingSource.EndEdit()
```

```
Me.TableAdapterManager.UpdateAll(Me.VickyDataSet)
    End Sub
    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the 'VickyDataSet2.std' table. You can
move, or remove it, as needed.
Me.StdTableAdapter1.Fill(Me.VickyDataSet.std)
    End Sub
    Private Sub insert_Click(sender As System.Object, e As System.EventArgs) Handles
Button1.Click
sqlconn = New SqlConnection("Data
Source=PRINCES\QLEXPRESS;InitialCatalog=vicky;Integrated Security=True")
sqlcomm = New SqlCommand("dbo.StoredProcedure1", sqlconn)
sqlcomm.CommandType = CommandType.StoredProcedure
sqlcomm.Parameters.AddWithValue("@name", TextBox1.Text)
sqlcomm.Parameters.AddWithValue("@regno", TextBox2.Text)
sqlcomm.Parameters.AddWithValue("@dept", TextBox3.Text)
sqlconn.Open()
sqlcomm.ExecuteNonQuery()
sqlconn.Close()
MessageBox.Show("inserted","Data Inserted")
    End Sub

    Private Sub delete_Click(sender As System.Object, e As System.EventArgs) Handles
Button2.Click
sqlconn = New SqlConnection("Data
Source=PRINCE\SQLEXPRESS;InitialCatalog=vicky;Integrated Security=True")
sqlcomm = New SqlCommand("dbo.StoredProcedure3", sqlconn)
sqlcomm.CommandType = CommandType.StoredProcedure
sqlcomm.Parameters.AddWithValue("@name", TextBox1.Text)
sqlconn.Open()
sqlcomm.ExecuteNonQuery()
sqlconn.Close()
```

```
MessageBox.Show("deleted","Record Deleted")
    End Sub
Private Sub update_Click(sender As System.Object, e As System.EventArgs) Handles
Button4.Click
sqlconn = New SqlConnection("Data
Source=PRINCE\SQLEXPRESS;InitialCatalog=vicky;Integrated Security=True")
sqlcomm = New SqlCommand("dbo.StoredProcedure5", sqlconn)
sqlcomm.CommandType = CommandType.StoredProcedure
sqlconn.Open()
sqlcomm.Parameters.AddWithValue("@name", TextBox1.Text)
sqlcomm.Parameters.AddWithValue("@regno", TextBox2.Text)
sqlcomm.Parameters.AddWithValue("@dept", TextBox3.Text)
        Try
sqlcomm.ExecuteNonQuery()
        Catch ex As SqlException
        Catch ex As Exception
        Finally
            If IsNothing(sqlcomm) = False Then
sqlcomm.Dispose()
sqlcomm = Nothing
            End If
sqlconn.Close()
        End Try
MessageBox.Show("Updated", "Record Updated")
    End Sub
End Class
   Private Sub select_Click(sender As System.Object, e As System.EventArgs) Handles
Button5.Click
sqlconn = New SqlConnection("Data
Source=PRINCE\SQLEXPRESS;InitialCatalog=vicky;Integrated Security=True")
sqlcomm = New SqlCommand("dbo.StoredProcedure6", sqlconn)
sqlcomm.CommandType = CommandType.StoredProcedure
sqlconn.Open()
```

```
sqldaat.SelectCommand = sqlcomm
sqldaat.Fill(ds, "dbo.StoredProcedure6")
sqlconn.Close()
    End Sub
Private Sub exit_Click(sender As System.Object, e As System.EventArgs) Handles
Button3.Click
        End
    End Sub
```

STORED PROCEDURE:

INSERT:

```
ALTER PROCEDURE dbo.StoredProcedure1
        @name nvarchar(50),@regnoint,@deptnvarchar(50)
AS
insert into std values(@name,@regno,@dept)
        RETURN
```

DELETE:

```
ALTER PROCEDURE dbo.StoredProcedure3
        @name nvarchar(50)
AS
delete from std where @name=name
        RETURN
```

UPDATE:

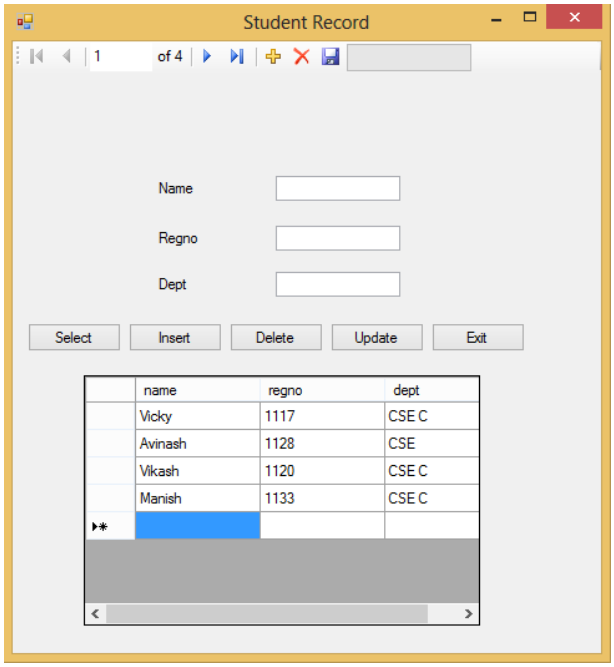
```
ALTER PROCEDURE dbo.StoredProcedure5
        @name nvarchar(50),@regnoint,@deptnvarchar(50)
AS
updatestd set name=@name,regno =@regno,dept=@dept where name=@name
        RETURN
```

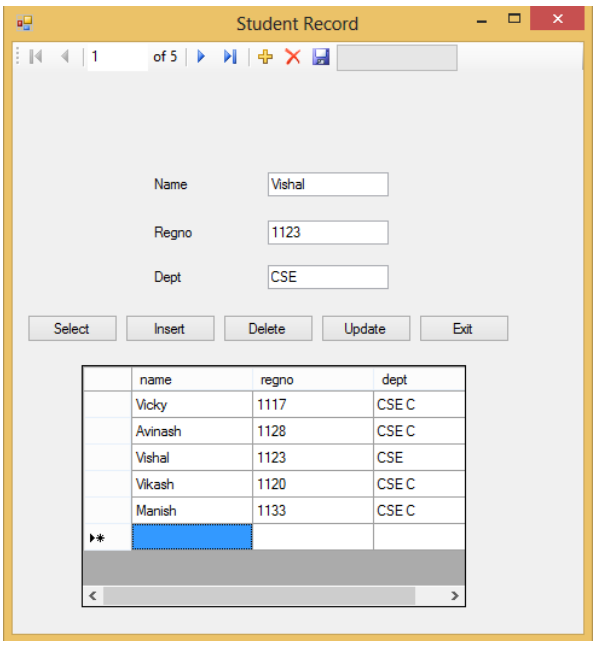
SELECT:

```
ALTER PROCEDURE dbo.StoredProcedure6
        @name nvarchar(50)
AS
select* from std where name=@name
```

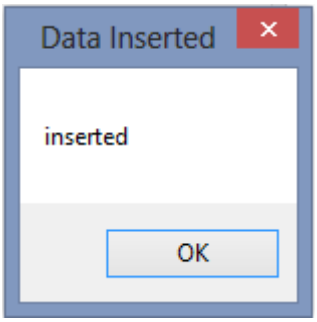
RETURN

OUTPUT:

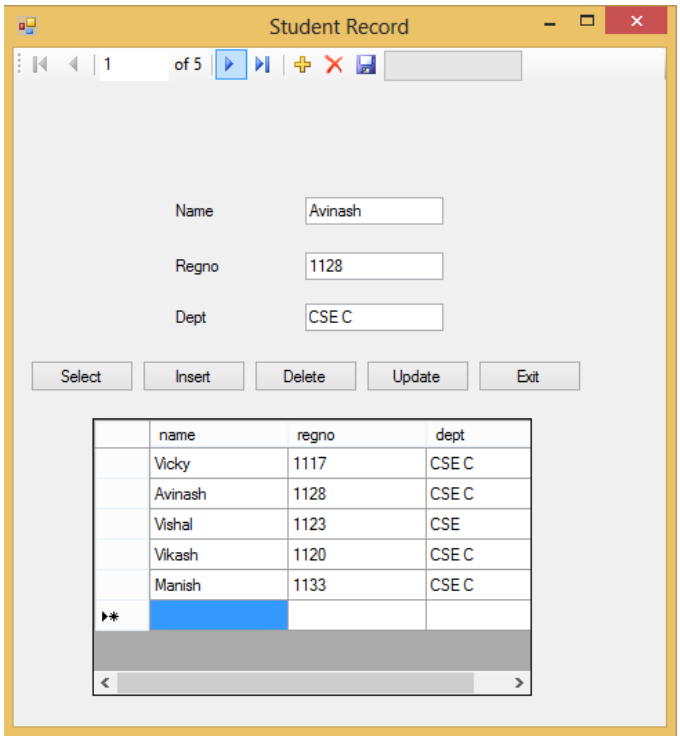


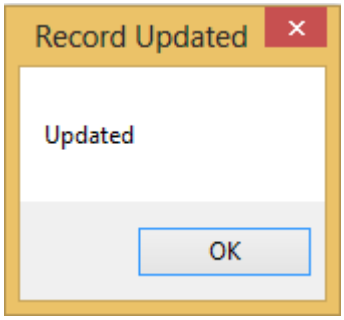


DATA INSERT:

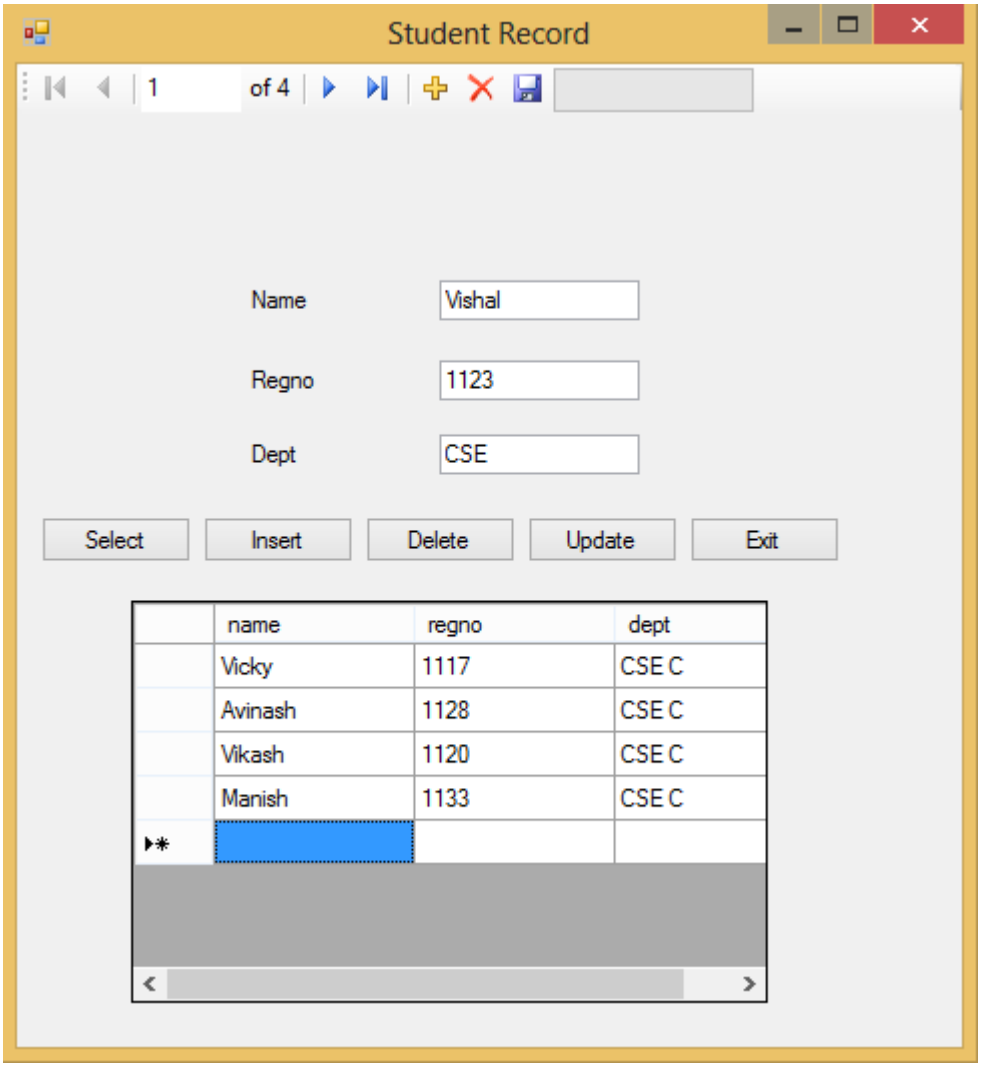


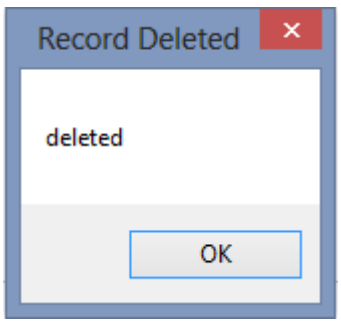
DATA UPDATE:


Student Record
1
of 5
Name
Avinash
Regno
1128
Dept
CSE C
Select
Insert
Delete
Update
Exit
name
regno
dept
Vicky
1117
CSE C
Avinash
1128
CSE C
Vishal
1123
CSE
Vikash
1120
CSE C
Manish
1133
CSE C


Record Updated
Updated
OK

DATA DELETE:





RESULT:

Hence, the program ado.net is executed & output is verified.

Experiment number: 8 (i)

ASP.NET
(PAYMENT DETAILS)

AIM:

To create a webpage using asp.net to find the gross amount.

ALGORITHM:

1. Start the program.
2. Create a dynamic web pages using html codes.
3. Design the label, text-box& button for getting the output.
4. Give the codes for each text-box & button.
5. After clicking the calculate button, we will get the gross amount, discount, & net amount of the employee.
6. Stop the program.

PROGRAM: PAYMENT DETAILS (ASP.NET)

HTML CODES:

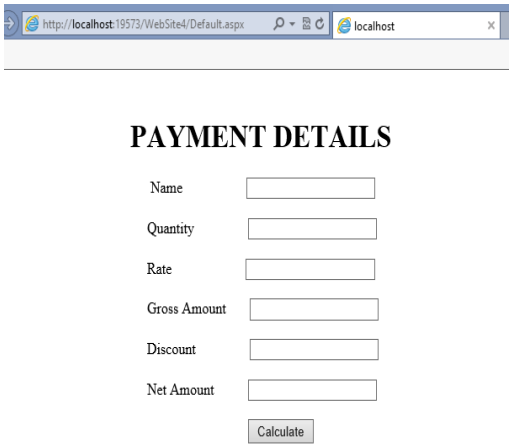
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!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>
<h1>PAYMENT DETAILS</h1>
```

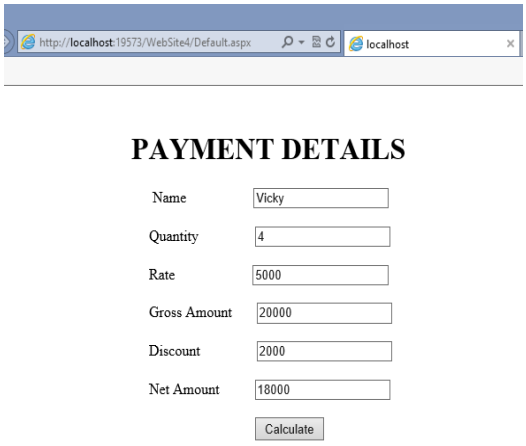
```
</div>
<div>
 Name<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<br />
<br />
Quantity<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
<br />
<br />
 Rate<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
<br />
<br />
Gross Amount<asp:TextBox ID="TextBox4"runat="server"></asp:TextBox>
<br />
<br />
        Discount<asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
<br />
<br />
 Net Amount <asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>
<br />
<br />
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
            Text="Calculate" />
</div>
</form>
</body>
</html>
```

C# CODE:

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

public partial class _Default : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
    {
    }
protected void Button1_Click(object sender, EventArgs e)
    {
int a, b, c;
float x, y;
        a = Convert.ToInt32(TextBox2.Text);
        b = Convert.ToInt32(TextBox3.Text);
        c = a * b;
        x = c * 10 / 100;
        y = c - x;
        TextBox4.Text = c.ToString();
        TextBox5.Text = x.ToString();
        TextBox6.Text = y.ToString();
    }
}
```

OUTPUT:





RESULT:

Hence, the asp.net program for employee salary is executed & output is verified.

Experiment Number: 8 (ii)

ASP.NET
(ATTENDANCE PERCENTAGE)

AIM:

To design a webpage using asp.net to find the student percentage

ALGORITHM:

1. Start the program.
2. Create a dynamic web pages using html codes.
3. Design the label, text-box & button.
4. Assign the codes for text-box & button.
5. After giving the values for no of total working days & present days, the output will be generated.
6. Stop the program.

PROGRAM: (ATTENDANCE PERCENTAGE)

HTML CODES:

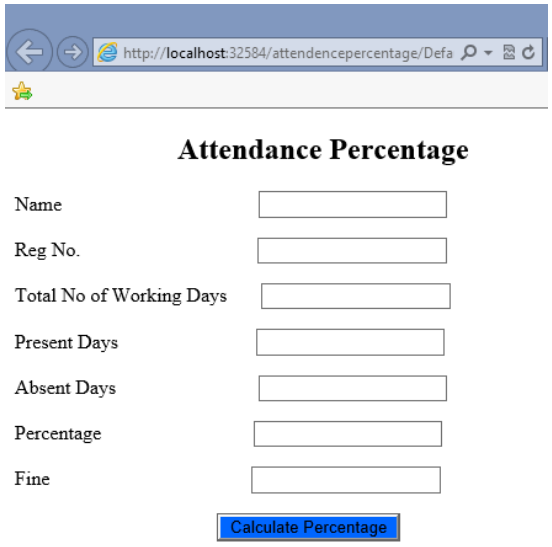
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<form id="form1" runat="server">
<div>
<h2><strong>Attendance Percentage </strong></h2>
<p>
<asp:TextBox ID="name" runat="server"></asp:TextBox>
</p>
<p>
```

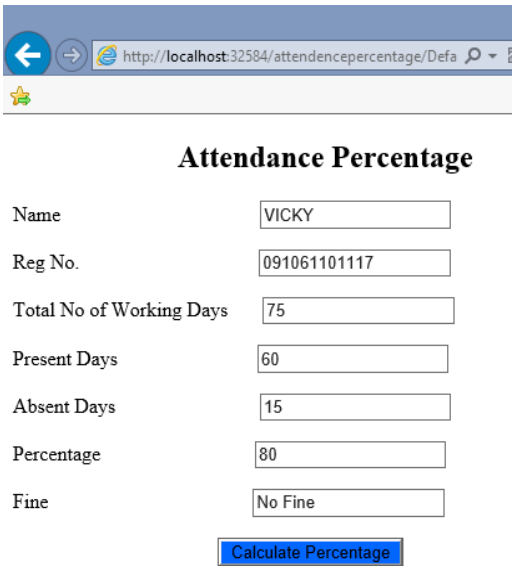
```
<asp:TextBox ID="regno" runat="server"></asp:TextBox>
</p>
<p
<asp:TextBox ID="total" runat="server"></asp:TextBox>
</p>
<p
<asp:TextBox ID="present" runat="server"></asp:TextBox>
</p>
<p
<asp:TextBox ID="absent" runat="server"></asp:TextBox>
</p>
<p
<asp:TextBox ID="percentage" runat="server"></asp:TextBox>
</p>
<p
<asp:TextBox ID="fine" runat="server"></asp:TextBox>
</p>
<p>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Calculate Percentage" />
</p>
</div>
</form>
</body>
</html>
```

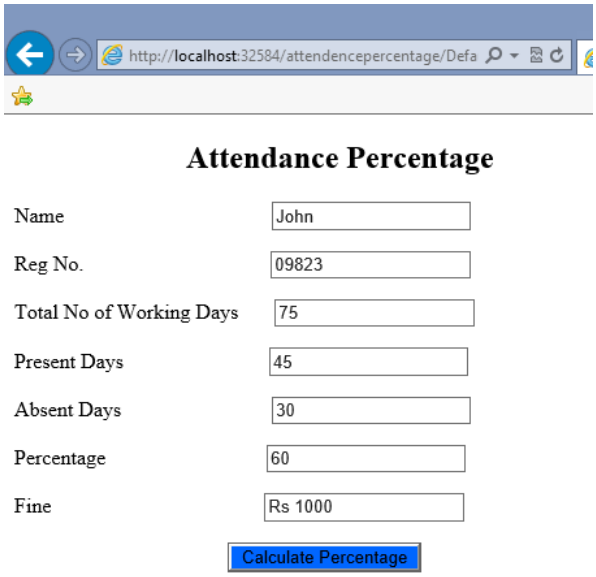
C# CODE:

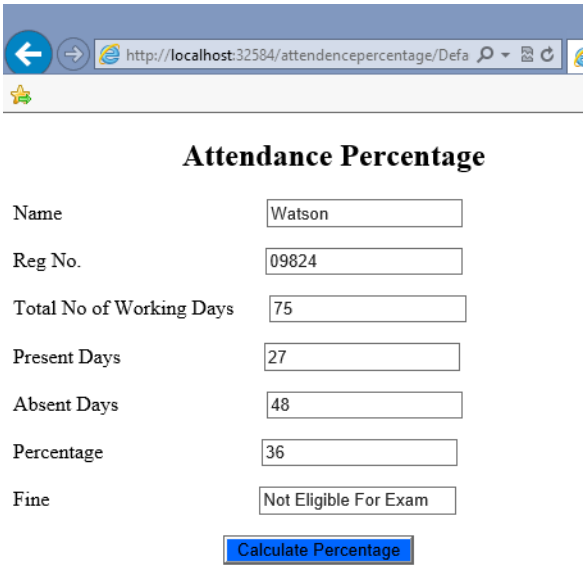
```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
protected void Button1_Click(object sender, EventArgs e)
    {
        int a, b,c ;
        float x;
        a = Convert.ToInt32(total.Text);
        b = Convert.ToInt32(present.Text);
        x = (float)b* 100 / a;
        c = a - b;
        percentage.Text = x.ToString();
        absent.Text = c.ToString();
        if (x >= 75)
        fine.Text = "No Fine";
        else if (x >= 60 && x < 75)
        fine.Text = "Rs 1000";
        else if (x >= 40 && x < 60)
        fine.Text = "Rs 2000";
        else
        fine.Text="Not Eligible for Exam";
    }
}
```

OUTPUT:









RESULT:

Hence, the asp.net program for finding the student attendance percentage is executed & output is verified.