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

BCS18L12 – DOT NET LAB

2024–2025 (ODD SEMESTER)

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

NAME	**:**	SUMIT KUMAR
REGISTER NO	**:**	211061101448
COURSE	**:**	B.TECH (CSE)
YEAR/SEM/SEC	**:**	IV / VII / H

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: 211061101448

Name of Lab: DOT NET LAB

Department: COMPUTER SCIENCE AND ENGINEERING

Certified that, this Record note book is a **bonafide** record of work done by **SUMIT KUMAR** of **IV Year B. Tech / CSE, Sec - 'H'** in the **DOT NET LAB** 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	Name of Experiment	Page No	Staff Signature
1		Implementation of Operator Overloading a) Complex Number b) Matrix c) Time	1-8	
2		Implementation of Multiple Inheritance a) Employee b) Area of an Object	9-12	
3		Implementing Multi-threading	13-14	
4		Exception Handling	15-16	
5		Designing a Calculator	17-20	
6		Implement File Handling	21-23	
7		Implement Exception Handling a) Employee Detail b) Voter c) Student Status	24-29	
8		Super Market	30-33	
9		Hotel Management System	34-37	
10		Student Attendance System	38-41	
11		Hospital Management System	42-45	
12		Income tax Calculation	46-48	

Exp No: 1 (a) **Date:**

COMPLEX NUMBER

AIM

To write a C# program to calculate complex number.

ALGORITHM

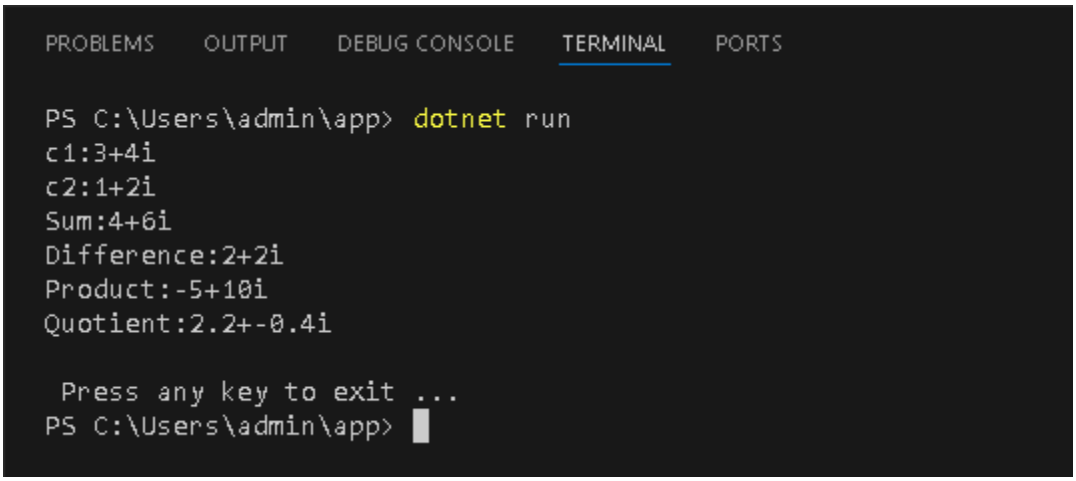
1. Start the program.
2. Declare the class complex.
3. Write a function to overload +operator.
4. Declare two objects of the complex class.
5. Assign real & imaginary value to both objects.
6. Add both objects using +operator.
7. Display the value.
8. Stop the program.

PROGRAM

```
using System;
namespace ComplexNumberExample {
public class Complex {
public double Real {
get; set;
}
public double Imaginary {
get; set;
}
// Constructor
public Complex(double real, double imaginary) {
Real = real;
Imaginary = imaginary;
}
// Overload the + operator
public static Complex operator +(Complex c1, Complex c2) {
return new Complex(c1.Real + c2.Real, c1.Imaginary + c2.Imaginary);
}
// Overload the – operator
public static Complex operator -(Complex c1, Complex c2){
return new Complex(c1.Real - c2.Real, c1.Imaginary - c2.Imaginary);
}
// Overload the * operator
public static Complex operator *(Complex c1, Complex c2){
```

```
double real = c1.Real * c2.Real - c1.Imaginary * c2.Imaginary;
double imaginary = c1.Real * c2.Imaginary + c1.Imaginary * c2.Real;
return new Complex(real, imaginary);
}
// Overload the / operator
public static Complex operator /(Complex c1, Complex c2) {
double denominator = c2.Real * c2.Real + c2.Imaginary * c2.Imaginary;
double real = (c1.Real * c2.Real + c1.Imaginary * c2.Imaginary) / denominator;
double imaginary = (c1.Imaginary * c2.Real - c1.Real * c2.Imaginary) / denominator;
return new Complex(real, imaginary);
}
// Override the ToStringmetho.
public override string ToString() {
return  + Real +"+" +Imaginary + "i";
}
}

class Program {
static void Main(string[] args) {
Complex c1 = new Complex(3, 4);
Complex c2 = new Complex(1, 2);
Complex sum = c1 + c2;
Complex difference = c1 - c2;
Complex product = c1 * c2;
Complex quotient = c1 / c2;
Console.WriteLine("c1:"+c1);
Console.WriteLine("c2:"+c2);
Console.WriteLine("Sum:" +sum);
Console.WriteLine("Difference:"+difference);
Console.WriteLine("Product:"+product);
Console.WriteLine("Quotient:"+quotient);
Console.WriteLine("\n Press any key to exit ...");
Console.ReadKey();
}
}
}
```

OUTPUT



RESULT

Thus, a C# program to calculate complex number has been successfully executed and the output is verified successfully

Exp No: 1 (b) **Date:**

MATRIX OPERATIONS

AIM

To write a C# program to create a matrix.

ALGORITHM

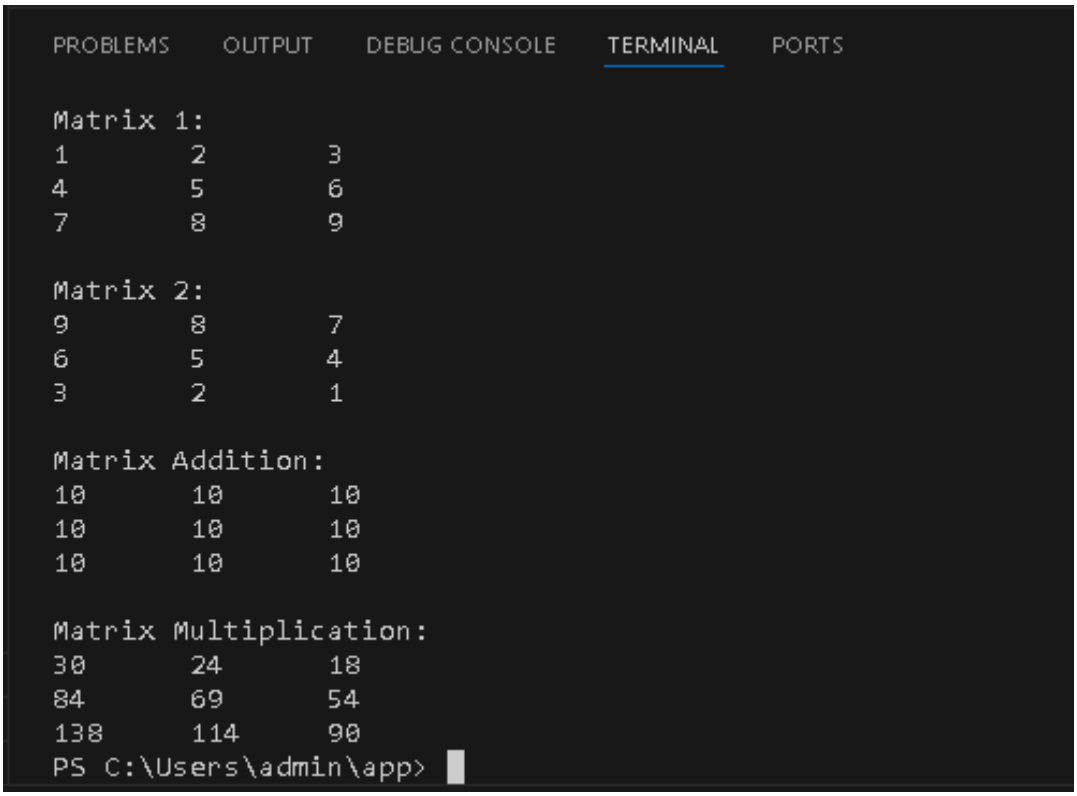
1. Start the program.
2. Create a class matrix.
3. Write a function to overload + & * operator.
4. Create two object of class matrix.
5. Take the matrix value that is randomly generated.
6. Add the two and multiple the matrix using + and * operator respectively.
7. Display the matrix.
8. Stop the program.

