Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY
University with Graded Autonomy Status
(An ISO 21001 : 2018 Certified Institution)
Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.

RECORD NOTEBOOK

BCS18I12 – DOT NET LAB

DEPARTEMENT

OF

COMPUTER SCIENCE AND ENGINEERING

NAME	**:**	VINAY RAM
REGISTER NO	**:**	211211101360
COURSE	**:**	B.TECH CSE-AI
YEAR/SEM/SEC	**:**	IV/VII/F

2024-2025 (ODD SEMESTER)

Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
DEEMED TO BE UNIVERSITY
University with Graded Autonomy Status
(An ISO 21001 : 2018 Certified Institution)
Periyar E.V.R. High Road, Maduravoyal, Chennai-95. Tamilnadu, India.

BONAFIDE CERTIFICATE

REGISTER NO : **21121110360**

NAME OF LAB**: DOT NET LAB - (BCS18I12)**

DEPARTMENT: **COMPUTER SCIENCE AND ENGINEERING-AI**

Certified that, this Record note book is a bonafide record of work done by **VINAY RAM** Of **IV** Year **B.Tech CSE-AI**, Sec-'**F**' in the **DOT NET LAB(BCS18I12)** 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

EXP NO	DATE	TITLE	PAGE N0	STAFF SIGNATURE
1		Fibonacci Series (C# .NET)		
2		Factorial (C# .NET)		
3		Complex Number (C# .NET)		
4		Matrix Addition (C# .NET)		
5		Student Status (C# .NET)		
6		Area of an Object (C# .NET)		
7		Enumerator (C# .NET)		
8		Structure (C# .NET)		
9		Calculator (VB .Net)		
10		Employee Details (VB .Net)		
11		Voters (VB .Net)		
12		Student Record (VB .Net)		
13		Payment Details (ASP.NET)		
14		Attendance Percentage (ADO.NET)		

Exp. No: 1

Date :

FIBONACCI SERIES

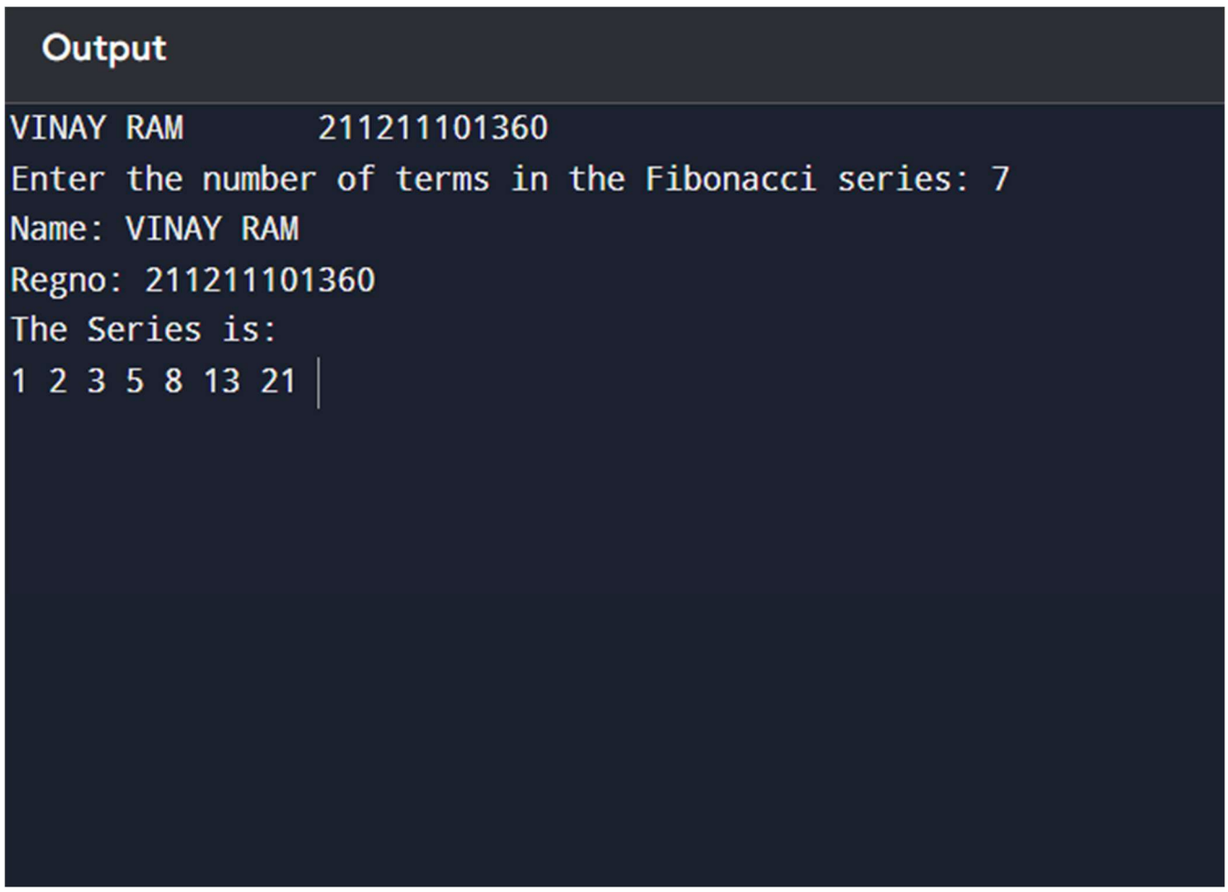
AIM:

ALGORITHM:

PROGRAM:(FIBONACCI SERIES)

```
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)
        {
            string name = "VINAY RAM";
            string regno = "211211101360";
            Console.WriteLine("VINAY RAM       211211101360”);

            int f1 = 0, f2 = 1, f3, i;

            Console.Write("Enter the number of terms in the Fibonacci series: ");
            int n = int.Parse(Console.ReadLine());

            Console.WriteLine("Name: " + name);

            Console.WriteLine("Regno: " + regno);
            Console.WriteLine("The Series is:");

            for (i = 0; i < n; i++)
            {
                f3 = f1 + f2;
                f1 = f2;
                f2 = f3;
                Console.Write(f3 + " ");
            }

            Console.ReadLine();
        }
    }
}
```

OUTPUT:



RESULT:

Exp.No: 2

Date:

FACTORIAL NUMBER

AIM:

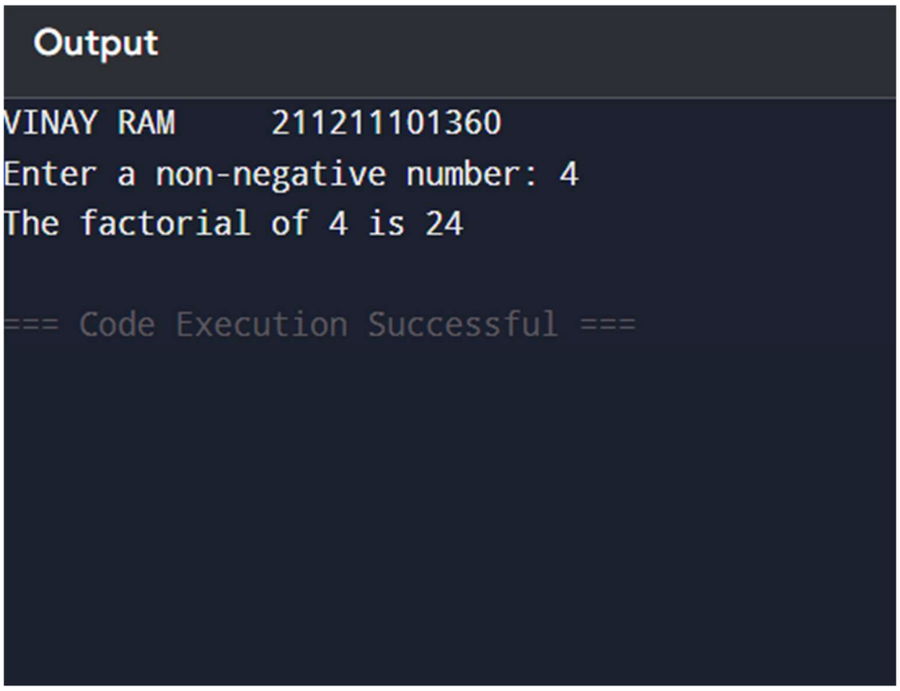
.

