



Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY

University with Graded Autonomy Status

(An ISO 21001 : 2018 Certified Institution)

Pertiyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.



RECORD NOTEBOOK

DOT NET LAB (BCS18L12)

2024-2025(ODD SEMESTER)

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

NAME : YELISETTI.GIRIJA

REGISTER NO : 211191101168

COURSE : B.TECH CSE-DS(AI)

YEAR/SEM/SEC : IV/VII/C



Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY

University with Graded Autonomy Status

(An ISO 21001 : 2018 Certified Institution)

Pertiyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.



BONAFIDE CERTIFICATE

REGISTER NO: 211191101168

NAME OF LAB : DOT NET LAB (BCS18L12)

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

Certified that this is the bonafide record of work done by **YELISETTI.GIRIJA** of **IV Year B.Tech CSE-DS(AI)**, Sec-‘C’ in the **DOT NET LAB(BCS18L12)** during the year **2024-2025**.

Signature of Lab-in-Charge

Signature of Head of Dept

Submitted for the Practical Examination held on _____

Internal Examiner

External Examiner

INDEX

Ex.No	Date	Name of the Experiment	Page No	Signature of the Staff
1		FIBONACCI SERIES		
2		FACTORIAL		
3		COMPLEX NUMBER		
4		MATRIX ADDITION		
5		STUDENT STATUS		
6		AREA OF AN OBJECT		
7		ENUMERATOR		
8		STRUCTURE		
9		CALCULATOR		
10		EMPLOYEE DETAILS		
11		VOTERS		
12		STUDENT RECORD		
13		PAYMENT DETAILS		
14		ATTENDANCE PERCENTAGE		

Ex.No	FIBONACCI SERIES	Date
1		

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:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace fibonacci
{
    class Program
    {
        static void Main(string[] args)
        {
            int f1 = -1, f2 = 2, f3, i;
            Console.WriteLine("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.WriteLine(f3 + " ");
            }
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

 C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Documents\DOT_NET>ex1i
Enter the number of fibonacci series : 22
The Series is:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946
```

RESULT:

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

Ex.No	FACTORIAL NUMBER	Date
2		

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:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.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:

```
C:\Users\Admin\Documents\DOT_NET>ex1ii
Enter the number to find factorial: 8

The factorial is:
8! = 40320
```

RESULT:

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

Ex.No	COMPLEX NUMBER	Date
3		

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:

```
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();
}
}
class Complextest
{
```

```
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:

```
C:\Users\Admin\Documents\DOT_NET>ex2i  
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.

Ex.No	MATRIX ADDITION	Date
4		

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:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.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:

```
C:\Users\Admin\Documents\DOT_NET>ex2ii
Enter the first matrix:
22
33
23
21
Enter the second matrix:
32
12
32
12
Resultant matrix is:
54 45
55 33
```

RESULT:

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

Ex.No	Student Status	Date
5		

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;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace studentlist
{
    struct student
    {
        public string name;
        public int rollnumber;
        public string dept;
        public int mark;
        public int 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:

C:\Windows\System32\cmd.exe

```
C:\Users\Admin\Documents\DOT_NET>ex3i
Enter the Name: Aakash K
Enter the Register Number: 1001
Enter the Dept.: DS(AI)
Enter the 5 Subject Marks:
95
96
98
99
98
Name: Aakash K
Register Number: 1001
Dept.: DS(AI)
Total Marks: 486
Average: 97
```

RESULT:

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

Ex.No	AREA OF AN OBJECT	Date
6		

AIM

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

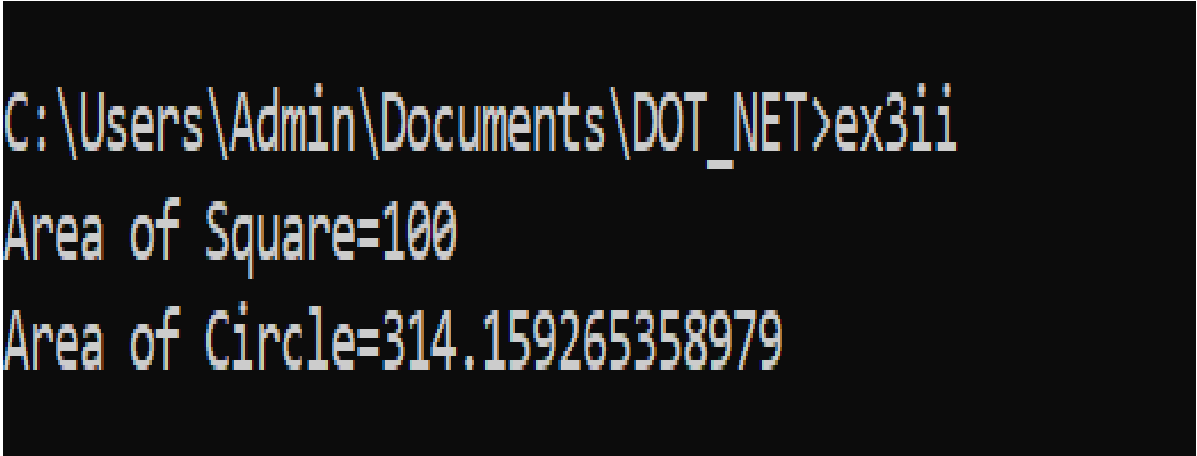
ALGORITHM

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:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace areaofanobject
{
    interface Area
    {
        double Compute(double x);
    }
    class Square:Area
    {
        public double Compute(double x)
        {
            return(x*x);
        }
    }
    class Circle: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:A screenshot of a Windows command prompt window with a black background and yellow text. The text shows the command 'ex3ii' being executed from the directory 'C:\Users\Admin\Documents\DOT_NET'. The output displays the area of a square as 100 and the area of a circle as 314.159265358979.