PROGRAM

```
using System;
class MatrixOperations {
static void Main() {
// Create two matrices
int[,] matrix1 = {
{ 1, 2, 3 },
{ 4, 5, 6 },
{ 7, 8, 9 }
};
int[,] matrix2 = {
{ 9, 8, 7 },
{ 6, 5, 4 },
{ 3, 2, 1 }
};
Console.WriteLine("Matrix 1:");
PrintMatrix(matrix1);
Console.WriteLine("\nMatrix 2:");
PrintMatrix(matrix2);
Console.WriteLine("\nMatrix Addition:");
int[,] resultAddition = AddMatrices(matrix1, matrix2);
PrintMatrix(resultAddition);
Console.WriteLine("\nMatrix Multiplication:");
int[,] resultMultiplication=MultiplyMatrices(matrix1,matrix2);
PrintMatrix(resultMultiplication);
```

```
}
// Function to print a matrix
static void PrintMatrix(int[,] matrix) {
int rows=matrix.GetLength(0);
int cols=matrix.GetLength(1);
for(int i=0;i<rows;i++) {
for(int j=0;j<cols;j++) {
Console.Write(matrix[i, j] + "\t");
}
Console.WriteLine();
}
}
// Function to add two matrices
static int[,] AddMatrices(int[,] matrix1, int[,] matrix2) {
int rows = matrix1.GetLength(0);
int cols = matrix1.GetLength(1);
int[,] result = new int[rows, cols];
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
result[i, j] = matrix1[i, j] + matrix2[i, j];
}
}
return result;
}
// Function to multiply two matrices
static int[,] MultiplyMatrices(int[,] matrix1, int[,] matrix2) {
int rows1 = matrix1.GetLength(0);
int cols1 = matrix1.GetLength(1);
int cols2 = matrix2.GetLength(1);
int[,] result = new int[rows1, cols2];
for (int i = 0; i < rows1; i++) {
for (int j = 0; j < cols2; j++) {
result[i, j] = 0;
for (int k = 0; k < cols1; k++) {
result[i, j] += matrix1[i, k] * matrix2[k, j];
}
}
}
return result;
}
}
```

OUTPUT



RESULT

Thus, a C# program to create a matrix has been executed and the output is verified successfully.

Exp No: 1 (c) **Date:**

TIME MANIPULATOIN

AIM

To write a C# program to manipulate the time.

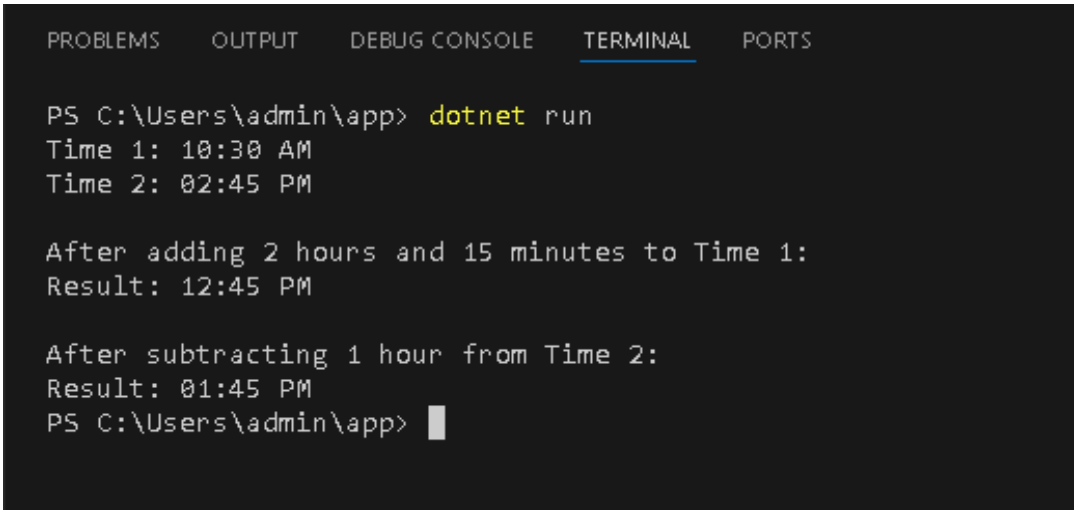
ALGORITHM

1. Start the program.
2. Create class and methods.
3. Create a string of array and store the methods.
4. Use predefined statement for getting current date and time.
5. Print the date and time.
6. Stop the program.

PROGRAM

```
using System;
class Program {
static void Main() {

// Creating two DateTime objects for demonstration
DateTime time1 = new DateTime(2024, 6, 26, 10, 30, 0); // 10:30 AM
DateTime time2 = new DateTime(2024, 6, 26, 14, 45, 0); // 2:45 PM

// Displaying the initial times
Console.WriteLine("Time 1: " + time1.ToString("hh:mm tt"));
Console.WriteLine("Time 2: " + time2.ToString("hh:mm tt"));

// Adding and subtracting time
DateTime addedTime = time1.AddHours(2).AddMinutes(15);
DateTime subtractedTime = time2.Subtract(TimeSpan.FromHours(1));

// Displaying the results
Console.WriteLine("\nAfter adding 2 hours and 15 minutes to Time 1:");
Console.WriteLine("Result: " + addedTime.ToString("hh:mm tt"));
Console.WriteLine("\nAfter subtracting 1 hour from Time 2:");
Console.WriteLine("Result: " + subtractedTime.ToString("hh:mm tt"));
}
}
```

OUTPUT



RESULT

Thus, a C# program to manipulate the time has been executed and the output is verified successfully.

Exp No: 2 (a) **Date:**

EMPLOYEE DETAILS - MULTIPLE INHERITANCE

AIM

To write a C# program to display employee details using multiple inheritance.

ALGORITHM

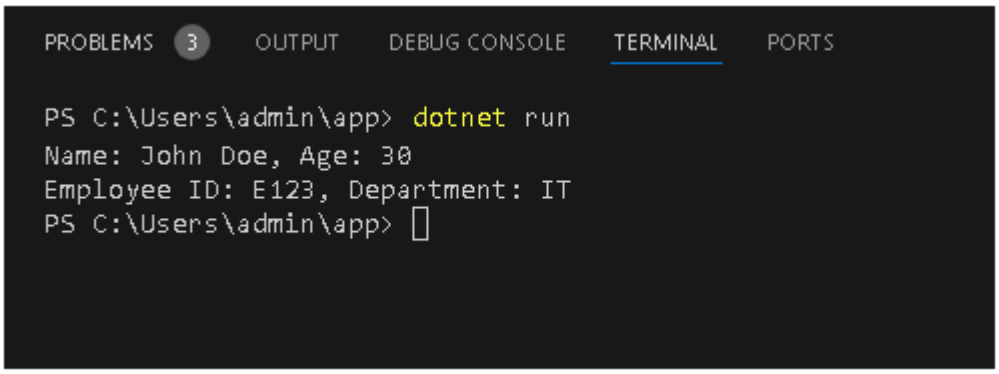
1. Start the program.
2. Create a class.
3. Create a variable and store user input values in it.
4. Display those values in other class using “Extend” keyword.
5. Create an object for first class and print the variables.
6. Stop the program.

PROGRAM

```
using System;
namespace MultipleInheritanceExample {
interface IEmployee {
void DisplayEmployeeDetails();
} interface IPerson {
void DisplayPersonDetails();
}
class EmployeeDetails : IEmployee, IPerson {
public string EmployeeId { get; set; }
public string Department { get; set; }
public string Name { get; set; }
public int Age { get; set; }
public void DisplayEmployeeDetails() {
Console.WriteLine($"Employee ID: {EmployeeId}, Department: {Department}");
}
public void DisplayPersonDetails() {
Console.WriteLine($"Name: {Name}, Age: {Age}");
}
}
class Program{
static void Main(string[] args){
EmployeeDetails employeeDetails = new EmployeeDetails{
EmployeeId = "E123",          Department = "IT",
Name = "John Doe",             Age = 30
};
employeeDetails.DisplayPersonDetails();
```

```
employeeDetails.DisplayEmployeeDetails();
}
}
}
```

OUTPUT



RESULT

Therefore, a C# program to display employee details using multiple inheritance has been executed and the output is verified successfully.

Exp No: 2 (b) **Date:**

AREA OF AN OBJECT - MULTIPLE INHERITANCE

AIM

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

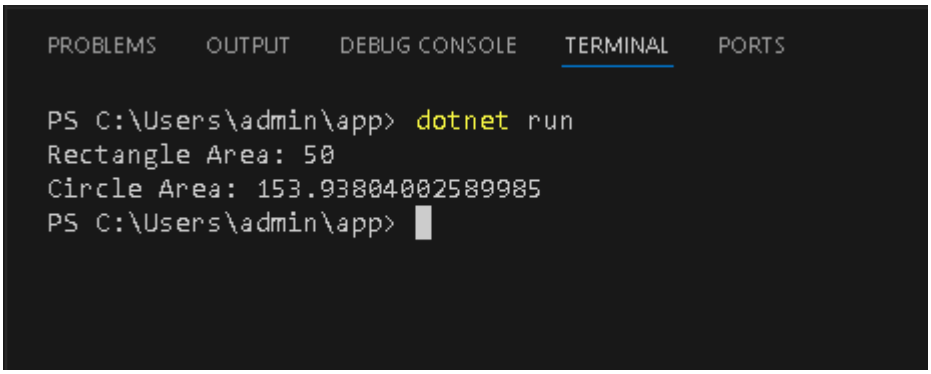
ALGORITHM

1. Start the program.
2. Create an interface area and declare a method compute.
3. Define two class Rectangle 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;
namespace MultipleInheritanceExample {
interface IArea {
double CalculateArea();
}
class Rectangle : IArea {
public double Width { get; set; }
public double Height { get; set; }
public double CalculateArea() => Width * Height;
}
class Circle : IArea {
public double Radius { get; set; }
public double CalculateArea() => Math.PI * Radius * Radius;
}
class Program {
static void Main(string[] args) {
IArea rectangle = new Rectangle { Width = 5, Height = 10 };
IArea circle = new Circle { Radius = 7 };
Console.WriteLine($"Rectangle Area: {rectangle.CalculateArea()}");
Console.WriteLine($"Circle Area: {circle.CalculateArea()}");
}
}
}
```

OUTPUT



RESULT

Therefore, a C# program to calculate area of an object using multiple inheritance has been executed and the output is verified successfully.

Exp No: 3 **Date:**

IMPLEMENTATION OF MULTI-THREADING

AIM

To write a C# program to implement multithreading.

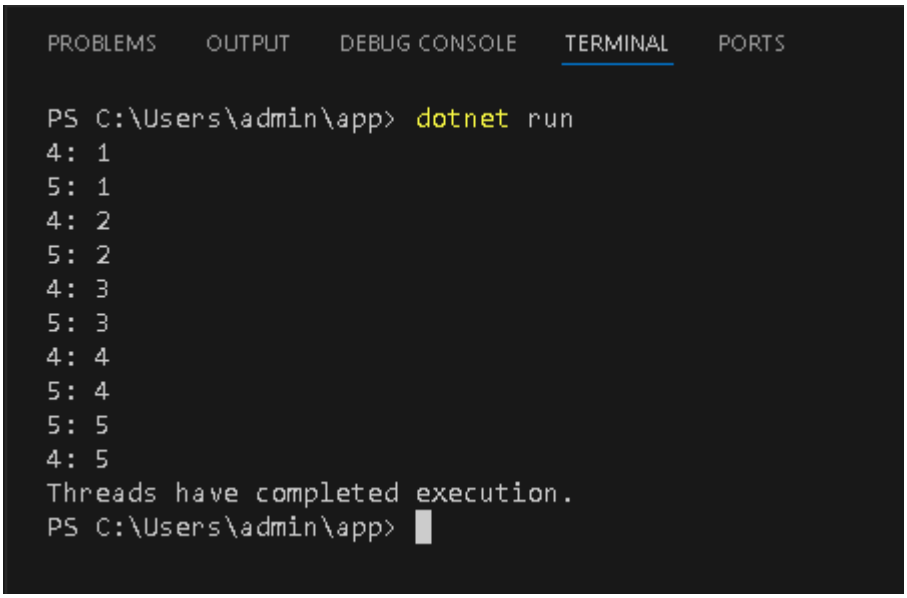
ALGORITHM

1. Start the program.
2. Create a class.
3. Create a threads thread1, thread2 and give name inside a main.
4. Create 3 methods for executing something.
5. Call 3 methods using 3 different threads.
6. Stop the program.

PROGRAM

```
using System;
using System.Threading;
namespace MultiThreadingExample {
class Program {
static void Main(string[] args) {
Thread thread1 = new Thread(PrintNumbers);
Thread thread2 = new Thread(PrintNumbers);
thread1.Start();
thread2.Start();
thread1.Join();
thread2.Join();
Console.WriteLine("Threads have completed execution.");
}
static void PrintNumbers() {
for (int i = 1; i <= 5; i++) {
Console.WriteLine($"{Thread.CurrentThread.ManagedThreadId}: {i}");
Thread.Sleep(500); // Simulate some work
}
}
}
}
```

OUTPUT



RESULT

Therefore, a C# program to implement multithreading has been executed and the output is verified successfully.

Exp No: 4 **Date:**

EXCEPTION HANDLING

AIM

To write a C# program to perform exception handling.

ALGORITHM

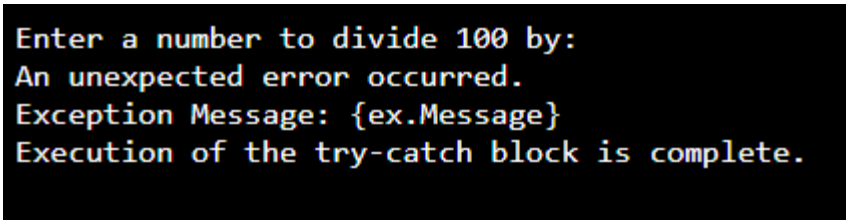
1. Start the program.
2. Create a class.
3. Create an array.
4. Use try, catch blocks to handle exceptions.
5. Stop the program.

PROGRAM

```
using System;
namespace ExceptionHandlingExample {
class Program {
static void Main(string[] args) {
// Example usage of exception handling
try {
// Code that may throw an exception
Console.WriteLine("Enter a number to divide 100 by:");
int divisor = int.Parse(Console.ReadLine());
// This will throw DivideByZeroException if the user inputs 0
int result = 100 / divisor;
Console.WriteLine("Result: " + result);
// Properly format the result
}
catch (DivideByZeroException ex) {
// Handle DivideByZeroException
Console.WriteLine("Error: Cannot divide by zero.");
Console.WriteLine("Exception Message: {ex.Message}");
}
catch (FormatException ex) {
// Handle FormatException
Console.WriteLine("Error: Input was not a valid number.");
Console.WriteLine("Exception Message: {ex.Message}");
}
catch (Exception ex) {
// Handle any other exceptions
Console.WriteLine("An unexpected error occurred.");
```

```
Console.WriteLine("Exception Message: {ex.Message}");
}
finally {
// This block will always execute, regardless of whether an exception was thrown
Console.WriteLine("Execution of the try-catch block is complete.");
}
}
}
}
```

OUTPUT



RESULT

Therefore, a C# program to perform exceptional handling has been executed and the output is verified successfully.

Exp No: 5 **Date:**

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.

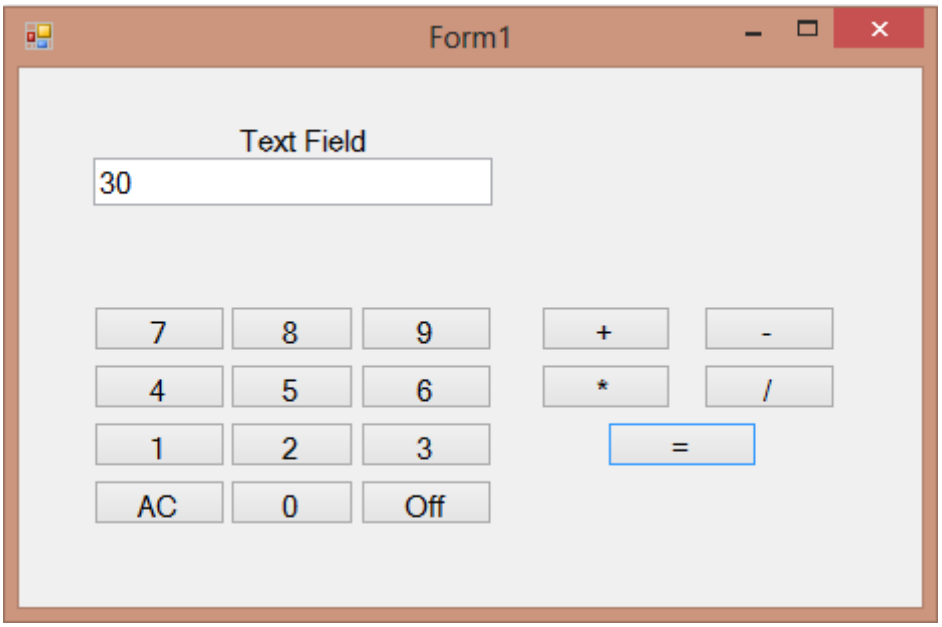
PROGRAM

```
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

Therefore, a VB.net program to create a calculator has been executed and the output is verified successfully.

Exp No: 6 **Date:**

FILE HANDLING

AIM

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

ALGORITHM

1. Start the program.
2. Create an object using its syntax.
3. Use file mode, file access to create or open a file.
4. Stop the program.

PROGRAM

To READ FILE

```
Imports System.IO
Module Module1
Sub Main()
Dim str As String = Nothing
Try
str = File.ReadAllText("D:\myFile.txt")
Console.WriteLine("Content of file: {0}", str)
Catch ex As FileNotFoundException
Console.WriteLine("File does not exist")
End Try End Sub
End Module
```

TO MODIFY OR APPEND

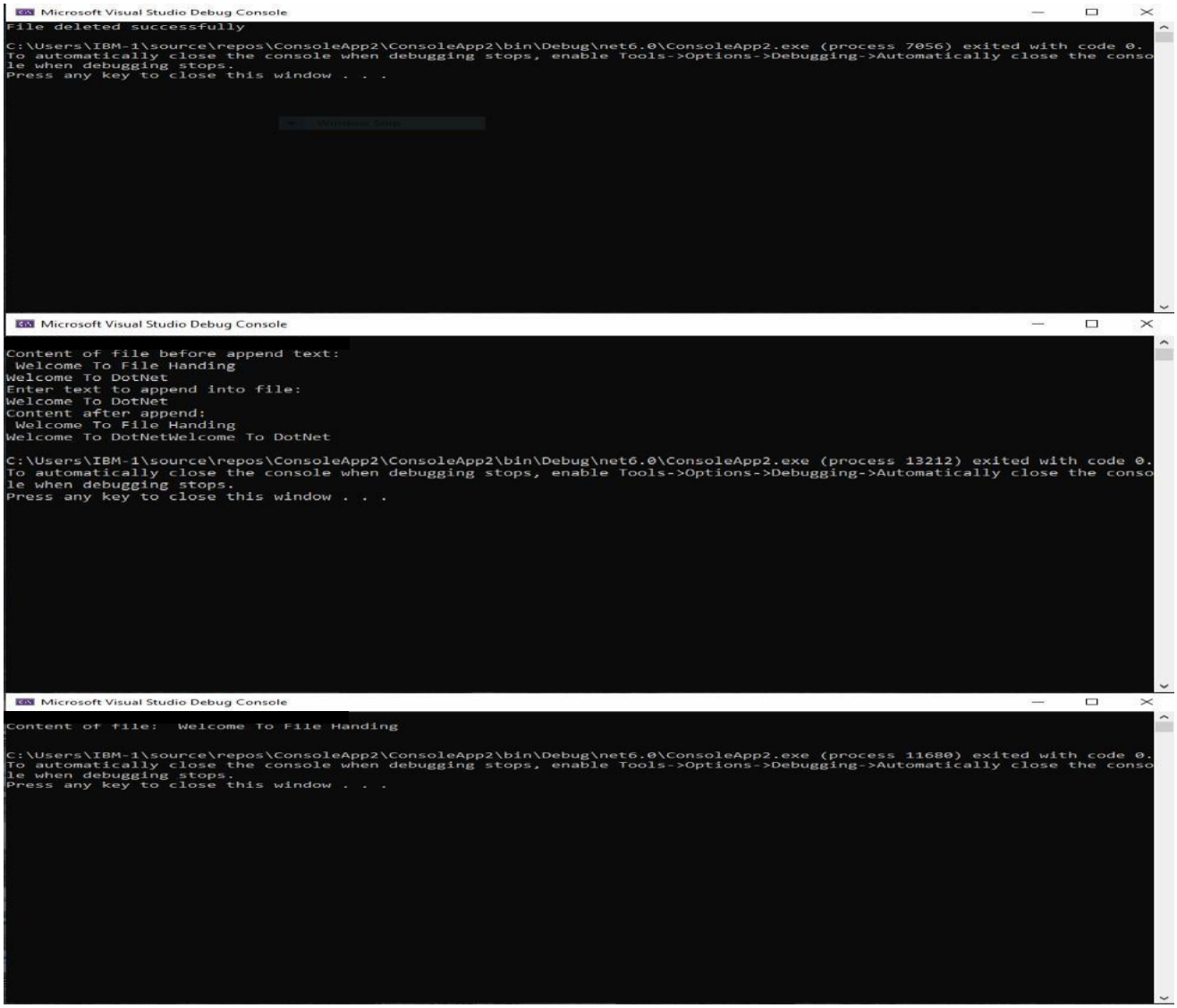
```
Imports System.IO
Module Module1
Sub Main()
Dim str As String = Nothing Try
str = File.ReadAllText("D:\myFile.txt")
Console.WriteLine("Content of file before append text: ")
Console.WriteLine(str)
Console.WriteLine("Enter text to append into file:")
```

```
str = Console.ReadLine()
File.AppendAllText("D:\myFile.txt", str)
Console.WriteLine("Content after append:")
str = File.ReadAllText("myFile.txt")
Console.WriteLine(str)
Catch ex As FileNotFoundException
Console.WriteLine("File does not exist")
End Try End Sub
End Module
```

To DELETE FILE

```
'VB.Net program to delete a specified file.
Imports System.IO
Module Module1
Sub Main()
Try
File.Delete("D:\myFile.txt")
Console.WriteLine("File deleted successfully")
Catch ex As FileNotFoundException
Console.WriteLine("File does not exist")
End Try End Sub
End Module
```

OUTPUT



RESULT

Therefore, the program for file handling has been executed and the output is verified successfully.

Exp No: 7 (i) **Date:**

EMPLOYEE DETAILS

AIM

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

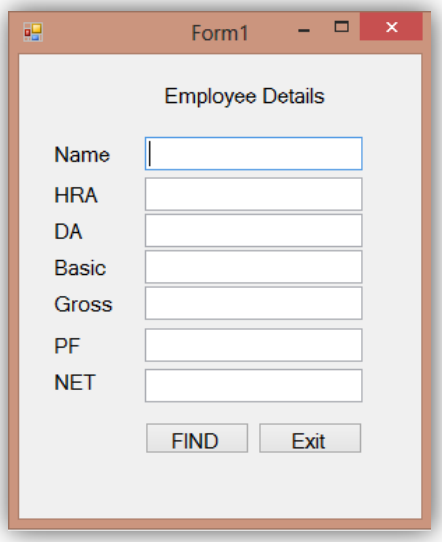
ALGORITHM

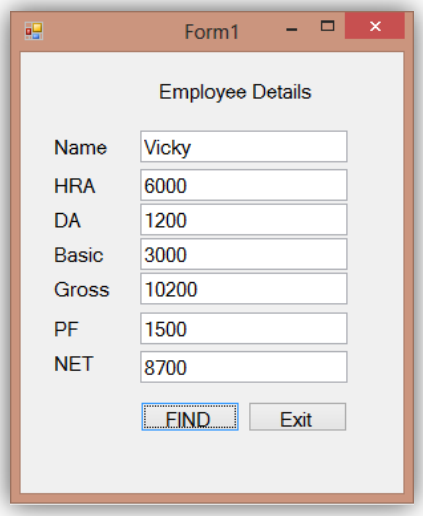
5. Start the program.
6. Create the GUI for the user.
7. Design some label, textbox & button.
8. After giving values by the user, system will find the gross &net salary of employee.
9. After clicking end button the GUI will be exit.
10. Stop the program.

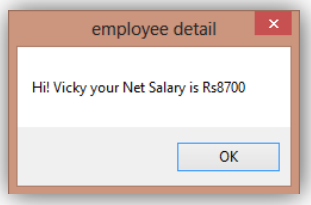
PROGRAM

```
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







RESULT

Therefore, a VB.net program to find the net salary of employee has been executed and the output is verified successfully.

Exp No: 7(ii) **Date:**

VOTERS

AIM

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

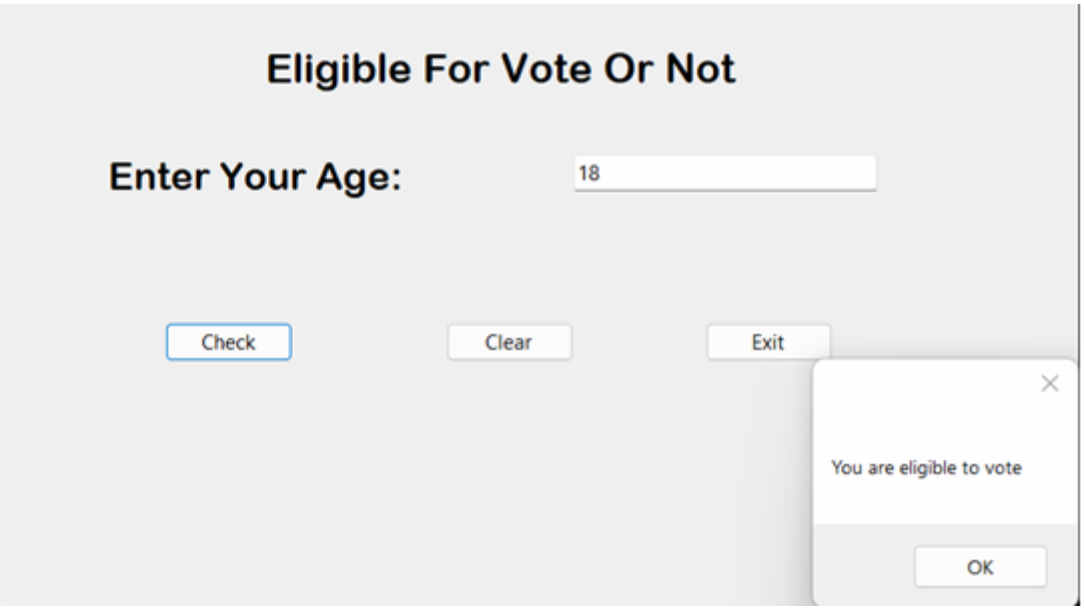
ALGORITHM

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 users are eligible for vote or not.
6. Stop the program.

PROGRAM

```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
Dim age As Integer
age = Val(TextBox1.Text)
If age >= 18 Then
MessageBox.Show("You are eligible to vote")
Else
MessageBox.Show("You are not not eligible to vote")
End If
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
TextBox1.Clear()
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
Close()
End Sub
End Class
```

OUTPUT



RESULT

Therefore, a VB.net program to find the eligibility of voters through exception handling has been executed and the output is verified successfully.

Exp No: 7(iii) **Date:**

STUDENT STATUS

AIM

To write a C# program to find student status through exception handling.

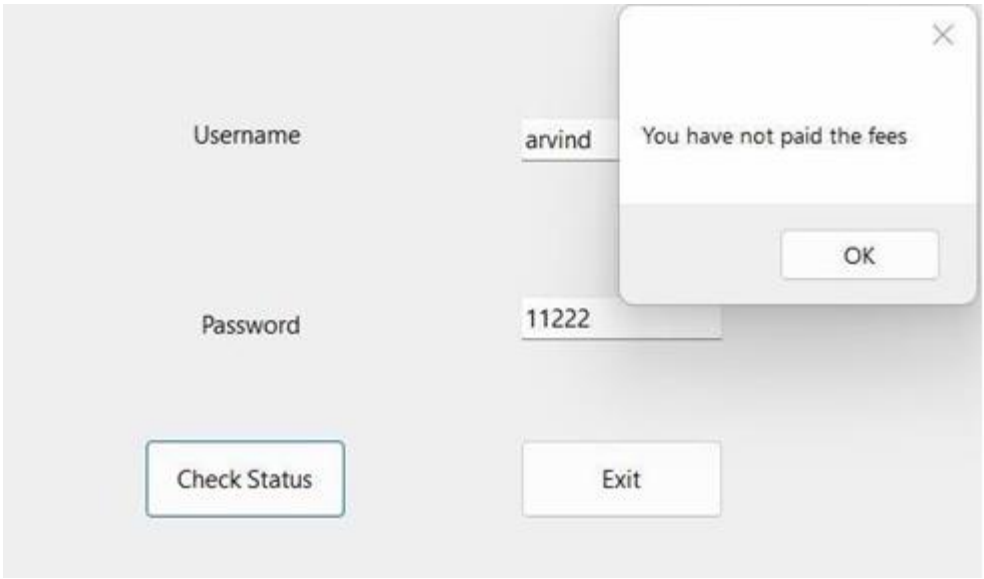
ALGORITHM

1. Start the program.
2. Create a list of students with four variables(Id, name department and semester).
3. Iterate through the student details by using for loop and get the student details by using selectclause
4. Display the student details
5. Stop the program

PROGRAM

```
Public Class Form1

Private Sub

Button1_Click(sender As Object, e As EventArgs) Handles

Button1.Click

MessageBox.Show("You have not paid the fees")

End Sub

Private Sub

Button2_Click(sender As Object, e As EventArgs) Handles

Button2.Click

Close()

End Sub

End Class
```

OUTPUT



RESULT

Therefore, a VB.net program to find the students status through exception handling has been executed and the output is verified successfully.

Exp No: 8 **Date:**

SUPER MARKET

AIM

To write a asp.net program to create super market.

ALGORITHM

1. Start the program.
2. Create a dynamic web page using html codes
3. Design a label textbox button
4. After clicking the submit button, we will get all the details about the super market
5. Stop the program

PROGRAM

WINFORM.ASPX.CS:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
usingSystem.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
namespace WebApplication3{
public partial class WebForm1 : System.Web.UI.Page{
SqlConnection con = new Sql Connection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sony\source\repos\WebA
pplicati on3\WebApplication3\App_Data\Database1.mdf;Integrated Security=True");
protected void Page_Load (object sender, EventArgs e){
if (con.State == ConnectionState.Open){
con.Close();
}
con.Open();
disp_data();
}
protected void Button1_Click(object sender, EventArgs e){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "insert into Market2 values ('"+custname.Text+"',
'"+phonenumber.Text+"', '"+city.Text+"', '"+itemname.Text+"', '"+price.Text+"')";
```

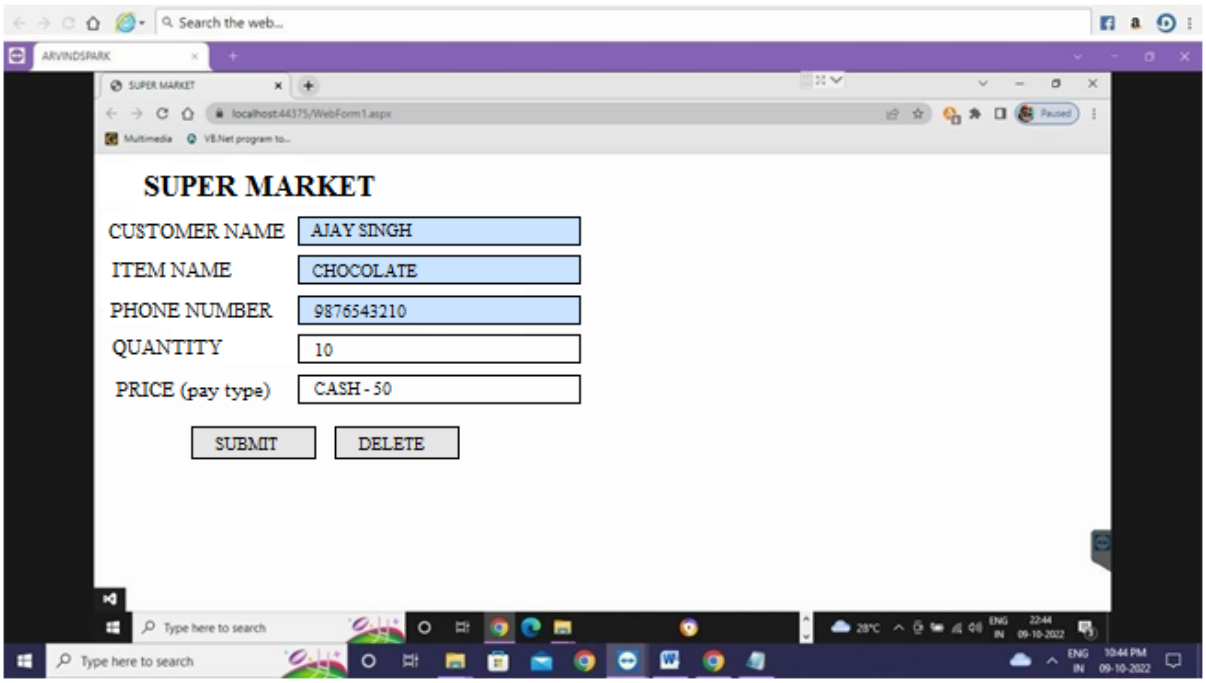
```
cmd.ExecuteNonQuery();
custname.Text = "";  phonenumber.Text = ""; city.Text = ""; itemname.Text = "";
price.Text = ""; disp_data();
}
public void disp_data(){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select * from Market2";
cmd.ExecuteNonQuery();
DataTabledt = newDataTable();
SqlDataAdapter da = newSqlDataAdapter(cmd);
da.Fill(dt);
//GridView1.DataSource = dt;
// GridView1.DataBind();
}

protectedvoid Button2_Click(object sender, EventArgs e){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "delete from Market2 where custname='"+custname.Text +"'";
cmd.ExecuteNonQuery();
custname.Text = ""; disp_data();
}
}
}
```

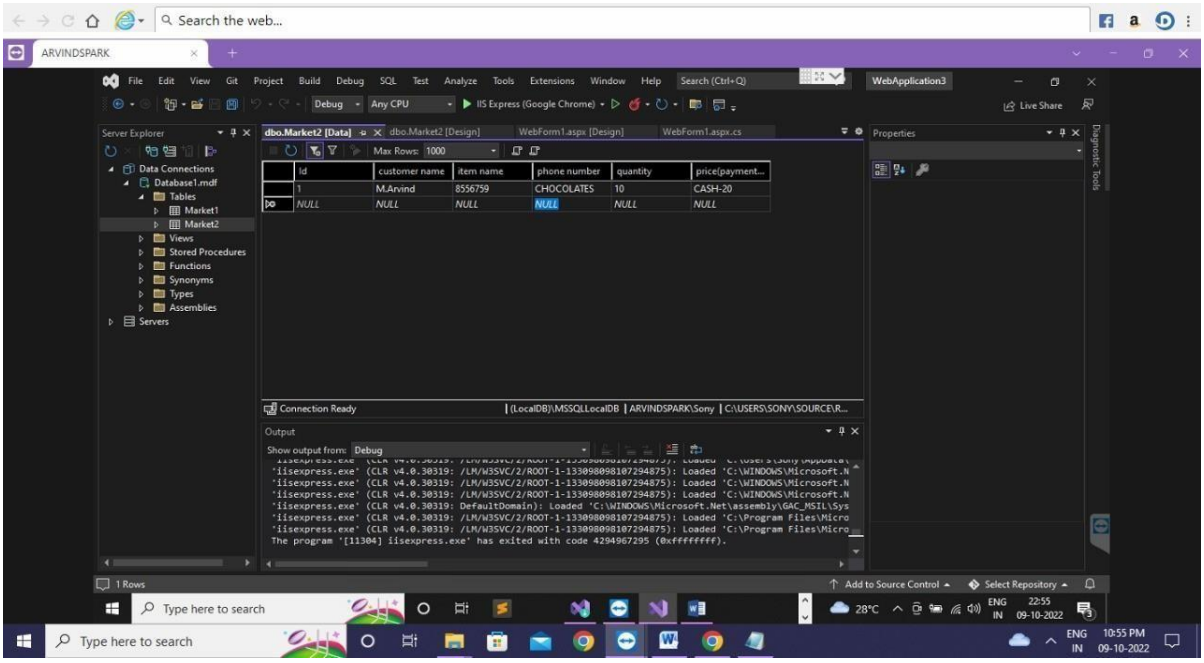
WINFORM.ASPX:

```
<%@Page Language = "C#"Auto Event Wire up="true" Code Behind = "Web
Form1.aspx.cs" Inherits =" We bApplication3.WebForm1"%>

<!DOCTYPEhtml>
<html xmlns="http://www.w3.org/1999/xhtml">
<head run at="server">
<title>SUPER MARKET</title>
</head>
<h1>SUPER MARKET</h1>
<body>
<for mid="form1" run at="server">
<div>
<table>
<tr>
<td> CUSTOMER NAME</td>
<td> <asp:TextBoxID="custname" run at="server"> </asp:TextBox></td>
</tr>
```

```
<tr>
<td>ITEM NAME</td>
<td><asp:TextBoxID="city" run at ="server"></asp:TextBox></td>
</tr>
<tr>
<td>PHONE NUMBER</td>
<td><asp:TextBoxID="phone number" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>QUANTITY</td>
 <td><asp:TextBoxID="item name" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>PRICE(payment type)</td>
<td><asp:TextBoxID="price" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td colspan="2"align="center">
<asp:ButtonID="Button1"runat="server"Text="SUBMIT"OnClick="Button1_Click"/>
<asp:ButtonID="Button2"runat="server"Text="DELETE"OnClick="Button2_Click"/>
</td>
</tr>
</table>
<br/>
</div>
</form>
</body>
</html>
```

OUTPUT





RESULT

Therefore, a VB.net program to create super market has been executed and the output is verified successfully.

Exp No: 9 **Date:**

HOTEL MANAGEMENT SYSTEM

AIM

To write a asp.net program to create a hotel management system.

ALGORITHM

1. Start the program.
2. Create a dynamic web page using html codes.
3. Design the label textbox and button for getting output.
4. Add all the details Customer Name, Customer Number and other details for the program.
5. Stop the program.

PROGRAM

WEBFROM1.ASPX :

```
<% @Page Language="C#" Auto Event Wireup="true" Code Behind="WebForm1.aspx.cs"
Inherits="We bApplication3.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head run at="server">
<title>HOTEL MANAGEMENT SYSTEM</title>
</head>
<h1>HOTEL MANAGEMENT SYSTEM</h1>
<body>
<for mid="form1" run at="server">
<div>
<table>
<tr>
<td>CUSTOMER NAME</td>
<td><asp:TextBoxID="custname" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>CUSTOMER NUMBER</td>
<td><asp:TextBoxID="city" run at="server"></asp:TextBox></td>
 </tr>
<tr>
<td>ROOM NUMBER</td>
<td><asp:TextBoxID="phone number" run at="server"></asp:TextBox></td>
</tr>
<tr>
```

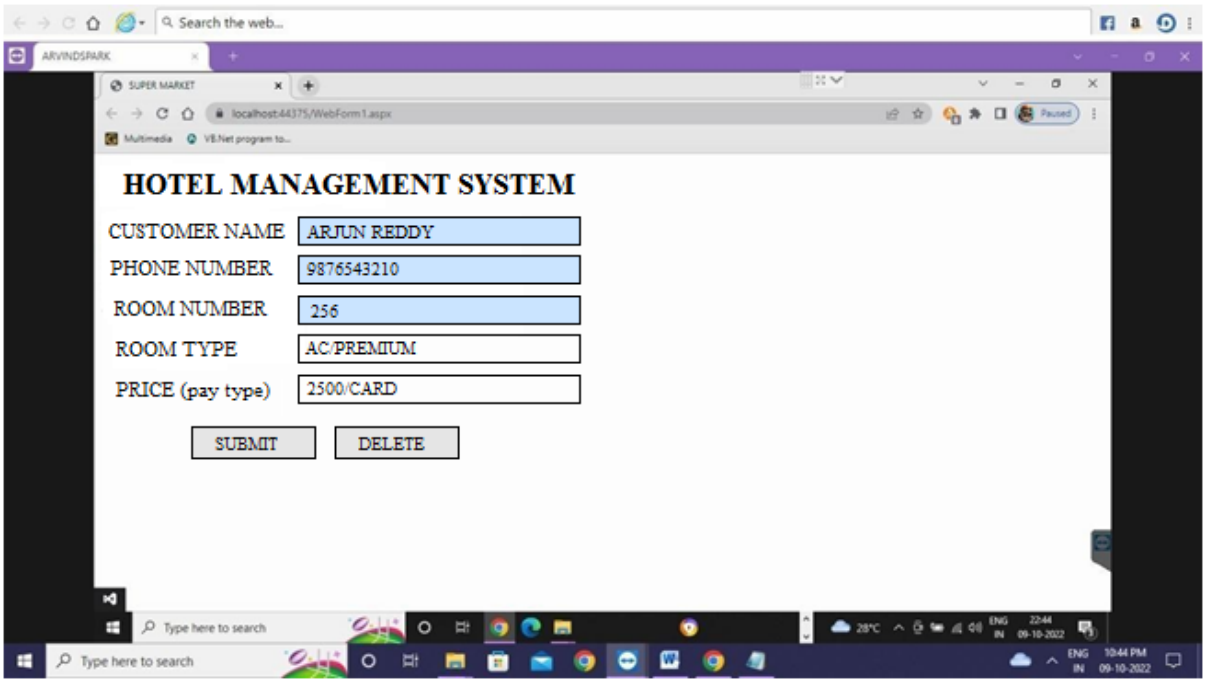
```
<td>ROOM TYPE</td>
<td><asp:TextBoxID="item name" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>PRICE(payment type)</td>
<td><asp:TextBoxID="price" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td cols pan="2"align="center">
<asp:ButtonID="Button1" run at="server" Text="SUBMIT" OnClick="Button1_Click"/>
<asp:ButtonID="Button2" run at="server" Text="DELETE" OnClick="Button2_Click"/>
</td>
</tr>
</table>
<br/>
</div>
</form>
</body>
</html>
```

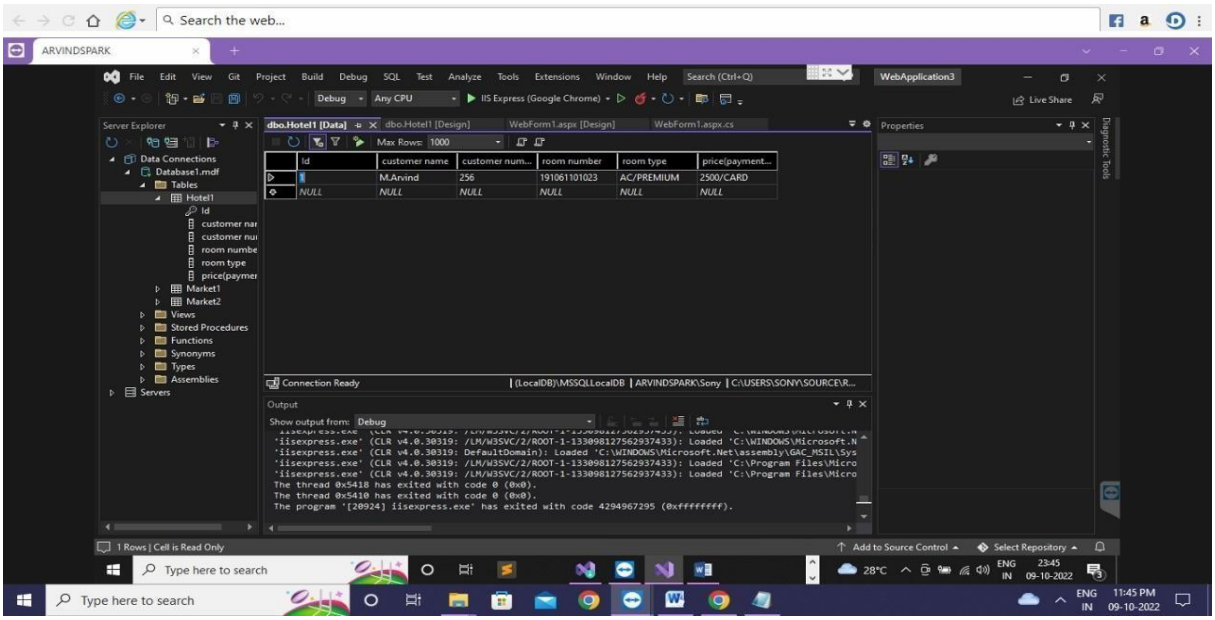
WEBFORM1.ASPX.CS:

```
using System;
using System.Collections.Generic;
using System.Linq; usingSystem.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
namespace WebApplication3{
public partial class WebForm1 : System.Web.UI.Page{
SqlConnection con = newSqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;
AttachDbFilename=C:\Users\Sony\source\repos\WebApplicati
on3\WebApplication3\App_Data\Database1.mdf; Integrated Security=True"); protected void
Page_Load(object sender, EventArgs e){
if (con.State == Connection State.Open){
con.Close();
}
con.Open();
disp_data();
}
protected void Button1_Click(object sender, EventArgs e){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
```

```
cmd.CommandText = "insert into Hotel1
values('"+custname.Text+"','"+phonenumber.Text+"','"+city.Text+"','"+itemname.Text+"','"+
price.Text+"')";
cmd.ExecuteNonQuery();
custname.Text = "";
phonenumber.Text = "";
city.Text = "";
itemname.Text = "";
price.Text = "";
disp_data();
}
public void disp_data(){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select * from Hotel1";
cmd.ExecuteNonQuery();
DataTabledt = newDataTable();
SqlDataAdapter da = newSqlDataAdapter(cmd);
da.Fill(dt);
//GridView1.DataSource = dt;
// GridView1.DataBind();
}
protected void Button2_Click(object sender, EventArgs e){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "delete from Hotel1 where custname='"+custname.Text +"'";
cmd.ExecuteNonQuery();
custname.Text = "";
disp_data();
}
}
}
```

OUTPUT





RESULT

Therefore, a VB.net program to create hotel management system has been executed and the output is verified successfully.

Exp No: 10 **Date:**

STUDENT ATTENDENCE CALCULATION

AIM

To write a asp.net program for student attendance calculation.

ALGORITHM

1. Start the program.
2. Create an empty C# web application.
3. Create a database using “sql server database”.
4. Create a web form.
5. Drag drop buttons, textbox, labels and grid view.
6. End the program.

PROGRAM

WEBFORM1.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Diagnostics;
namespace WebApplication6{
public partial class WebForm1 : System.Web.UI.Page{
SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Sony\source\repos\Web
pplicati on6\WebApplication6\App_Data\Database1.mdf;Integrated Security=True");
protected void Page_Load(object sender, EventArgs e){
if (con.State == ConnectionState.Open){
con.Close();
}
con.Open(); disp_data();
}
protected void Button1_Click(object sender, EventArgs e){
SqlCommand cmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "insert into Attendance1 values('" +name.Text + "','" +regno.Text +
"','" +wd.Text+ "', '" +pd.Text+ "', '" +ad.Text + "', '"+percent.Text+"','"+fine.Text+"')";
```

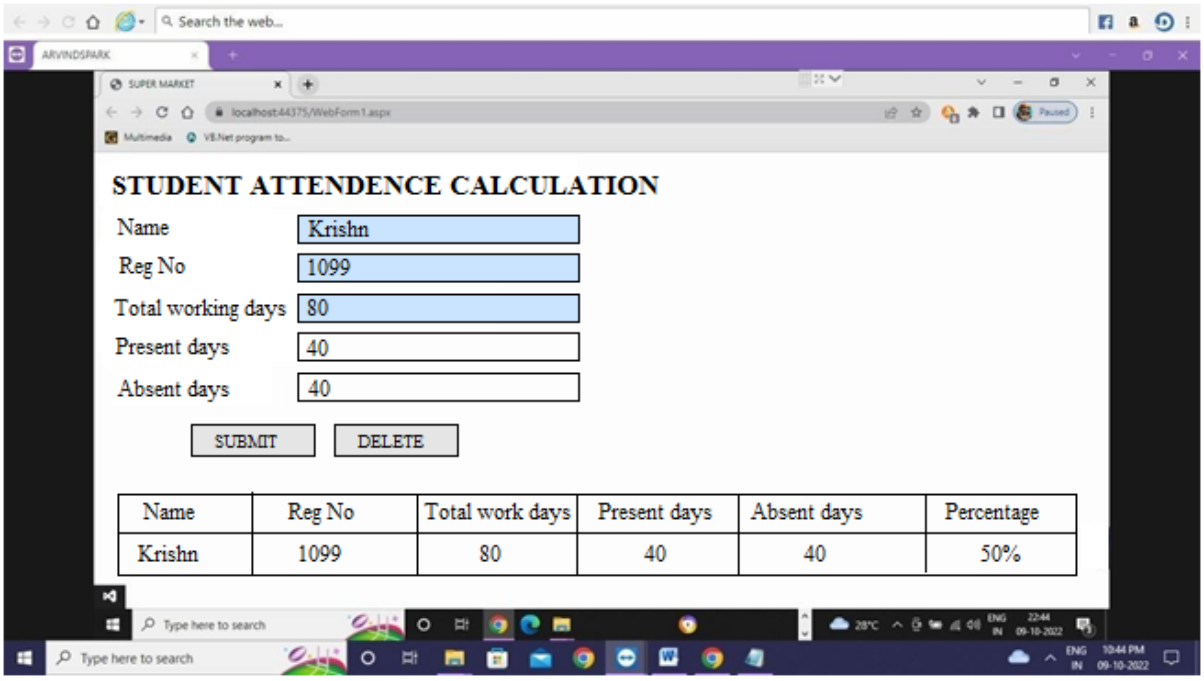
```
cmd.ExecuteNonQuery();
name.Text = ""; regno.Text = ""; wd.Text = "";
pd.Text = "";
ad.Text = ""; percent.Text = ""; fine.Text = "";
disp_data();
}
public void disp_data(){
SqlCommand cmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select * from Attendance1";
cmd.ExecuteNonQuery();
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
GridView1.DataSource = dt;
GridView1.DataBind();
}
}
}
```

WEBFORM.ASPX:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication6.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>STUDENT ATTENDENCE CALCULATION</title>
</head>
<body>
<h1>STUDENT ATTENDENCE CALCULATION</h1>
<h1>SUVEDHINI REDDY 191061101602</h1>
<form id="form1" runat="server">
<div>
<table>
<tr>
<td>Name</td>
<td><asp:TextBox ID="name" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td>Reg No</td>
<td><asp:TextBox ID="regno" runat="server"></asp:TextBox></td>
</tr>
<tr>
```

```
<td>Total No of working days</td>
<td><asp:TextBox ID="wd" runat="server"></asp:TextBox></td>
 </tr>
<tr>
<td>Present Days</td>
<td><asp:TextBox ID="pd" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td>Absent Days</td>
<td><asp:TextBox ID="ad" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td>Percentage</td>
<td><asp:TextBox ID="percent" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td>Fine</td>
<td><asp:TextBox ID="fine" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td colspan="2" align="center">
<asp:Button ID="Button1" runat="server" Text="SUBMIT" OnClick="Button1_Click" />
<asp:Button ID="Button2" runat="server" Text="DELETE" />
</td>
</tr>
</table>
<br />
<asp:GridView ID="GridView1" runat="server"></asp:GridView>
</div>
</form>
</body>
</html>
```

OUTPUT



RESULT

Therefore, a VB.net program for student attendance calculation has been executed and the output is verified successfully.

Exp No: 11 **Date:**

HOSPITAL MANAGEMENT SYSTEM

AIM

To write a asp.net program for hospital management system.

ALGORITHM

1. Start the program.
2. Create an empty C# web application.
3. Create a database using “sql server database”.
4. Create a web form.
5. Drag drop buttons, textbox, labels and grid view.
6. End the program.

PROGRAM

WEBFORM1.aspx

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeBehind="WebForm1.aspx.cs"Inheri
ts="We bApplication3.WebForm1"%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head run at="server">
<title>HOSPITAL MANAGEMENT SYSTEM</title>
</head>
<h1>HOSPITAL MANAGEMENT SYSTEM</h1>
<body>
<for mid="form1" run at="server">
<div>
<table>
<tr>
<td>Patient Name</td>
<td><asp:TextBoxID="custname" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>PhoneNumber</td>
<td><asp:TextBoxID="phone number" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>Address</td>
<td><asp:TextBoxID="city" run at="server"></asp:TextBox></td>
</tr>
<tr>
```

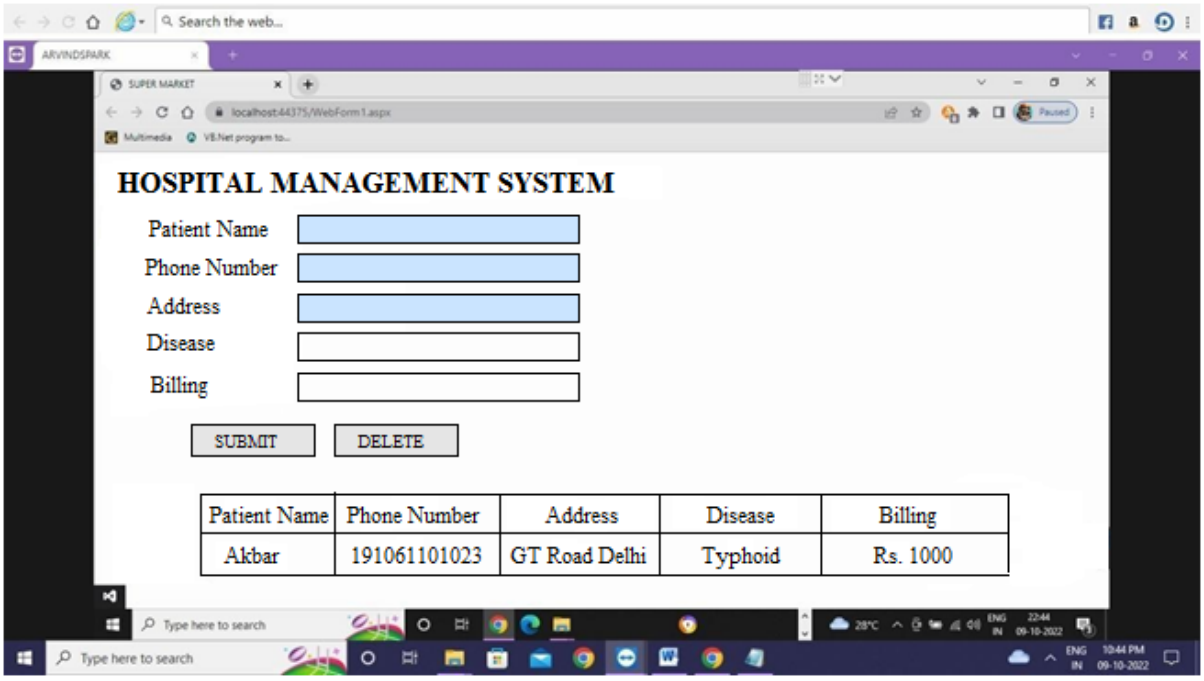
```
<td>Disease</td>
<td><asp:TextBoxID="item name" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td>Billing</td>
<td><asp:TextBoxID="price" run at="server"></asp:TextBox></td>
</tr>
<tr>
<td colspan="2"align="center">
<asp:ButtonID="Button1" run at="server" Text="SUBMIT" OnClick="Button1_Click"/>
<asp:ButtonID="Button2" run at="server" Text="DELETE" OnClick="Button2_Click"/>
</td>
</tr>
</table>
<br/>
<asp:Grid View ID="Grid View1" run at="server"></asp:GridView>
</div>
</form>\
</body></html>Webform.aspx.cs using System;
```

WEBFORM.ASPX:

```
using  System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
namespace WebApplication3{
public partial class WebForm1 : System.Web.UI.Page{
Sql Connection con = new Sql Connection(@"Data Source=(LocalDB)\MSSQLLocalDB;
Attach Db File name=C:\Users\Sony\source\repos\WebApplicati
on3\WebApplication3\App_Data\Database1.mdf;
Integrated Security=True");
protected void Page_Load(object sender, EventArgs e){
if (con.State == Connection State.Open){
con.Close();
}
con.Open(); disp_data();
}
protected void Button1_Click(object sender, EventArgs e){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
```

```
cmd.CommandText = "insert into Market1
values('"+custname.Text+"','"+phone
number.Text+"','"+city.Text+"','"+itemname.Text+"','"+price.Te xt+"')";
cmd.ExecuteNonQuery();
cust name.Text = "";
phone number.Text = "";
city.Text = "";
itemname.Text = "";
price.Text = "";
disp_data();
}
public void disp_data(){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select * from Market1";
cmd.ExecuteNonQuery();
Data Table dt = new Data Table();
SqlDataAdapter da = new Sql Data Adapter(cmd);
da.Fill(dt);
Grid View1.DataSource = dt;
Grid View1.DataBind();
}
protected void Button2_Click(object sender, EventArgs e){
SqlCommandcmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "delete from Market1 where custname='"+custname.Text +"'";
cmd.ExecuteNonQuery();
custname.Text = ""; disp_data();
}
}
}
```

OUTPUT



RESULT

Therefore, a VB.net program for hospital management system has been executed and the output is verified successfully.

Exp No: 12 **Date:**

INCOME TAX CALCULATION

AIM

To write an asp.net program for income tax calculation.

ALGORITHM

1. Start the program.
2. Create an empty C# web application.
3. Create a web server.
4. Create a new window C# project within web service (we can create a number of new projects within a single web service).
5. Design the form that means drag an drop buttons, textbox and labels.
6. End the program.

PROGRAM

WEBSERVICE1.asmx.cs:

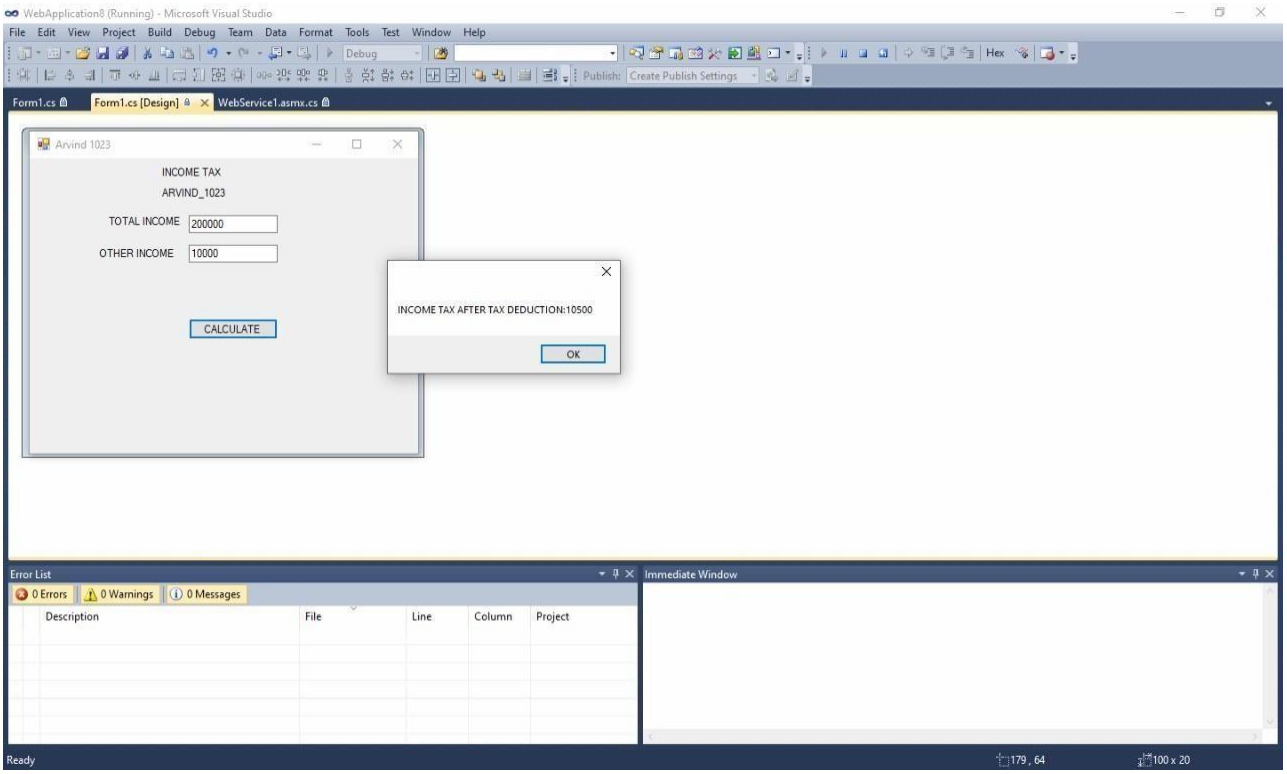
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
namespace WebApplication8{
/// <summary>
/// Summary description for WebService1
/// </summary>

[WebService(Namespace = "http://tempuri.org/")] [WebServiceBinding(ConformsTo =
WsiProfiles.BasicProfile1_1)] [System.ComponentModel.ToolboxItem(false)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the
following line.
// [System.Web.Script.Services.ScriptService]

public class WebService1 : System.Web.Services.WebService{
[WebMethod]
public string HelloWorld(){
return "INCOME TAX";
}
[WebMethod]
public int add(int a,int b){
```

```
int c=a+b;
int g= (c/100)*5;
return (g);
}
}
}
```

Sumit 1023.cs*:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Windows Forms Application1.Service Reference1;
namespace Windows Forms Application1{
public partial class Form1 : Form{
WebService1SoapClient obj;
public Form1(){
Initialize Component();
}
private void button1_Click(object sender, EventArgs e){
int a = Convert.ToInt32(textBox1.Text);
int b = Convert.ToInt32(textBox2.Text); int ans = obj.add(a, b);
MessageBox.Show("INCOME TAX AFTER TAX DEDUCTION:" + ans);
}
private void Form1_Load(object sender, EventArgs e){
obj = new WebService1SoapClient();
}
private void textBox1_TextChanged(object sender, EventArgs e){
}
private void label4_Click(object sender, EventArgs e){
}
}
}
```

OUTPUT



RESULT

Therefore, a VB.net program for income tax calculation has been executed and the output is verified successfully.