ALGORITHM:

PROGRAM:(FACTORIAL)

```
using System;

namespace Factorial
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("VINAY RAM     211211101360");

         int number;

         Console.Write("Enter a non-negative number: ");
         string input = Console.ReadLine();

         if (!int.TryParse(input, out number) || number < 0)
         {
            Console.WriteLine("Invalid input. Please enter a non-negative number.");
            return;
         }

         int factorial = CalculateFactorial(number);

         Console.WriteLine("The factorial of {0} is {1}", number, factorial);
      }

      static int CalculateFactorial(int number)
      {
         if (number == 0)
         {
            return 1; // Base case: factorial of 0 is 1
         }
         else
         {
            return number * CalculateFactorial(number - 1); // Recursive call
         }
      }
   }
}
```

OUTPUT:



RESULT:

Exp.No: 3

Date:

COMPLEX NUMBER

AIM:

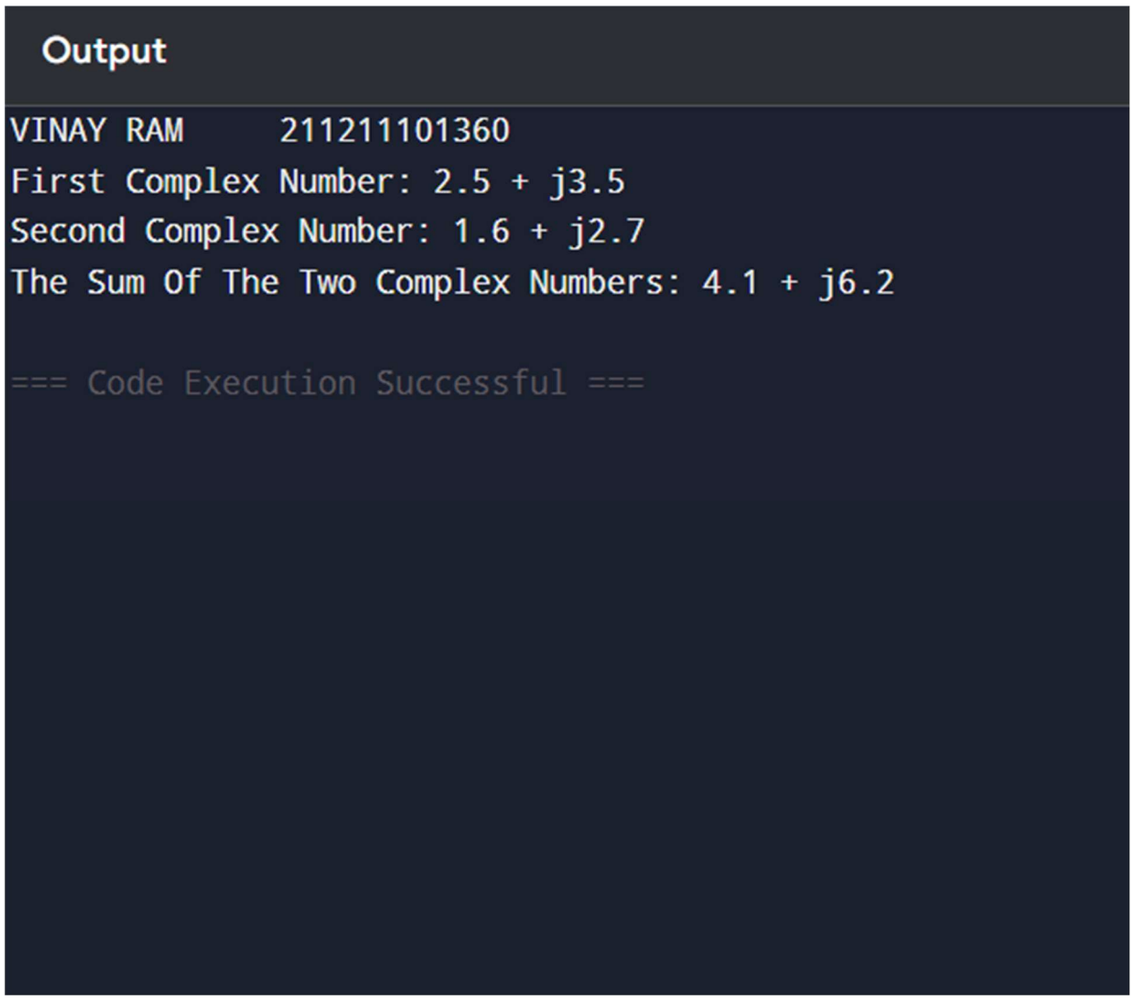
ALGORITHM:

PROGRAM:(COMPLEX NUMBER)

```
using System;

namespace ComplexNumbers
{
   public class ComplexNumber
   {
      private double realPart;
      private double imaginaryPart;

      public ComplexNumber(double real, double imaginary)
      {
         realPart = real;
         imaginaryPart = imaginary;
      }

      // Properties to access the real and imaginary parts
      public double Real
      {
         get { return realPart; }
         set { realPart = value; }
      }

      public double Imaginary
      {
         get { return imaginaryPart; }
         set { imaginaryPart = value; }
      }

      // Overloading the addition operator (+)
      public static ComplexNumber operator +(ComplexNumber c1, ComplexNumber c2)
      {
         return new ComplexNumber(c1.Real + c2.Real, c1.Imaginary + c2.Imaginary);
      }

      // Overriding the ToString() method for custom string representation
      public override string ToString()
      {
         if (imaginaryPart >= 0)
         {
            return $"{realPart} + j{imaginaryPart}";
         }
         else
         {
            return $"{realPart} - j{Math.Abs(imaginaryPart)}";
         }
      }
```

```
    }

    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("VINAY RAM     211211101360");

            ComplexNumber c1 = new ComplexNumber(2.5, 3.5);
            ComplexNumber c2 = new ComplexNumber(1.6, 2.7);

            ComplexNumber sum = c1 + c2;

            Console.WriteLine("First Complex Number: " + c1);
            Console.WriteLine("Second Complex Number: " + c2);
            Console.WriteLine("The Sum Of The Two Complex Numbers: " + sum);
        }
    }
}
```

OUTPUT:



RESULT:

Exp.No: 4

Date:

MATRIX ADDITION

AIM:

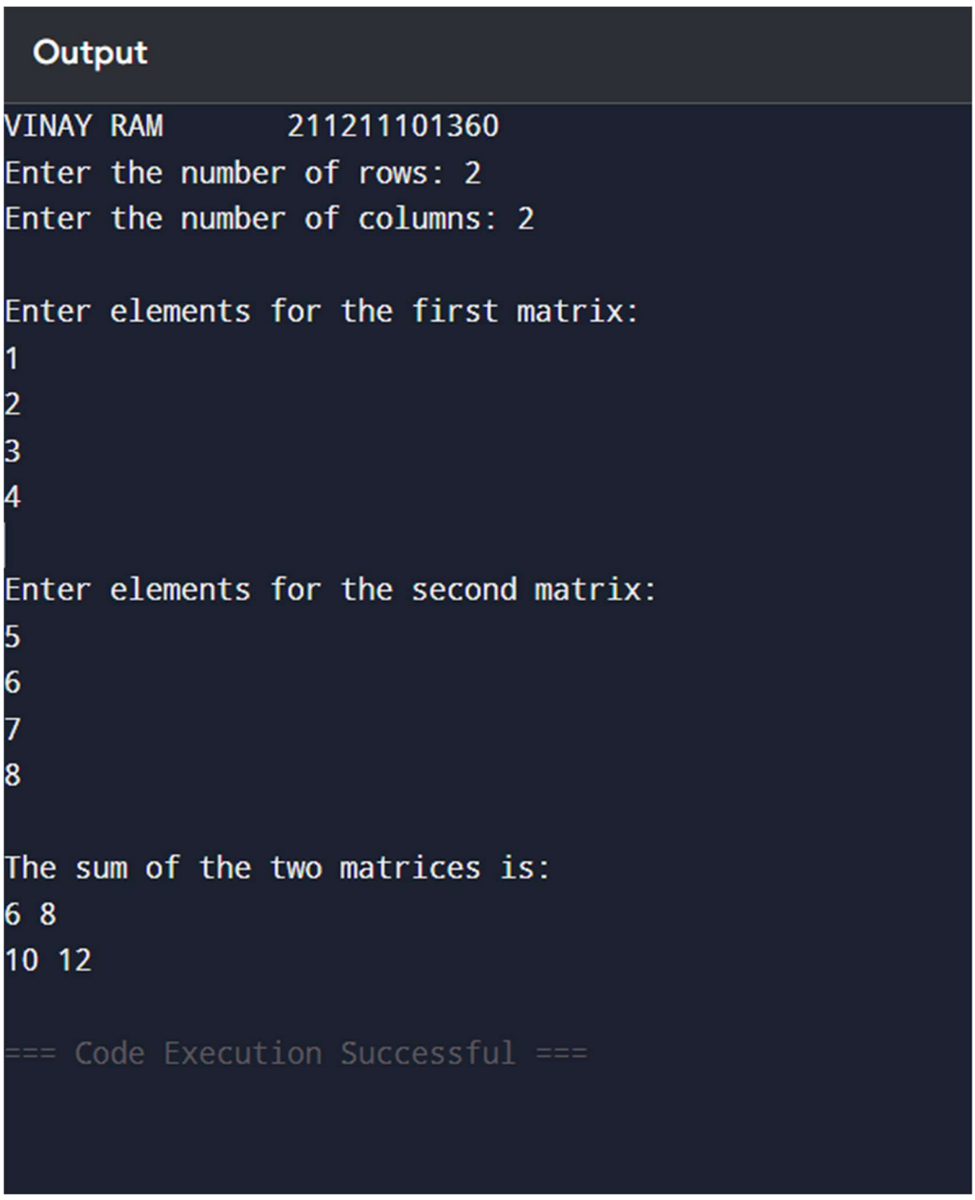
ALGORITHM:

PROGRAM: (MATRIX ADDITION)

```
using System;

namespace MatrixAddition
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("VINAY RAM       211211101360");

         int rows, cols;

         Console.Write("Enter the number of rows: ");
         rows = int.Parse(Console.ReadLine());

         Console.Write("Enter the number of columns: ");
         cols = int.Parse(Console.ReadLine());

         int[,] matrix1 = new int[rows, cols];
         int[,] matrix2 = new int[rows, cols];
         int[,] sumMatrix = new int[rows, cols];

         Console.WriteLine("\nEnter elements for the first matrix:");
         for (int i = 0; i < rows; i++)
         {
            for (int j = 0; j < cols; j++)
            {
               matrix1[i, j] = int.Parse(Console.ReadLine());
            }
         }

         Console.WriteLine("\nEnter elements for the second matrix:");
         for (int i = 0; i < rows; i++)
         {
            for (int j = 0; j < cols; j++)
            {
               matrix2[i, j] = int.Parse(Console.ReadLine());
            }
         }

         // Add the matrices
         for (int i = 0; i < rows; i++)
         {
            for (int j = 0; j < cols; j++)
            {
               sumMatrix[i, j] = matrix1[i, j] + matrix2[i, j];
```

```
            }
        }

        Console.WriteLine("\nThe sum of the two matrices is:");
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                Console.Write(sumMatrix[i, j] + " ");
            }
            Console.WriteLine();
        }
    }
  }
}
```

OUTPUT:



RESULT:

Exp.No: 5 Date:

Student Status (Multiple Inheritance)

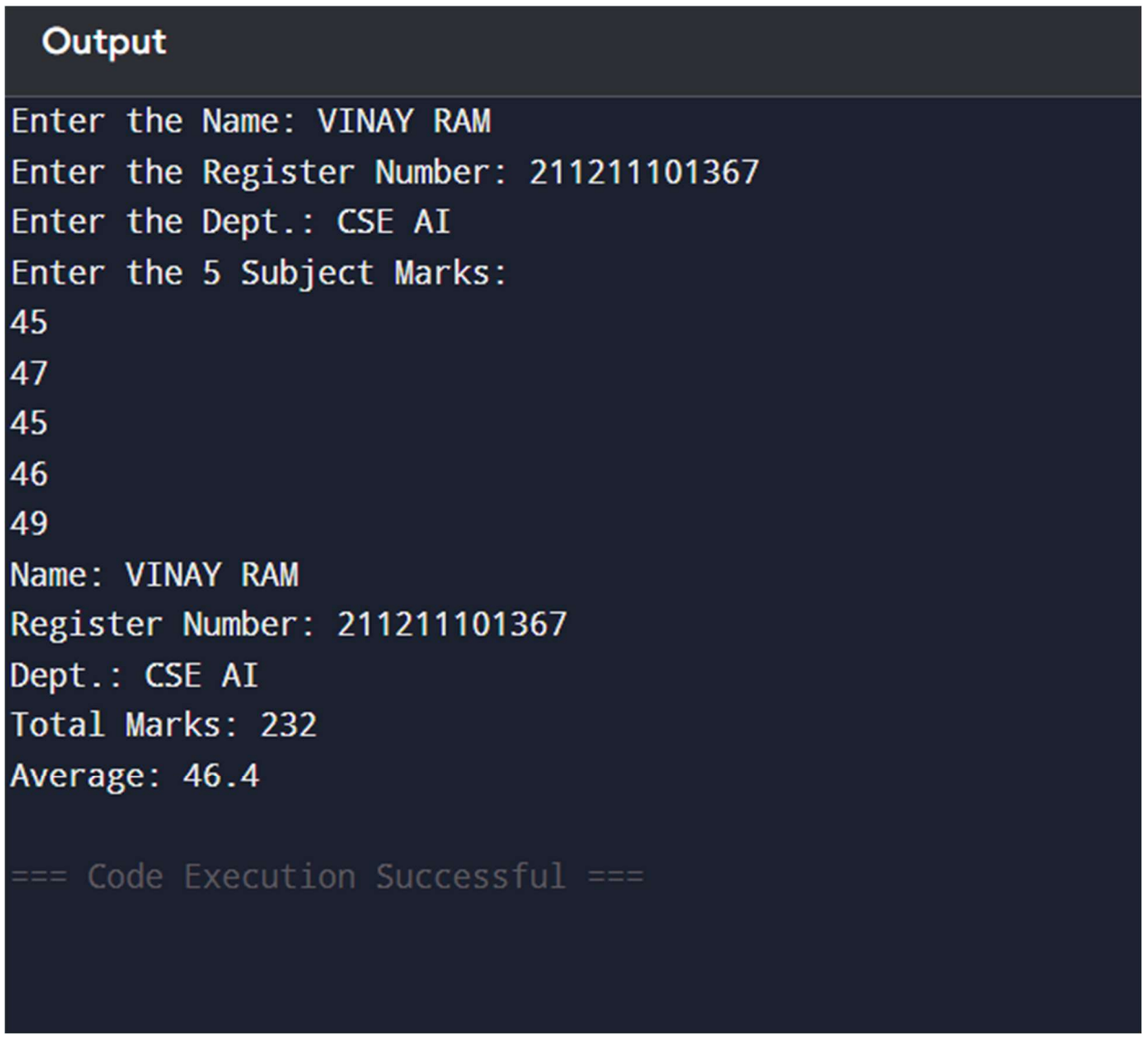
Aim:

Algorithm:

PROGRAM

```
using System;

namespace StudentInformation
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.Write("Enter the Name: ");
         string name = Console.ReadLine();

         Console.Write("Enter the Register Number: ");
         string regno = Console.ReadLine();

         Console.Write("Enter the Dept.: ");
         string dept = Console.ReadLine();

         Console.WriteLine("Enter the 5 Subject Marks:");

         int[] marks = new int[5];
         for (int i = 0; i < 5; i++)
         {
            marks[i] = int.Parse(Console.ReadLine());
         }

         int totalMarks = 0;
         for (int i = 0; i < 5; i++)
         {
            totalMarks += marks[i];
         }

         double average = (double)totalMarks / 5;

         Console.WriteLine("Name: " + name);
         Console.WriteLine("Register Number: " + regno);
         Console.WriteLine("Dept.: " + dept);
         Console.WriteLine("Total Marks: " + totalMarks);
         Console.WriteLine("Average: " + average);
      }
   }
```

OUTPUT:



RESULT:

Exp.No: 6

Date:

AREA OF AN OBJECT (MULTIPLE INHERITANCE)

AIM:

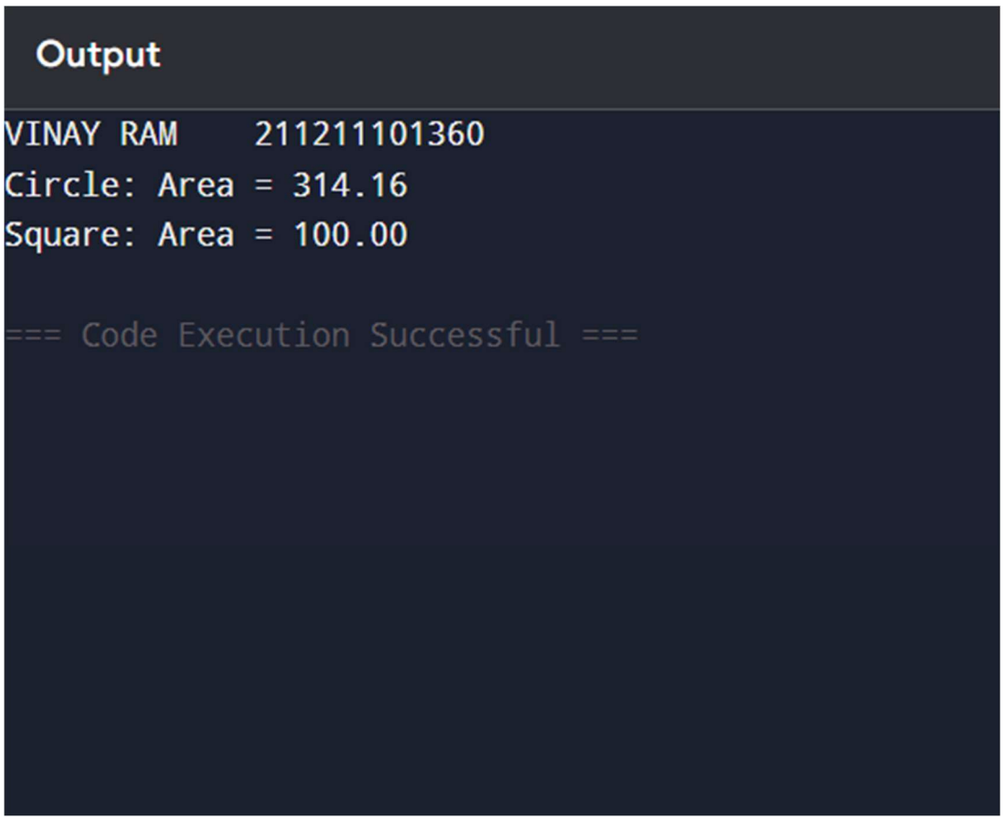
ALGORITHM:

PROGRAM: (AREA OF AN OBJECT)

```
using System;

class Shape
{
    public virtual double Area()
    {
        return 0;
    }
}

class Circle : Shape
{
    private double radius;

    public Circle(double radius)
    {
        this.radius = radius;
    }

    public override double Area()
    {
        return Math.PI * radius * radius;
    }
}

class Square : Shape
{
    private double side;

    public Square(double side)
    {
        this.side = side;
    }

    public override double Area()
    {
        return side * side;
    }
}

class ShapeCalculator
{
    public static void CalculateArea(Shape shape)
    {
```

```
        double area = shape.Area();
        Console.WriteLine($"{shape.GetType().Name}: Area = {area:F2}");
    }
}

class Program
{
    static void Main(string[] args)
    {

        Console.WriteLine("VINAY RAM    211211101360");

        Circle circle = new Circle(10);
        Square square = new Square(10);

        ShapeCalculator.CalculateArea(circle);
        ShapeCalculator.CalculateArea(square);
    }
}
```

OUTPUT:



RESULT:

Exp.No: 7

Date:

ENUMERATOR

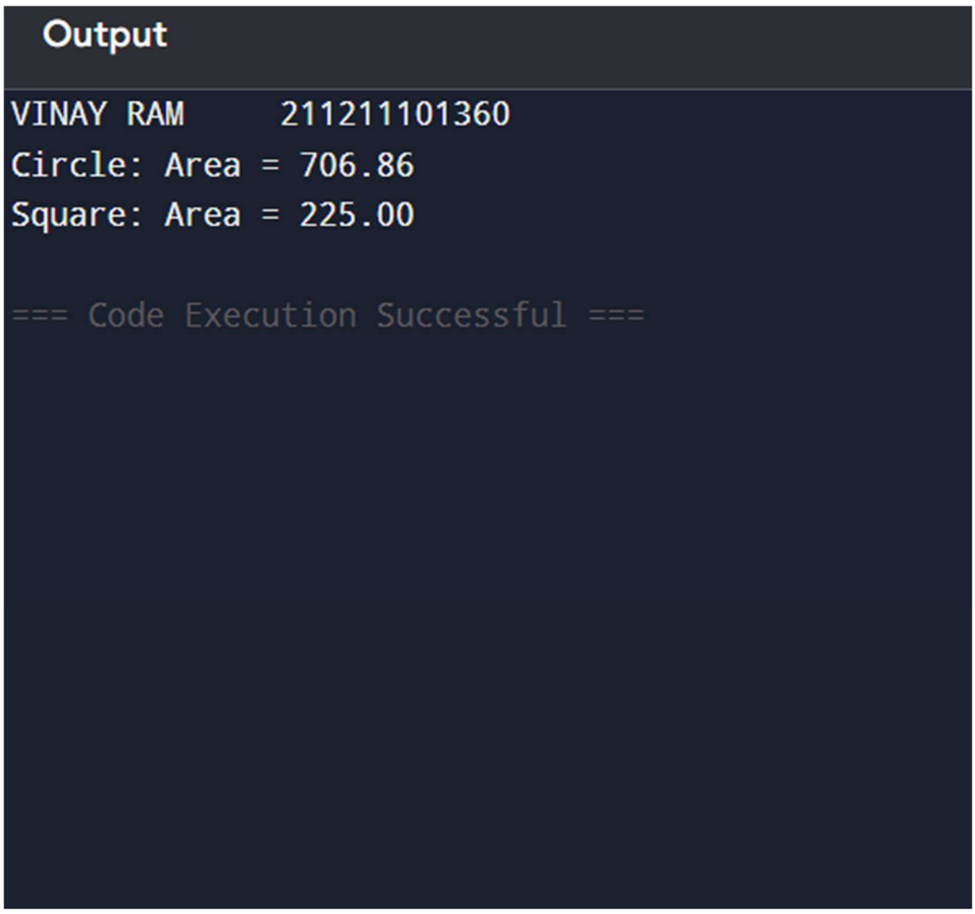
AIM:

ALGORITHM:

PROGRAM: (ENUMERATOR)

```
using System;
enum ShapeType { Circle, Square }
class Shape
{
   public ShapeType Type { get; set; }
   public double Radius { get; set; }
   public double Side { get; set; }

   public double CalculateArea()
   {
      switch (Type)
      {
         case ShapeType.Circle:
            return Math.PI * Radius * Radius;
         case ShapeType.Square:
            return Side * Side;
         default:
            throw new ArgumentException("Invalid ShapeType");
      }
   }
}

class Program
{
   static void Main(string[] args)
   {
      Console.WriteLine("VINAY RAM     211211101360");

      Shape circle = new Shape { Type = ShapeType.Circle, Radius = 15 };
      Shape square = new Shape { Type = ShapeType.Square, Side = 15 };

      Console.WriteLine($"{circle.Type}: Area = {circle.CalculateArea():F2}");
      Console.WriteLine($"{square.Type}: Area = {square.CalculateArea():F2}");
   }
}
```

OUTPUT:



RESULT:

Exp.No: 8

Date:

STRUCTURE

AIM:

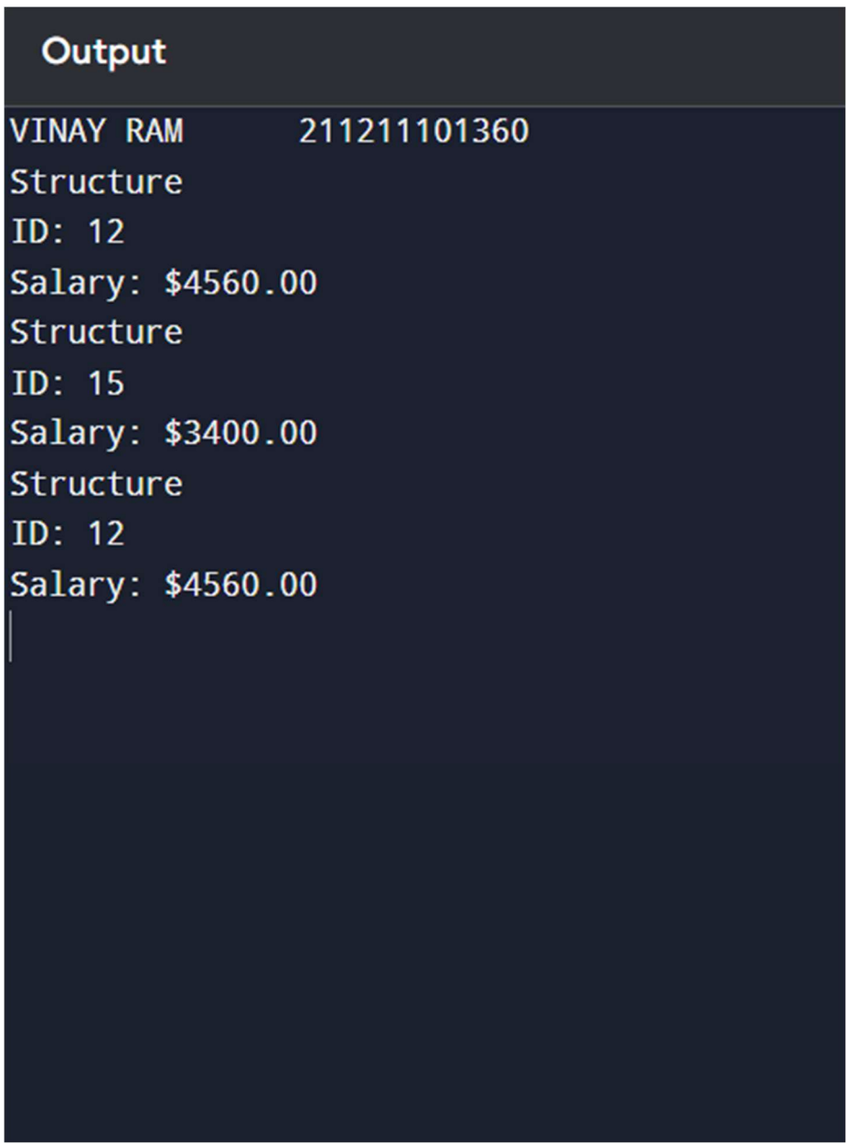
ALGORITHM:

PROGRAM: (STRUCTURE)

```
using System;

namespace MyStructures
{
   public struct Employee
   {
      public int Id { get; set; }
      public double Salary { get; set; }

      public Employee(int id, double salary)
      {
         Id = id;
         Salary = salary;
      }

      public Employee(int id) : this(id, 3400.00)
      {
      }

      public Employee(Employee other)
      {
         Id = other.Id;
         Salary = other.Salary;
      }

      public void DisplayValues()
      {
         Console.WriteLine("Structure");
         Console.WriteLine($"ID: {Id}");
         Console.WriteLine($"Salary: ${Salary:F2}");
      }
   }

   class Program
   {
      static void Main(string[] args)
      {

         Console.WriteLine("VINAY RAM      211211101360”);
                  Employee emp1 = new Employee(12, 4560.00);
         Employee emp2 = new Employee(15);
         Employee emp3 = new Employee(emp1);
```

```
            emp1.DisplayValues();
            emp2.DisplayValues();
            emp3.DisplayValues();

            Console.ReadLine();
        }
    }
}
```

OUTPUT:



RESULT:

Exp.No: 9

Date:

DESIGN A CALCULATOR

AIM:

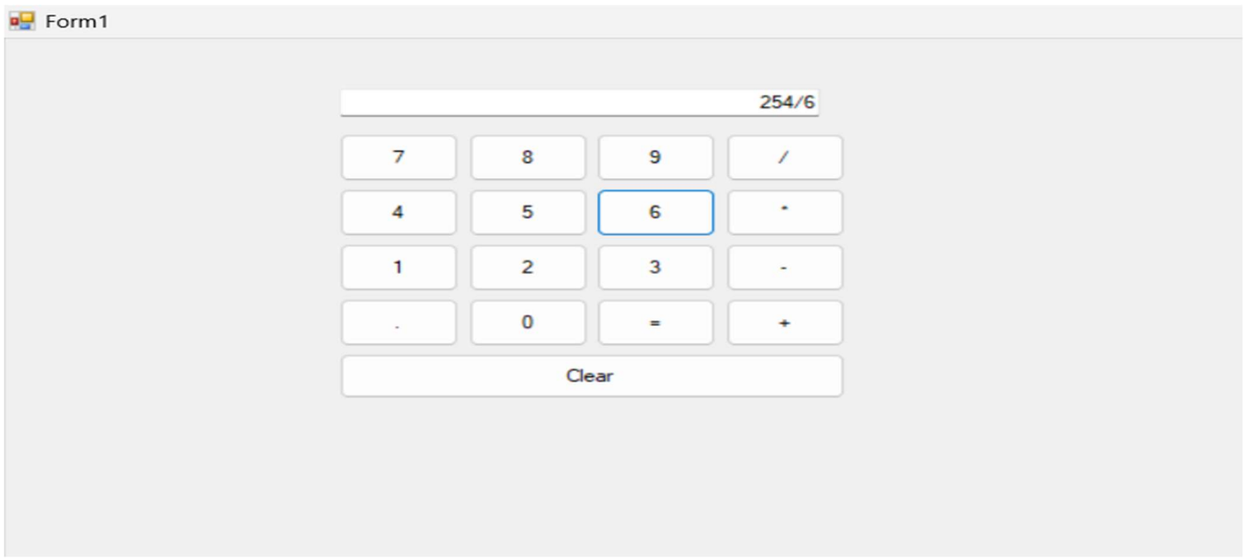
ALGORITHM:

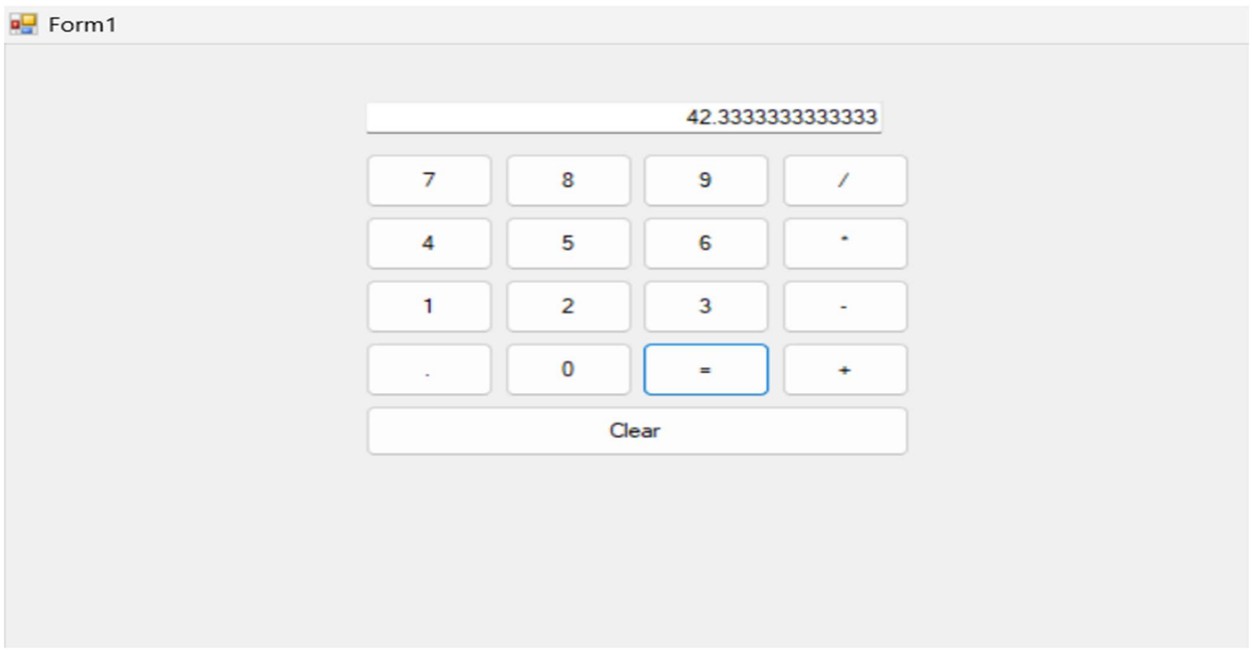
PROGRAM: (CALCULATOR)

```
Imports System.Net.Security

Public Class Form1
    Private Sub buttonClick(sender As Object, e As EventArgs) Handles btn1.Click, btn2.Click,
btn3.Click, btn4.Click, btn5.Click, btn6.Click, btn7.Click, btn8.Click, btn9.Click, btn0.Click,
btnDot.Click, btnMinus.Click, btnPlus.Click, btnEqual.Click, btnDiv.Click, btnMul.Click,
btnClear.Click
        Dim button As Button = CType(sender, Button)
        If button.Name = "btn1" Then
            TextBox1.Text = TextBox1.Text + "7"
        End If
        If button.Name = "btn2" Then
            TextBox1.Text = TextBox1.Text + "8"
        End If
        If button.Name = "btn3" Then
            TextBox1.Text = TextBox1.Text + "9"
        End If
        If button.Name = "btn4" Then
            TextBox1.Text = TextBox1.Text + "4"
        End If
        If button.Name = "btn5" Then
            TextBox1.Text = TextBox1.Text + "5"
        End If
        If button.Name = "btn6" Then
            TextBox1.Text = TextBox1.Text + "6"
        End If
        If button.Name = "btn7" Then
            TextBox1.Text = TextBox1.Text + "1"
        End If
        If button.Name = "btn8" Then
            TextBox1.Text = TextBox1.Text + "2"
        End If
        If button.Name = "btn9" Then
            TextBox1.Text = TextBox1.Text + "3"
        End If
        If button.Name = "btn0" Then
            TextBox1.Text = TextBox1.Text + "0"
        End If
        If button.Name = "btnDiv" Then
            TextBox1.Text = TextBox1.Text + "/"
        End If
        If button.Name = "btnMul" Then
            TextBox1.Text = TextBox1.Text + "*"
        End If
        If button.Name = "btnDot" Then
            TextBox1.Text = TextBox1.Text + "."
```

```
        End If
        If button.Name = "btnMinus" Then
            TextBox1.Text = TextBox1.Text + "-"
        End If
        If button.Name = "btnPlus" Then
            TextBox1.Text = TextBox1.Text + "+"
        End If
        If button.Name = "btnClear" Then
            TextBox1.Text = ""
        End If
        If button.Name = "btnEqual" Then
            Dim str As String = TextBox1.Text
            Dim result = New DataTable().Compute(str, Nothing)
            TextBox1.Text = result
        End If

    End Sub
End Class
```