```
C:\Users\Admin\Documents\DOT_NET>ex3ii
Area of Square=100
Area of Circle=314.159265358979
```

RESULT:

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

Ex.No	ENUMERATOR	Date
7		

AIM

To write a C# program to implement enum type.

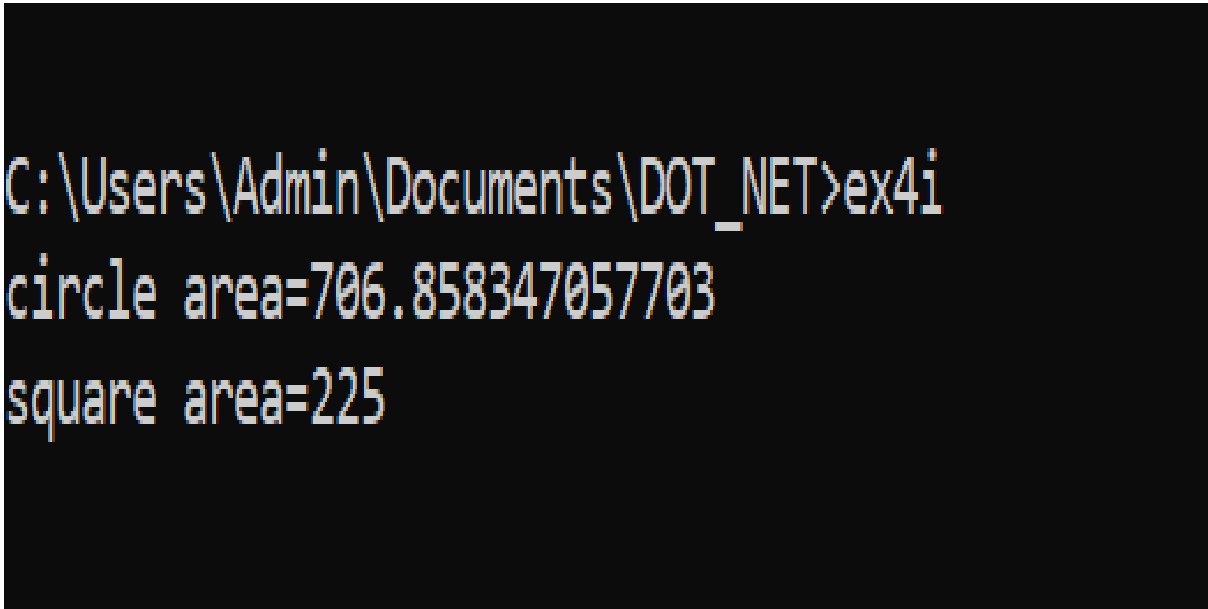
ALGORITHM

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:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace enumerator
{
    class Area
    {
        public enum Shape
        {
            circle,
            square
        }
        public void areashape(int x, Shape shape)
        {
            double area;
            switch (shape)
            {
                case Shape.circle:
                    area = Math.PI * x * x;
                    Console.WriteLine("circle area=" + area);
                    break;
                case Shape.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:A screenshot of a Windows command prompt window. The title bar is not visible. The background is black, and the text is displayed in a yellow, monospaced font. The text shows the current directory path, a command to run a program, and the resulting output for a circle and a square.

```
C:\Users\Admin\Documents\DOT_NET>ex4i  
circle area=706.858347057703  
square area=225
```

RESULT:

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

Ex.No	STRUCTURE	Date
8		

AIM

To write a C# program to implement structure.

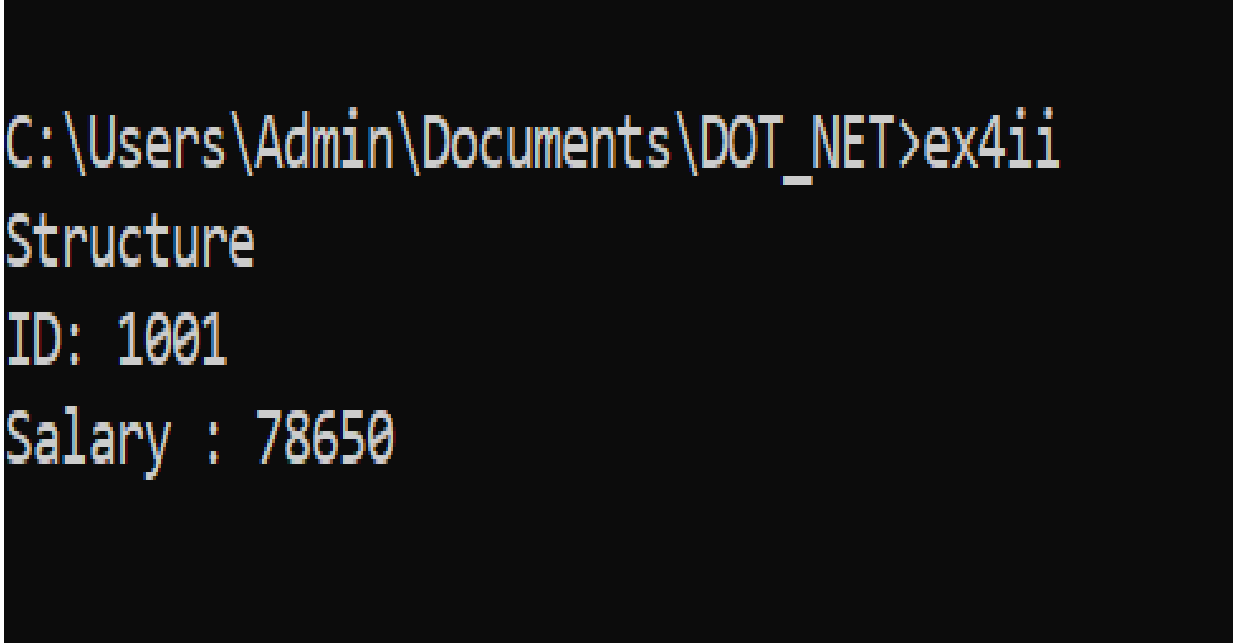
ALGORITHM

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:

```
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)  
    {  
        employee emp = new employee(12,4560.00);  
        emp.DisplayValues();  
        Console.ReadLine();  
    }  
}
```


OUTPUT:

```
C:\Users\Admin\Documents\DOT_NET>ex4ii
Structure
ID: 1001
Salary : 78650
```

RESULT:

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

Ex.No	CALCULATOR	Date
9		

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:

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

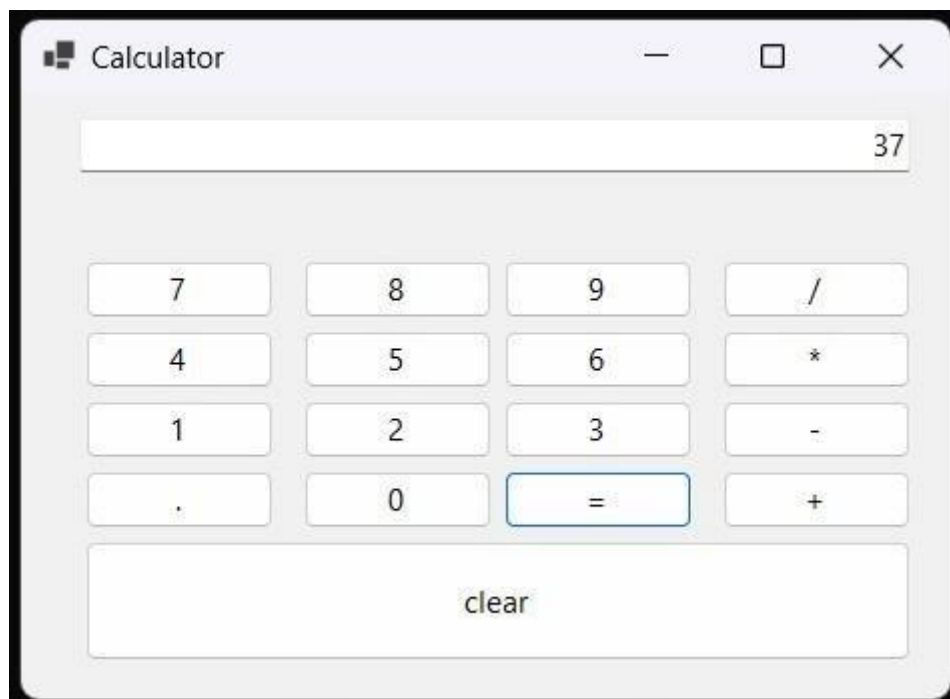
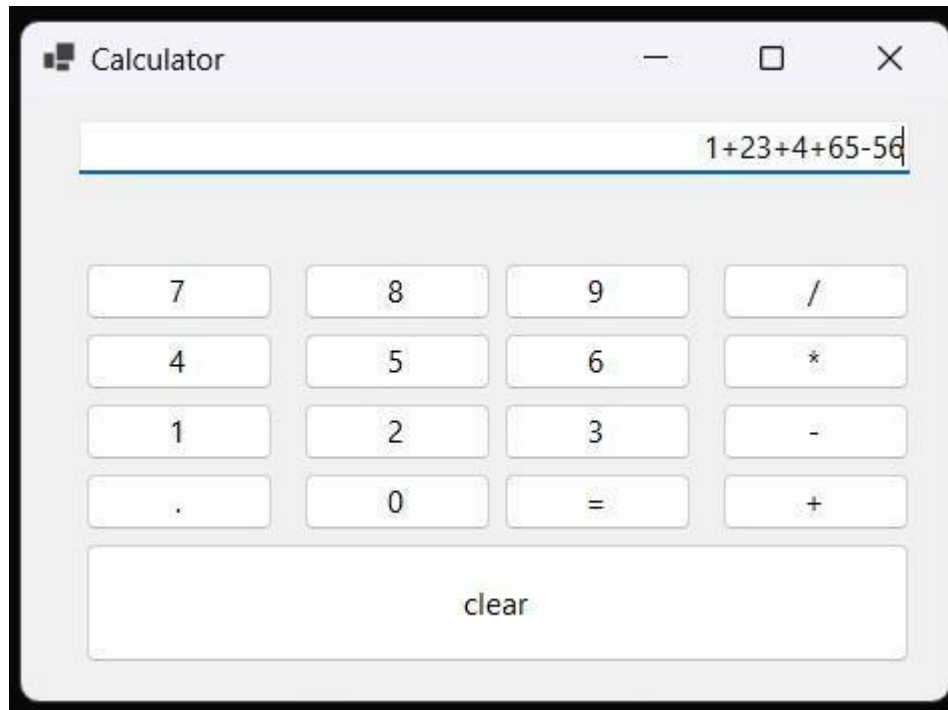
End Sub

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.

Ex.No	EMPLOYEE DETAILS	Date
10		

AIM:

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

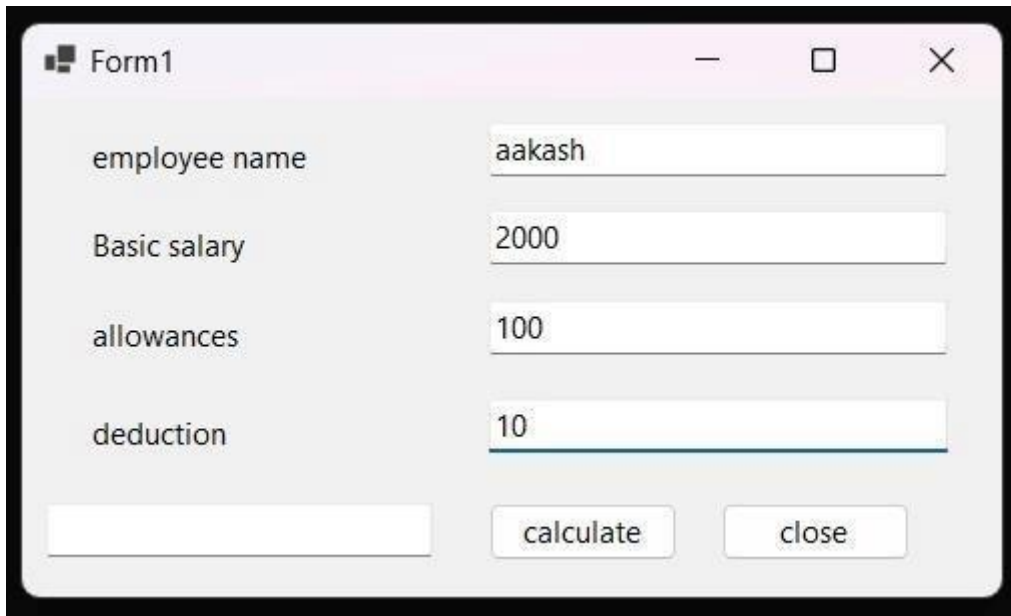
ALGORITHM:

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.

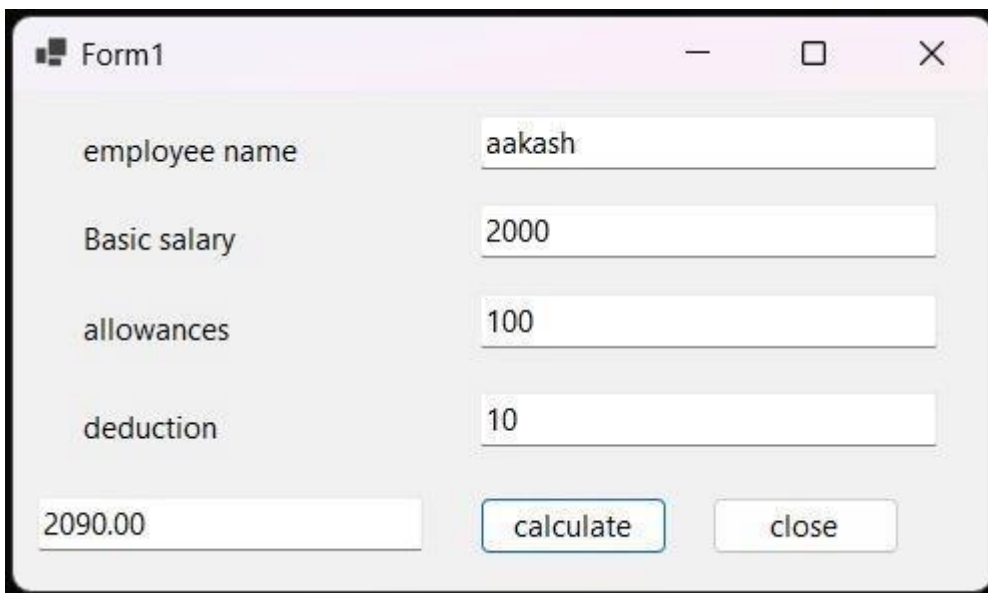
PROGRAM:

```
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:



A screenshot of a Windows application window titled "Form1". The window contains four input fields with labels: "employee name" (containing "aakash"), "Basic salary" (containing "2000"), "allowances" (containing "100"), and "deduction" (containing "10"). Below these fields is an empty input field, a "calculate" button, and a "close" button.



A screenshot of the same "Form1" window after calculation. The "employee name", "Basic salary", "allowances", and "deduction" fields remain the same. The previously empty input field now contains the value "2090.00". The "calculate" button is highlighted with a blue border, and the "close" button remains visible.

RESULT:

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

Ex.No	VOTERS	Date
11		

AIM

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

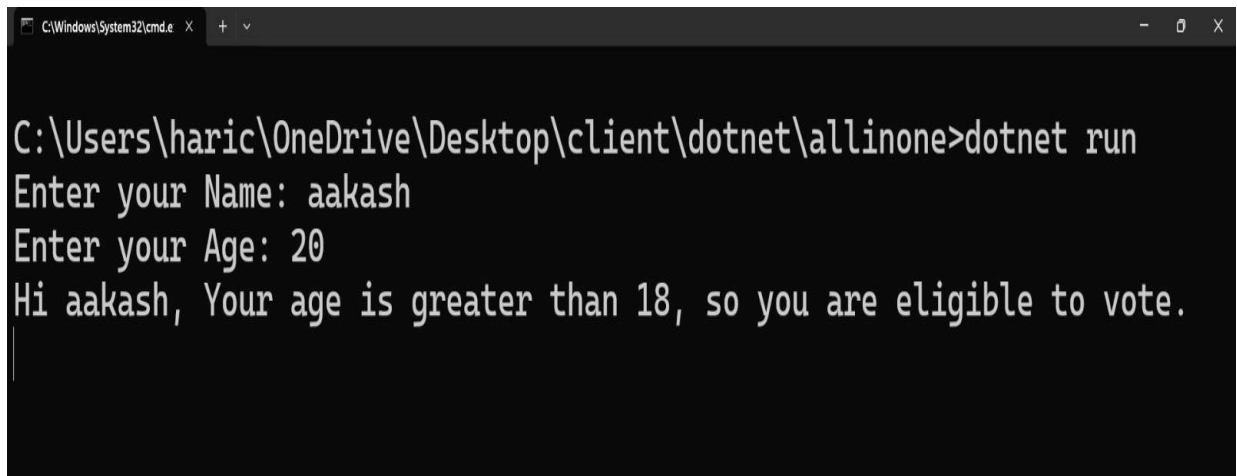
ALGORITHM

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.

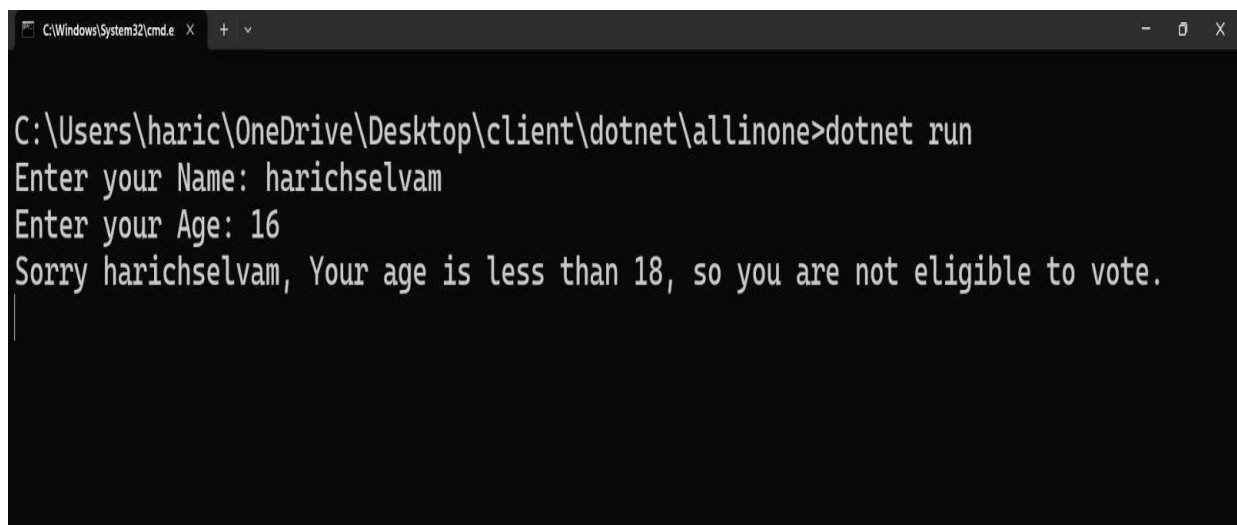
PROGRAM:

```
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:



```
C:\Windows\System32\cmd.exe X + v
C:\Users\haric\OneDrive\Desktop\client\dotnet\allinone>dotnet run
Enter your Name: aakash
Enter your Age: 20
Hi aakash, Your age is greater than 18, so you are eligible to vote.
```



```
C:\Windows\System32\cmd.exe X + v
C:\Users\haric\OneDrive\Desktop\client\dotnet\allinone>dotnet run
Enter your Name: harichselvam
Enter your Age: 16
Sorry harichselvam, Your age is less than 18, so you are not eligible to vote.
```

RESULT:

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

Ex.No	STUDENT RECORD	Date
12		

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:

```
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:

The 'Student Record' application window displays a table with 5 records. The input fields for Name, Regno, and Dept are empty. The table has columns: name, regno, dept.

name	regno	dept
Ayman	1015	DS(AI)
Abhi	1002	DS(AI)
Akil	1004	DS(AI)
Azam	1065	DS(AI)

DATA INSERT:

The 'Student Record' application window shows the 'Insert' button highlighted. The input fields contain: Name: Aakash, Regno: 1001, Dept: DS(AI). The table now has 6 records.

name	regno	dept
Aakash	1001	DS(AI)
Abhi	1002	DS(AI)
Akil	1004	DS(AI)
Ayman	1008	DS(AI)
Azam	1015	DS(AI)

A dialog box titled 'Data Inserted' with a red 'X' button in the top right corner. The text 'inserted' is displayed in the center. An 'OK' button is at the bottom.

DATA UPDATE:

Student Record

Name: Aakash
Regno: 1001
Dept: DS(AI)

Select Insert Delete Update Exit

name	regno	dept
Aakash	1001	DS(AI)
Abhi	1002	DS(AI)
Akil	1004	DS(AI)
Ayman	1008	DS(AI)
Azam	1015	DS(AI)

Record Updated

Updated

OK

DATA UPDATE:

Student Record

Name: Aakash
Regno: 1001
Dept: DS(AI)

Select Insert Delete Update Exit

name	regno	dept
Azam	1065	DS(AI)
Abhi	1002	DS(AI)
Akil	1004	DS(AI)
Ayman	1015	DS(AI)

Record Deleted

deleted

OK

RESULT:

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

Ex.No		Date
13	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:

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

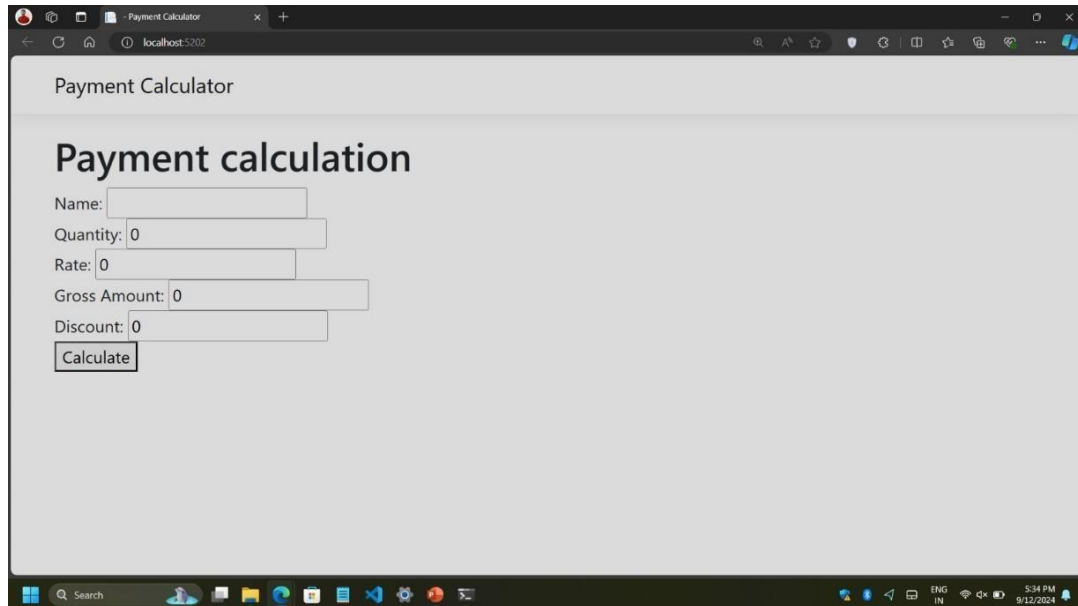
        a = Convert.ToInt32(TextBox2.Text);
        b = Convert.ToInt32(TextBox3.Text);
    }
}

```



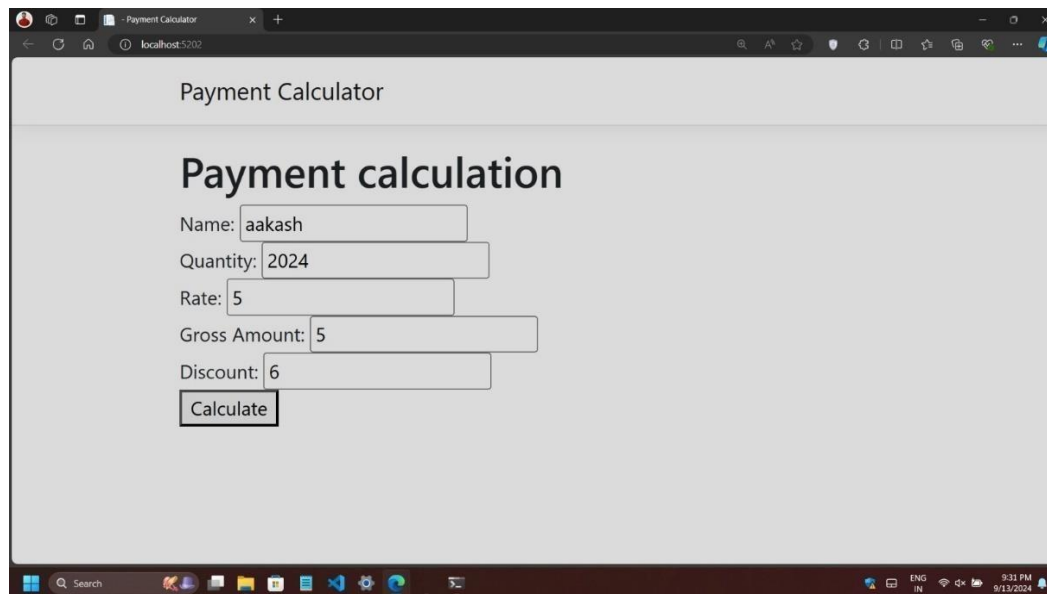
```
c = a * b;  
x = c * 10 / 100;  
y = c - x;  
TextBox4.Text = c.ToString();  
TextBox5.Text = x.ToString();  
TextBox6.Text = y.ToString();  
}  
}
```

OUTPUT:



The screenshot shows a web browser window titled "Payment Calculator" with the URL "localhost:5202". The page content includes a title "Payment calculation" and a form with the following fields and values:

Field	Value
Name:	
Quantity:	0
Rate:	0
Gross Amount:	0
Discount:	0
Calculate	Button



The screenshot shows the same web browser window with the "Payment Calculator" form now filled with sample data:

Field	Value
Name:	aakash
Quantity:	2024
Rate:	5
Gross Amount:	5
Discount:	6
Calculate	Button

RESULT:

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

Ex.No		Date
14	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:

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:

```

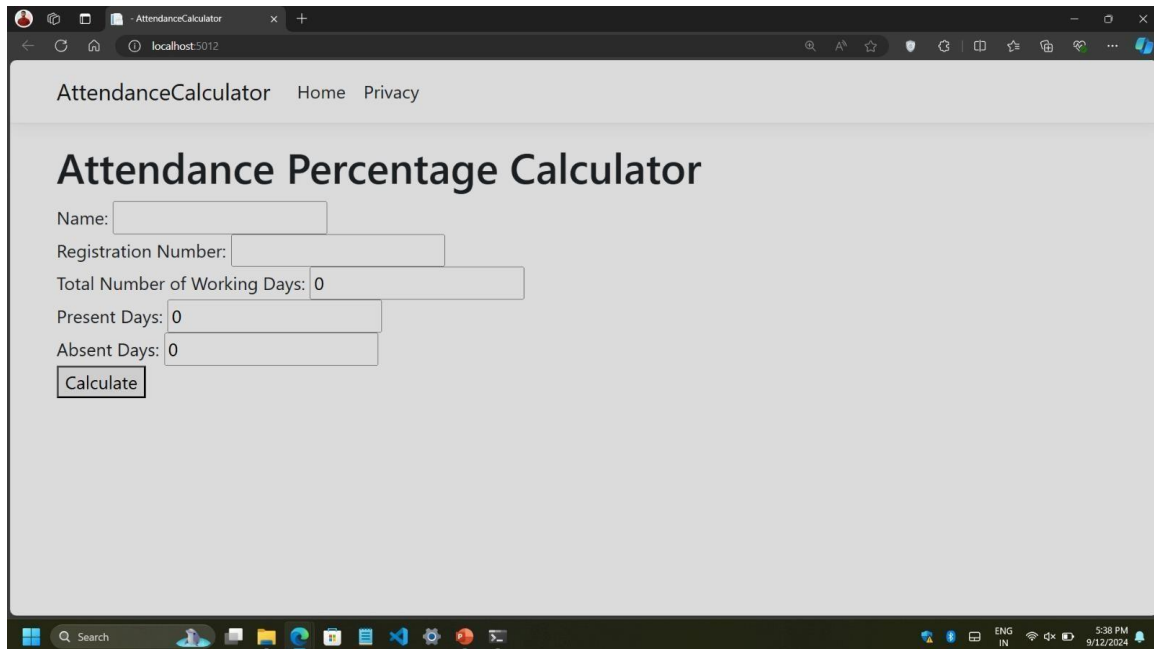
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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;
    }
}

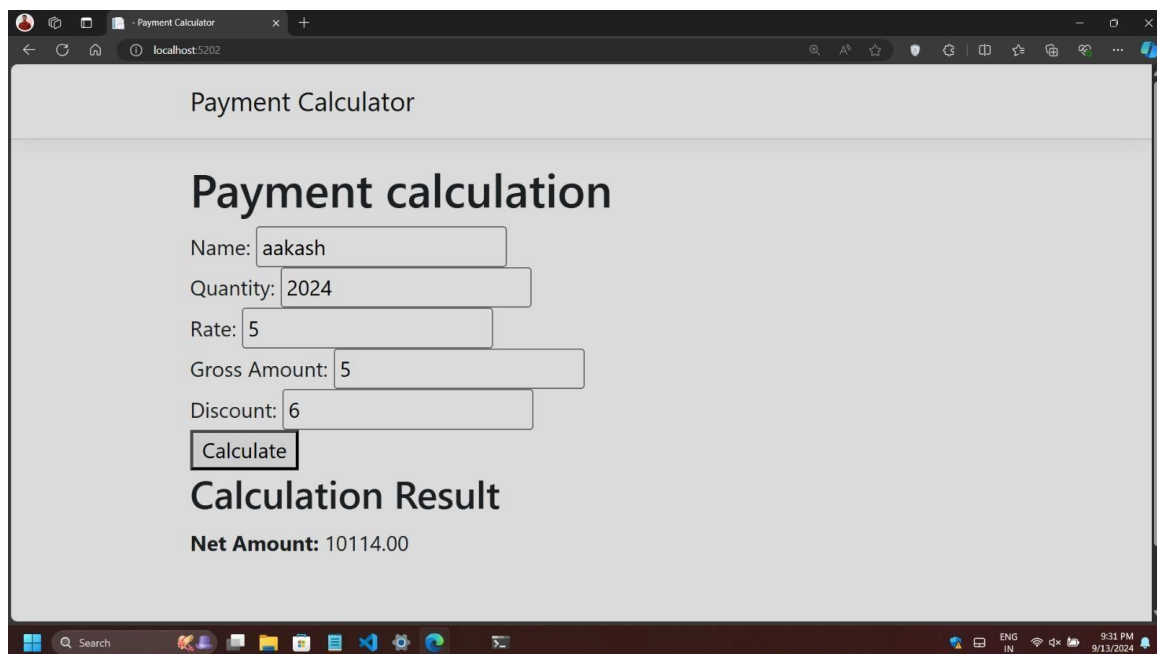
```

```
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:



The screenshot shows a web browser window with the title "AttendanceCalculator". The page has a navigation bar with "Home" and "Privacy" links. The main heading is "Attendance Percentage Calculator". Below the heading, there are input fields for "Name:", "Registration Number:", "Total Number of Working Days: 0", "Present Days: 0", and "Absent Days: 0". A "Calculate" button is located below the input fields. The browser's address bar shows "localhost:5012". The Windows taskbar at the bottom shows the time as 5:38 PM on 9/12/2024.



The screenshot shows a web browser window with the title "Payment Calculator". The page has a heading "Payment calculation". Below the heading, there are input fields for "Name: aakash", "Quantity: 2024", "Rate: 5", "Gross Amount: 5", and "Discount: 6". A "Calculate" button is located below the input fields. Below the button, the heading "Calculation Result" is displayed, followed by the text "Net Amount: 10114.00". The browser's address bar shows "localhost:5022". The Windows taskbar at the bottom shows the time as 9:31 PM on 9/13/2024.

RESULT:

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