OUTPUT:





RESULT:

Exp.No: 10 Date:

EMPLOYEE DETAILS

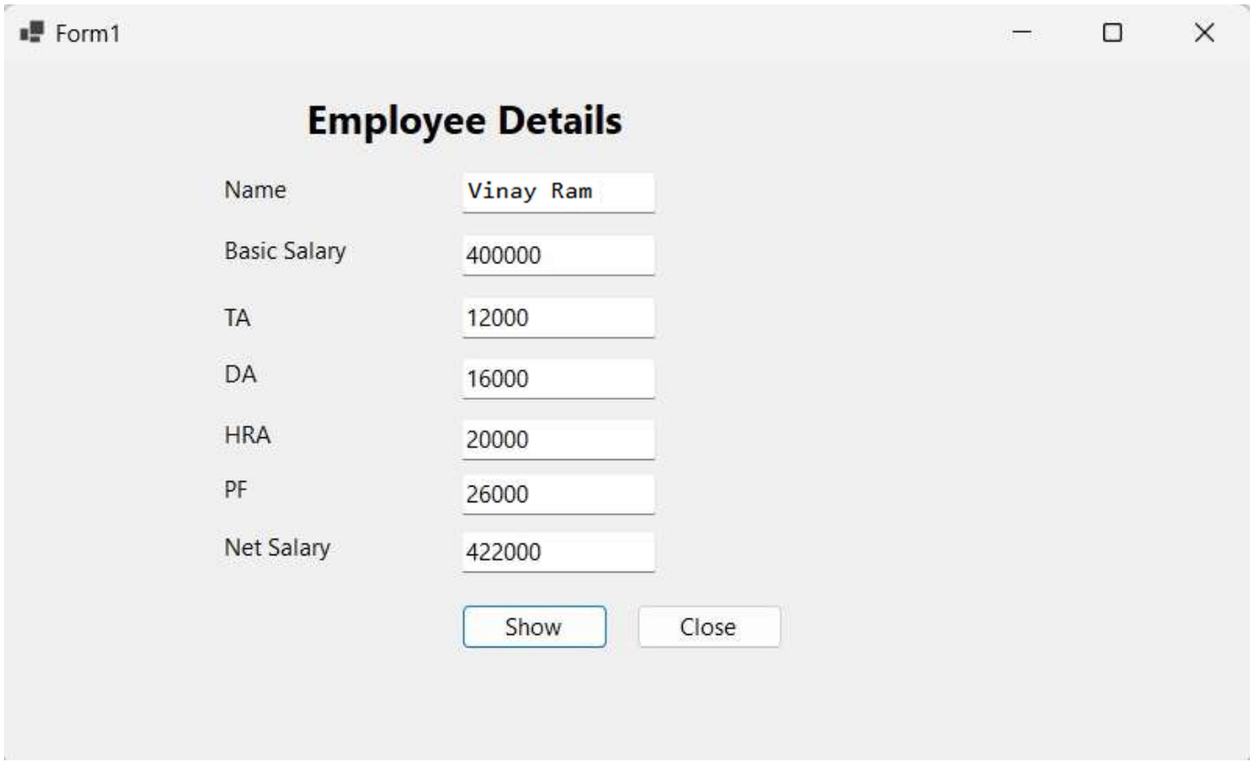
AIM:

ALGORITHM:

PROGRAM: (EMPLOYEE DETAILS)

```
Public Class Form1
    Dim nm As String
    Dim s As Integer
    Dim ta, da, hra, pf, nt As Single

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

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        nm = TextBox1.Text
        s = TextBox2.Text
        ta = s * 3 / 100
        TextBox3.Text = ta
        da = s * 4 / 100
        TextBox4.Text = da
        hra = s * 5 / 100
        TextBox5.Text = hra
        pf = s * 6.5 / 100
        TextBox6.Text = pf
        nt = s + ta + da + hra - pf
        TextBox7.Text = nt
    End Sub
End Class
```

OUTPUT:



RESULT:

Exp.No: 11

Date:

VOTERS
(EXCEPTION HANDLING)

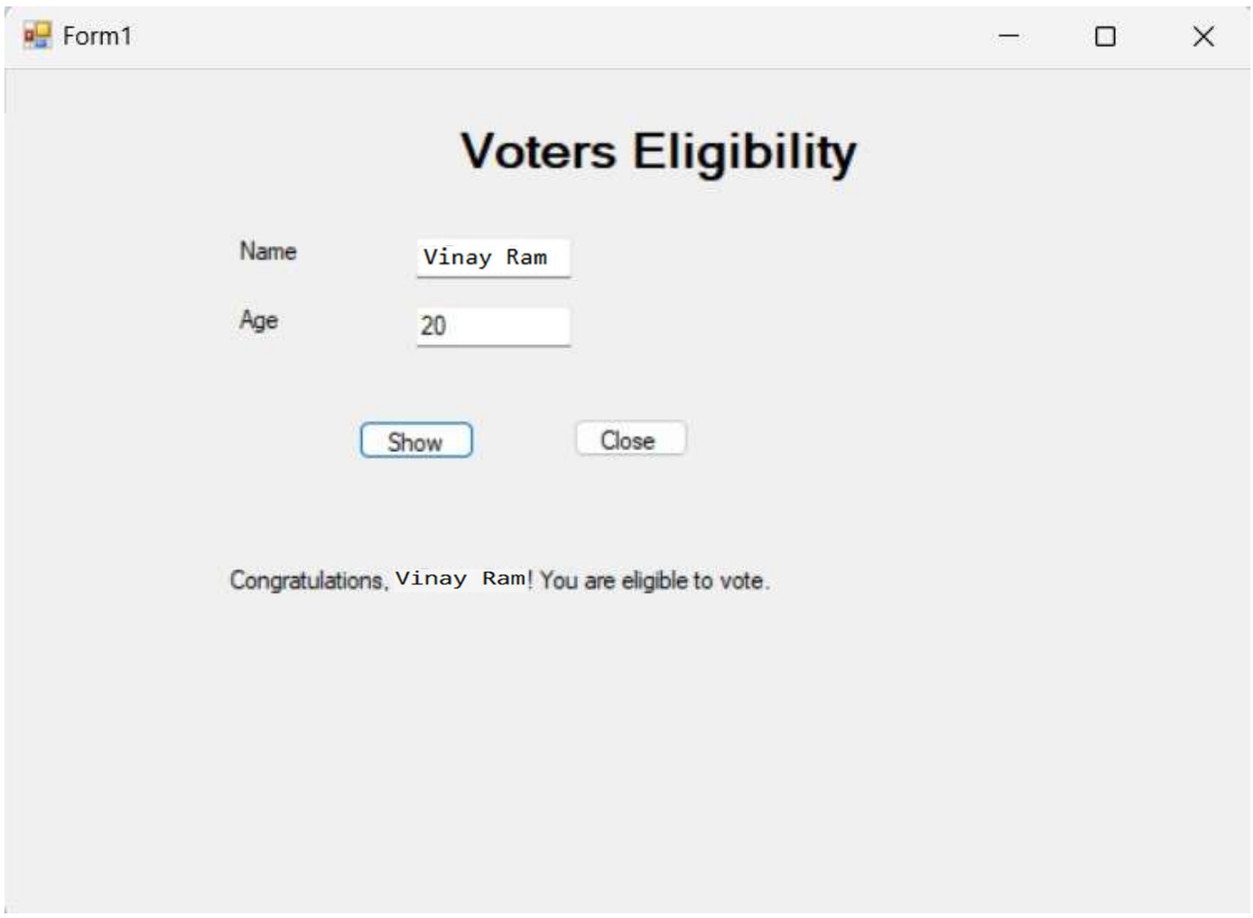
AIM:

ALGORITHM:

PROGRAM: (VOTERS)

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Try
            ' Get user input
            Dim name As String = TextBox1.Text.Trim()  ' Trim leading/trailing spaces
            Dim age As Integer = Integer.Parse(TextBox2.Text)

            ' Validate name (optional)
            If name = "" Then
                Throw New Exception("Please enter your name.")
            End If

            ' Validate age
            If age < 18 Then
                Throw New Exception("You are not eligible to vote. Minimum age is 18.")
            End If

            ' Display eligibility message
            Label4.Text = String.Format("Congratulations, {0}! You are eligible to vote.", name)
        Catch ex As Exception
            ' Handle exceptions (including parsing errors)
            Label4.Text = "An error occurred: " + ex.Message
        End Try
    End Sub
End Class
```

OUTPUT:



RESULT:

Exp.No: 12

Date:

ADO.NET

AIM:

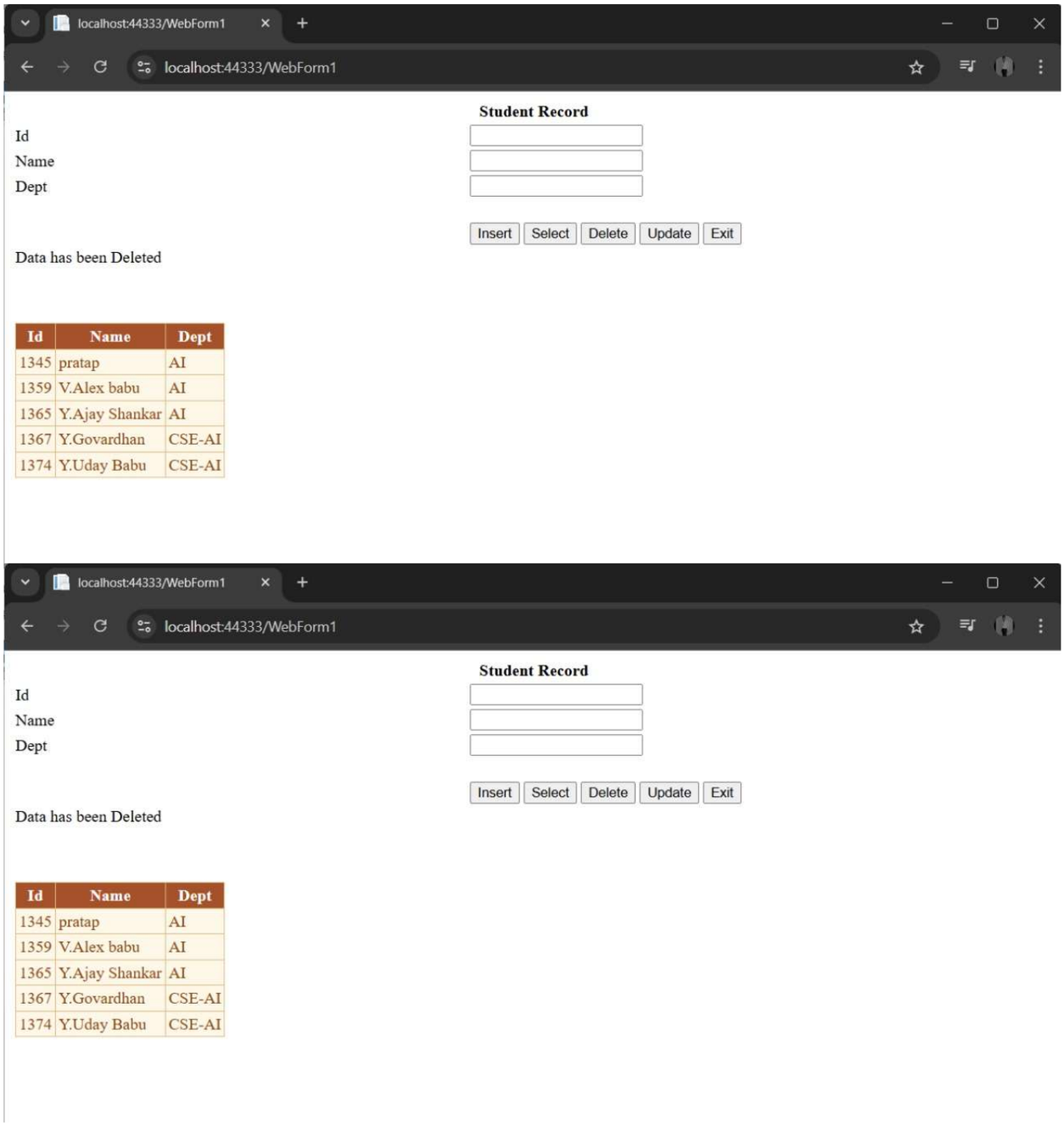
ALGORITHM:

PROGRAM: (STUDENT RECORD)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;

namespace WebApplication17
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);
      protected void Page_Load(object sender, EventArgs e)
      {
         con.Open();
      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         SqlCommand cmd = new SqlCommand("insert into utbl values('" + TextBox1.Text + "','" +
TextBox2.Text + "','" + TextBox3.Text + "')", con);
         cmd.ExecuteNonQuery();
         con.Close();
         Label1.Text = "Data has been inserted";
         GridView1.DataBind();
         TextBox1.Text = "";
         TextBox2.Text = "";
         TextBox3.Text = "";
      }

      protected void Button2_Click(object sender, EventArgs e)
      {
         SqlCommand cmd = new SqlCommand("update utbl set name='" + TextBox2.Text + "', age='"
+ TextBox3.Text + "' where Id='" + TextBox1.Text + "'", con);
         cmd.ExecuteNonQuery();
         con.Close();
         Label1.Text = "Data has been Updated";
         GridView1.DataBind();
         TextBox1.Text = "";
         TextBox2.Text = "";
         TextBox3.Text = "";
      }
```

```
        protected void Button3_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("delete from utbl where Id='" +
    Convert.ToInt32(TextBox1.Text).ToString() + "'", con);
            cmd.ExecuteNonQuery();
            con.Close();
            Label1.Text = "Data has been Deleted";
            GridView1.DataBind();
            TextBox1.Text = "";
            TextBox2.Text = "";
            TextBox3.Text = "";
        }

        protected void Button4_Click(object sender, EventArgs e)
        {
            string find = "select * from utbl where (Id like '%' +@Id+ '%')";
            SqlCommand cmd = new SqlCommand(find, con);
            cmd.Parameters.Add("@Id", SqlDbType.NVarChar).Value = TextBox4.Text;
            cmd.ExecuteNonQuery();

            SqlDataAdapter da = new SqlDataAdapter();
            da.SelectCommand = cmd;
            DataSet ds = new DataSet();
            da.Fill(ds, "Id");

            GridView1.DataSourceID = null;
            GridView1.DataSource = ds;
            GridView1.DataBind();

            con.Close();
            Label1.Text = "data has been selected";

        }
    }
}
```

OUTPUT:



RESULT:

Exp.No: 13 Date:

ASP.NET
(PAYMENT DETAILS)

AIM:

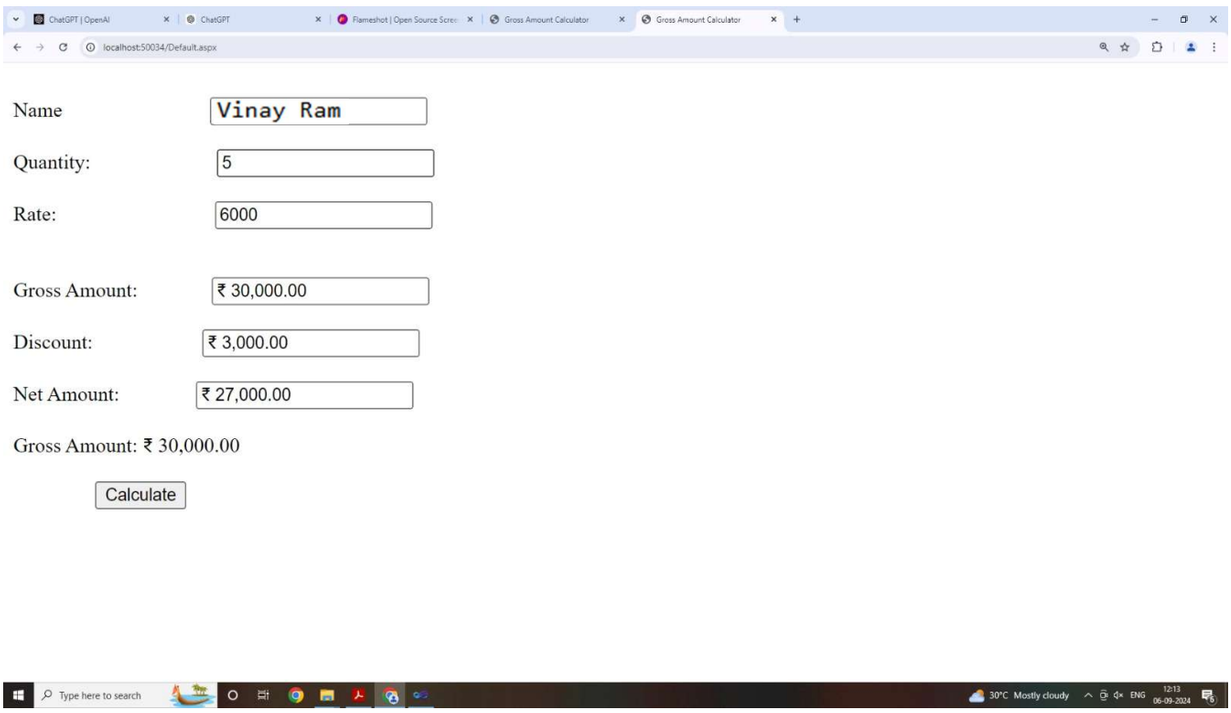
ALGORITHM:

1. .

PROGRAM: PAYMENT DETAILS (ASP.NET)

```
using System;
using System.Web.UI;

public partial class Form2
{
   protected void Button1_Click(object sender, EventArgs e)
   {
      double quantity, rate, discount, grossAmount, netAmount;

      try
      {
         quantity = double.Parse(TextBox2.Text);
         rate = double.Parse(TextBox3.Text);
         discount = double.Parse(TextBox5.Text);

         grossAmount = quantity * rate;
         netAmount = grossAmount - discount;

         Label7.Text = grossAmount.ToString("0.00");
         Lable6.Text = netAmount.ToString("0.00");
      }
      catch (Exception ex)
      {
         Label7.Text = "Error";
         Label6.Text = "Error";
         Console.WriteLine(ex.Message);
      }
   }
```

OUTPUT:



RESULT:

Exp.No: 14 Date:

ASP.NET
(ATTENDANCE PERCENTAGE)

AIM:

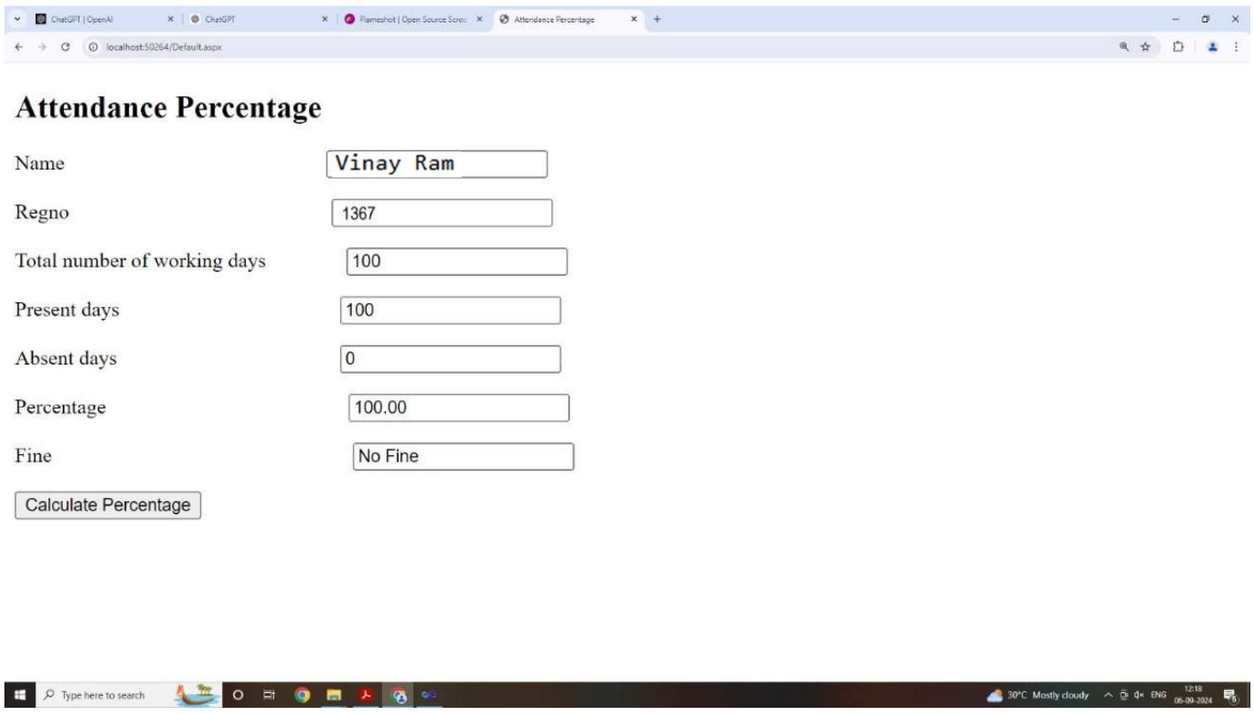
ALGORITHM:

PROGRAM: (ATTENDANCE PERCENTAGE)

```
using System;
using System.Web.UI;

public partial class Form1
{
   protected void Button1_check(object sender, EventArgs e)
   {
      double totalDays, presentDays, percentage, fine = 0;
      string examEligibility = "";

      try
      {
         totalDays = double.Parse(TextBox3.Text);
         presentDays = double.Parse(TextBox4.Text);

         if (totalDays <= 0)
         {
            TextBox6.Text = "Error: Total days cannot be zero.";
         }
         else
         {
            percentage = (presentDays / totalDays) * 100;
            TextBox6.Text = percentage.ToString("0.00");

            if (!string.IsNullOrEmpty(TextBox7.Text))
            {
               fine = double.Parse(TextBox7.Text);
            }

            if (percentage >= 70)
            {
               fine = 0;
               examEligibility = "Eligible";
            }
            else if (percentage >= 60)
            {
               fine = 1000 ;
               examEligibility = "Eligible with Fine";
            }
            else if (percentage >= 50)
            {
               fine = 2000 ;
               examEligibility = "Eligible with Fine";
            }
            else if (percentage >= 40)
            {
```

```
            fine = 3000 ;
            examEligibility = "Eligible with Fine";
        }
        else
        {
            fine = 0;
            examEligibility = "Not Eligible";
        }

        TextBox7.Text = fine.ToString("0.00");
        TextBox7.Text = examEligibility;
    }
}
catch (Exception ex)
{
    TextBox6.Text = "Error: Invalid input.";
    TextBox7.Text = "N/A";
```

OUTPUT:



RESULT: