VIDYAVAHINI GROUP OF INSTITUTIONS

VIDYAVAHINI FIRST GRADE COLLEGE

(AFFILIATED TO TUMKUR UNIVERSITY)

KUVEMPUNAGAR, TUMKUR-572103

Department of BCA

3rd SEM – C# and Dot net Framework Lab Manual

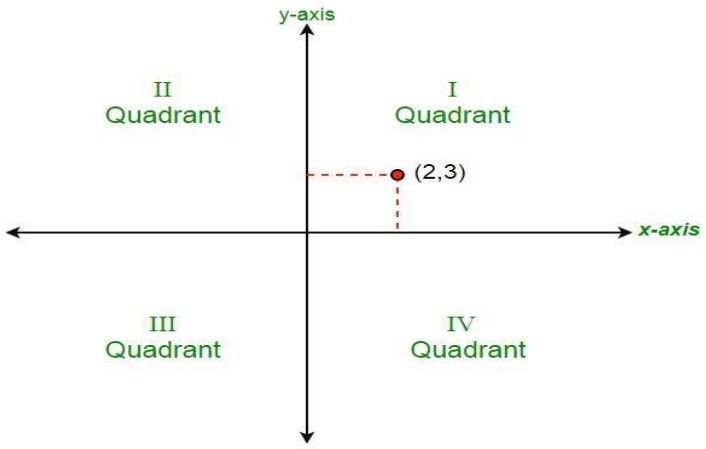
SL.NO.	Title of the Program
1	Program to determine the quadrant of the Cartesian plane using if-else ladder
2	Program to Check whether the alphabet is a Vowel or not using Switch Case in C#
3	To develop a c# application to print the students list using classes and objects
4	To Develop a console application to implement binary operator overloading concept in C#
5	Program to Demonstrate multithreaded programming in c#.net
6	Using Try, Catch and Finally blocks write a C# program to demonstrate error handling
7	To develop a c# console application to implement delegates concept
8	Develop Student Information System in C#.NET that demonstrates the windows Controls.
9	To Design a notepad application to implement menus, custom dialog box and MDI Concepts.
10	Develop a Windows application with database for Student Information System [Insert, Update and Delete Commands].

Lab-1:

Program to determine the quadrant of the Cartesian plane using if-else ladder

Explanation:

Given co-ordinates (x, y), determine the quadrant of the Cartesian plane.



There are 9 conditions that needs to be checked to determine where does the points lies –

If in first quadrant then, $x > 0$ and $y > 0$

If in second quadrant then, $x < 0$ and $y > 0$

If in third quadrant then, $x < 0$ and $y < 0$

If in fourth quadrant then, $x > 0$ and $y < 0$

If in positive x-axis then, $y = 0$ and $x > 0$

If in negative x-axis then, $y = 0$ and $x < 0$

If in positive y-axis then, $x = 0$ and $y > 0$

If in negative y-axis then, $x = 0$ and $y < 0$

If at origin then, x = 0 and y = 0

Code:

```
using System;
 class Demo {

   // Function to check quadrant
   static void quadrant(int x, int y)
   {
      if (x > 0 && y > 0)
         Console.WriteLine("lies in First quadrant");

      else if (x < 0 && y > 0)
         Console.WriteLine("lies in Second quadrant");

      else if (x < 0 && y < 0)
         Console.WriteLine("lies in Third quadrant");

      else if (x > 0 && y < 0)
         Console.WriteLine("lies in Fourth quadrant");

      else if (x == 0 && y > 0)
         Console.WriteLine("lies at positive y axis");

      else if (x == 0 && y < 0)
         Console.WriteLine("lies at negative y axis");

      else if (y == 0 && x < 0)
         Console.WriteLine("lies at negative x axis");
```

```
        else if (y == 0 && x > 0)
            Console.WriteLine("lies at positive x axis");

        else
            Console.WriteLine("lies at origin");
    }

    public static void Main()
    {
        int x = 1, y = 1;

        // Function Calling
        quadrant(x, y);
    }
}
```

Output:

Enter two numbers:

8

-9

Points are in fourth quadrant

Lab-2:

Check whether the alphabet is a Vowel or not using Switch Case in C#

```
using System;

class Demo
```

```
{
    static void Main(string[] args)
    {
        char ch;
        Console.WriteLine("Enter the Charachter ");
        ch = char.Parse(Console.ReadLine());
        Console.WriteLine("Checking Charachter.....");
        switch (ch)
        {
            case 'a':
            case 'A':
            case 'E':
            case 'e':
            case 'I':
            case 'i':
            case 'o':
            case 'O':
            case 'u':
            case 'U':
            Console.WriteLine("{0} is a vowel", ch);
             break;
```

```
            default: Console.WriteLine("{0} is not a vowel", ch);
             break;
        }
        Console.ReadKey();
    }
  }
```

Output:

```
Enter a character:
u
u is a vowel
Enter a character:
A
A is a vowel
Enter a character:
j
j  is not a vowel
```

Lab-3:

To develop a c# application to print the students list using classes and objects

```
using System;
  public class Student
```

```
{
    public int id;
    public String name;
    public void insert(int i, String n)
    {
        id = i;
        name = n;
    }
    public void display()
    {
        Console.WriteLine(id + " " + name);
    }
}
class Test{
    public static void Main(string[] args)
    {
        Student s1 = new Student();
        Student s2 = new Student();
        s1.insert(101, "Virat");
        s2.insert(102, "Max");
        s1.display();
        s2.display();
    }
}
```

Output:

101 Virat

102 Max

Lab-4:

To develop a console application to implement binary operator overloading concept in C#

```
using System;
class Complex
{
    private int x;
    private int y;
    public Complex()
}
public Complex(int i, int j)
{
    x = i;
    y = j;
}
public void ShowXY()
{
    Console.WriteLine("{0} {1}", x, y);
}
public static Complex operator +(Complex c1, Complex c2)
{
    Complex temp = new Complex();
    temp.x = c1.x + c2.x;
    temp.y = c1.y + c2.y;
    return temp;
}
public static Complex operator -(Complex c1, Complex c2)
{
    Complex temp = new Complex();
```

```
        temp.x = c1.x - c2.x;
        temp.y = c1.y - c2.y;
        return temp;
    }

    }
    class MyClient
    {
        public static void Main()
        {
            Complex c1 = new Complex(10, 20);
            c1.ShowXY(); // displays 10 & 20
            Complex c2 = new Complex(20, 30);
            c2.ShowXY(); // displays 20 & 30
            Complex c3 = new Complex();
            c3 = c1 + c2;
            c3.ShowXY(); // dislplays 30 & 50
            Complex c4 = new Complex();
            c4 = c1 - c2;
            c4.ShowXY();
        }
    }
```

Output:

```
10+i20
20+i30
30+i50
-10+i -10
```

Lab-5:

Demonstrate multithreaded programming in c#.net

```
using System;
using System.Threading;
class Program
{
   public static void Main()
   {
      Thread ThreadObject1 = new Thread(Example1); //Creating the Thread
      Thread ThreadObject2 = new Thread(Example2);
      ThreadObject1.Start(); //Starting the Thread
      ThreadObject2.Start();
   }
   static void Example1()
   {
      Console.WriteLine("Thread1 Started");
      for (int i = 0; i <= 5; i++)
      {
         Console.WriteLine("Thread1 Executing");
         Thread.Sleep(5000); //Sleep is used to pause a thread and 5000 is
MilliSeconds that means 5 Seconds
      }
   }
   static void Example2()
   {
      Console.WriteLine("Thread2 Started");
      for (int i = 0; i <= 5; i++)
      {
         Console.WriteLine("Thread2 Executing");
```

```
            Thread.Sleep(5000);
        }
    }
}
```

Output:

```
Thread 1 started
Thread 1 is Executing
Thread 2 started
Thread 2 is Executing
Thread 2 is Executing
Thread 1 is Executing
Thread 2 is Executing
Thread 1 is Executing
Thread 2 is Executing
Thread 1 is Executing
Thread 2 is Executing
Thread 1 is Executing
```

Lab-6:

Using Try, Catch and Finally blocks write a C# program to demonstrate error handling

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

        Console.WriteLine("Enter first number:");
        int Number1 = int.Parse(Console.ReadLine());
```

```
        Console.WriteLine("Enter second number:");
        int Number2 = int.Parse(Console.ReadLine());

        try
        {
            // code that may raise raise an exception
            int Result = Number1 / Number2;
            Console.WriteLine("Division of two numbers is: " + Result);
        }

        // this catch block gets executed only when an exception is raised
        catch (Exception e)
        {
            Console.WriteLine("An exception occurred: " + e.Message);
        }

        finally
        {
            // this code is always executed whether of exception occurred or not
            Console.WriteLine("Sum of two numbers is: " + (firstNumber +
    secondNumber));
        }
    }
}
```

Output:

```
Enter first number:
100
Enter Second number:
20
Division of two number is:5
```

```
Sum of 2 numbers is 120
Enter first number:
100
enter Second number:
0
An exception is OccuredAttempted to divide by zero.
sum of 2 number is 100
```

Lab-7:

To develop a c# console application to implement delegates concept

```
using System;
namespace DelegateDemo
{
    //defining delegate
    public delegate void AddDelegate(int x, int y);
    public delegate string SayDelegate(string Name);
    class Program
    {
        public void AddNums(int a,int b)
        {
            Console.WriteLine(a+b);
        }
        public static string SayHello(string Name)
        {
            return "hello" +Name;
        }
```

```
        static void Main(string[] args)
        {
            Program p=new Program();
            //instatiating the delegate
            AddDelegate ad=new AddDelegate(p.AddNums);
            //call the method through delegate
            ad(100,50);
            SayDelegate sd=new SayDelegate(Program.SayHello);
            string str=sd("Dev");
            Console.WriteLine(str);
        }
    }
}
```

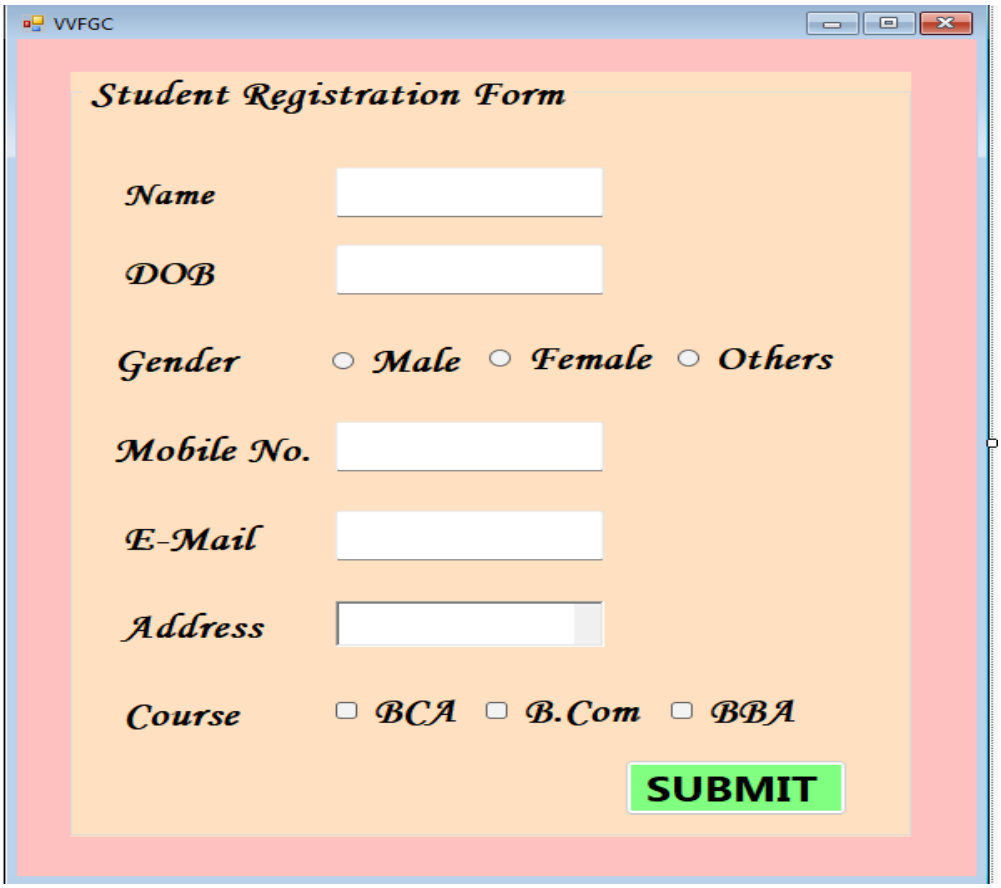
Output:

300

Hello Dev

Lab-8:

Develop Student Information System in C#.NET that demonstrates the windows Controls.



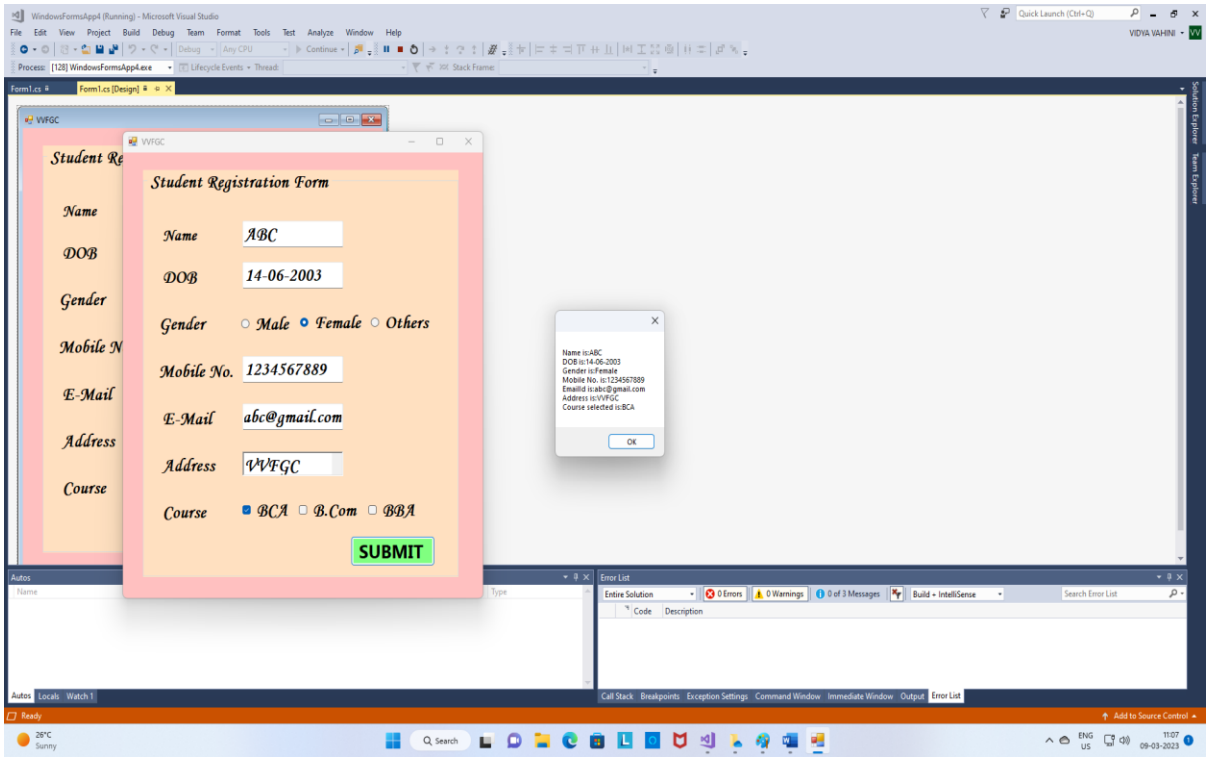
- **Step 1:** Open new Project go to Visual Studio => New=> Project=> select C#.net => Windows Form Application => form window will be opened.
- **Step 2:** We need Label, Textbox, Rich Textbox, Button, Check box , Radio button and Group box Controls to design the form.
- **Step 3:** After Drag and Drop all these form looks like this.
- **Step 4:** We need 7 labels controls and renamed as Name, DOB, Gender, Mobile Number, E-Mail, Address and Course.
- **Step 5:** We need 4 Text boxes and 1 Rich Text box for Address.
- **Step 6:** We need 3 Radio buttons like Named as Male, Female and Others.
- **Step 7:** We need 3 Check boxes named as BC, BCom and BBA.
- **Step 8:** We need 1 button Control named as Submit.
- **Step 9:** Now We Enter all values of labels for that needs to be write coding part

When Double click on the Submit button control.

```
Private void button1_Click(object sender, EventArgs e)
    {
        String name, dob, gender, email, address, courses;
        long mobile;
        name = textBox1.Text;
        dob = textBox2.Text;
        email = textBox4.Text;
        address = richTextBox1.Text;
        mobile = long.Parse(textBox3.Text);
        if(radioButton1.Checked==true)
            gender = radioButton1.Text;
        else if (radioButton2.Checked == true)
            gender = radioButton2.Text;
        else
            gender = radioButton3.Text;
        if  (checkBox1.Checked == true)
            courses += checkBox1.Text;
        if (checkBox2.Checked == true)
            courses += checkBox2.Text;
        if (checkBox3.Checked == true)
            courses += checkBox3.Text;
        MessageBox.Show("Name is:" + name + "\n DOB is:" + dob + "\n Gender
is:" + gender + "\n Mobile No. is:" + mobile + "\n EmailId is:" + email + "\n
Address is:" + address + "\n Course selected is:" + courses);
 }
```

- **Step 10:** Run the Project by using Start Button.
- **Step 11:** We can get output like this.

Output:



Lab-9:

To Design a notepad application to implement menus, custom dialog box and MDI Concepts.

Procedure to Create a MDI Form:

Step 1: First we need to create 4 or more forms in the same project.

Step 2: For that we need to go to the current project => right click on that => then we need to select ADD option => in that we select Add items => select windows forms=> click okbutton.

Step 3: In this way we can create number of forms in the same project.

Step 4: Now we have many forms in that one Project and any one form must be parent form.

Step 5: For example we have 4 forms like Form1, Form2, Form3 and Form and Rename all the forms as Form1 as Notepad, Form2 as NewFile, Form3 as Open and

Form4 as Save.

Step 6: I would like to take Form1 (Notepad) as parent form for this

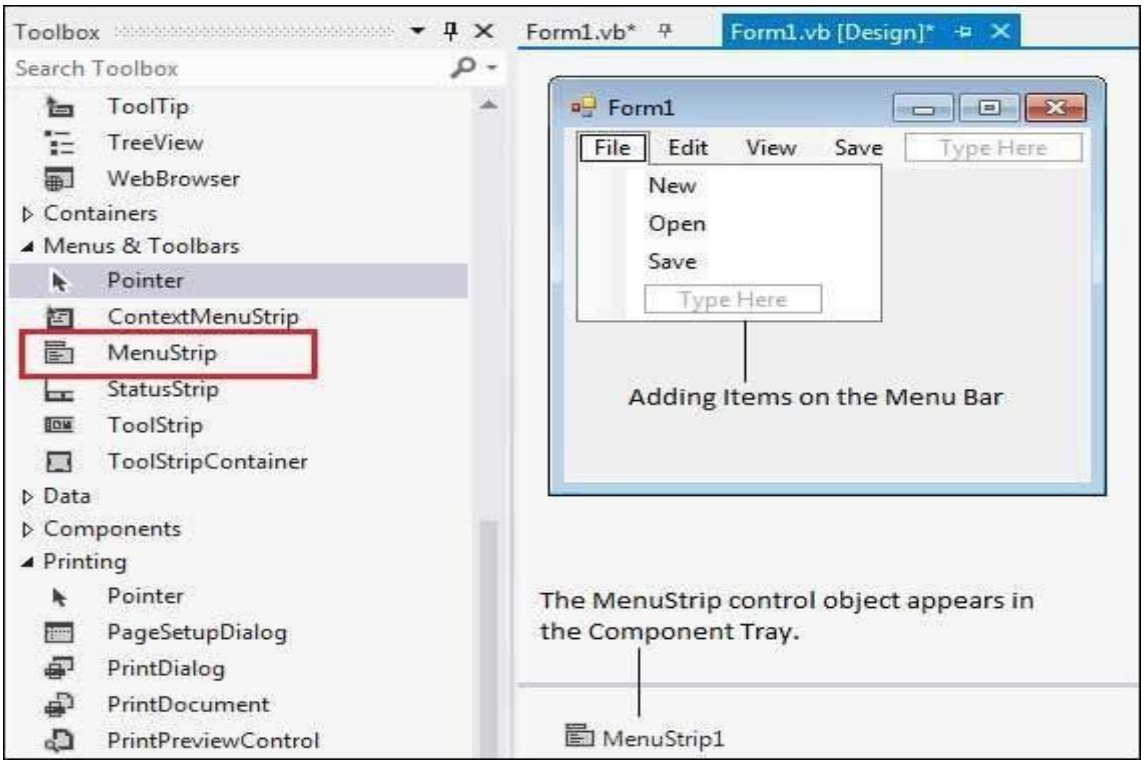
First select Form1 => then go to the properties windows => in that select "**is MdiContainer".**

By default each form property is **false** but now it will becomes to **"true".**

Step 7: Then it (Form1/ Notepad) is a parent form in my project.

Step 8: Again we create 4 menus in Form1(Notepad).

Step 9: For that we need to take 4 **"menu strip"** tools from toolbox and drop it on Form1 Rename like 1st Menu as "File", 2nd Menu as "Edit", 3rd Menu as View and 4th Menu as "Format".



Step 10: Then we created a 4 menus in Form1.

Step 11: Now select Form2 / NewFile in **menu** and double click on that , then it will open code windowForm2 code will be:

```
NewFile  F2=new NewFile();
```

```
F2. MdiParent = this;
F2.Show()
```

Step 12: Now select Form3 / Open in **menu** and double click on that, then it will open code will be:

```
Open  F3=new Open();
F3. MdiParent = this;
F3.Show()
```

Step 13: Now select Form4 / Save in **menu** and double click on that, then it will open code will be:

```
Save  F4=new  Save();
F4. MdiParent = this;
F4.Show()
```

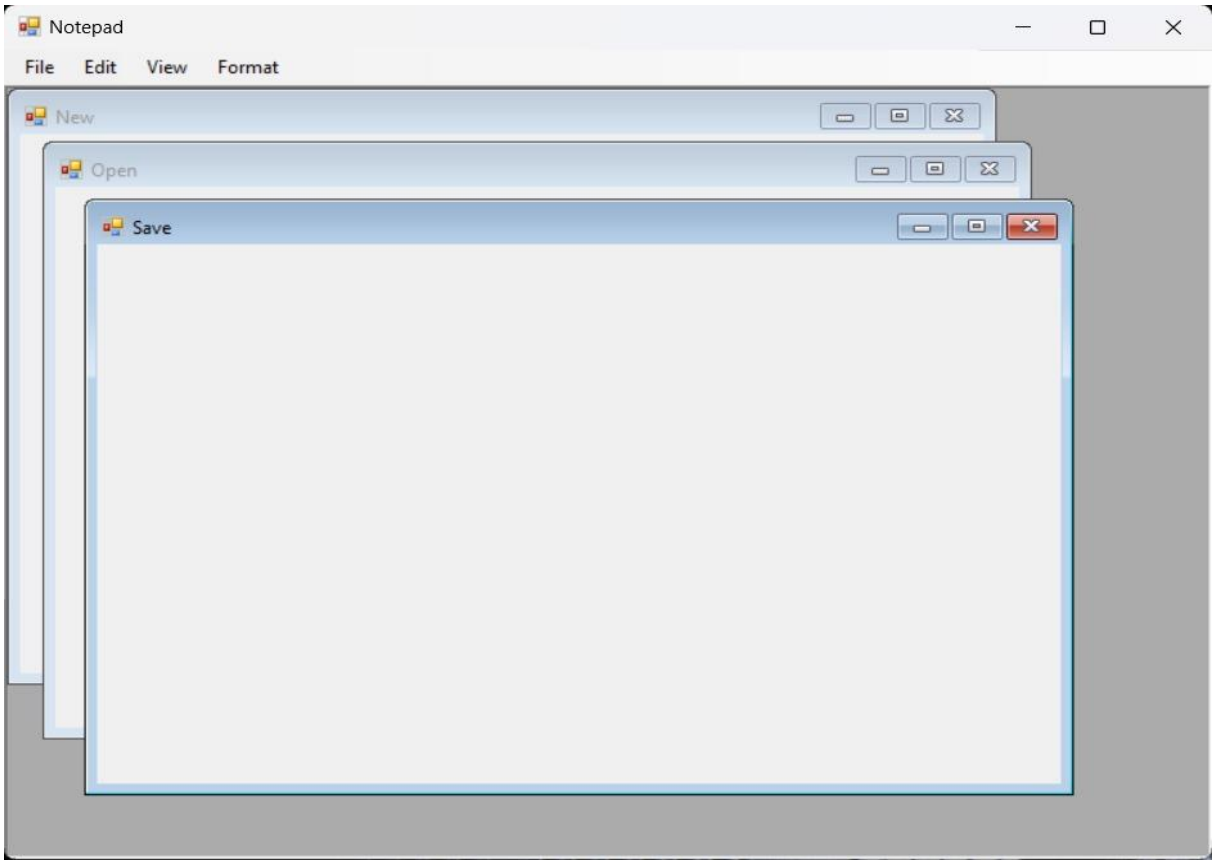
Step 14: Then code is ready.

Step 15: Run the code by using "start" button in the menu bar.

Step 16: It will open Form1/ Notepad window in that 4 menus will be framed Out of all those 4 menus in 1st menu that is "New File", "Open File", "Save".

Step 17: This is the way we can create multiple instances of single form.

Output:



Lab-10:

Develop a Windows application with database for Student Information System [Insert, Update and Delete Commands].

Step 1: First User Should Create a data base for that

Go to **"Solution Explorer"** widow => right click on **the project name** => click on **"Add"** option =>Select **"add new item"** => select **"service Based database"** => click on **"Add".**

Now data base will be added to our project.

Step 2: If we wants to rename database => right click on the project => using rename option we canrename the database.

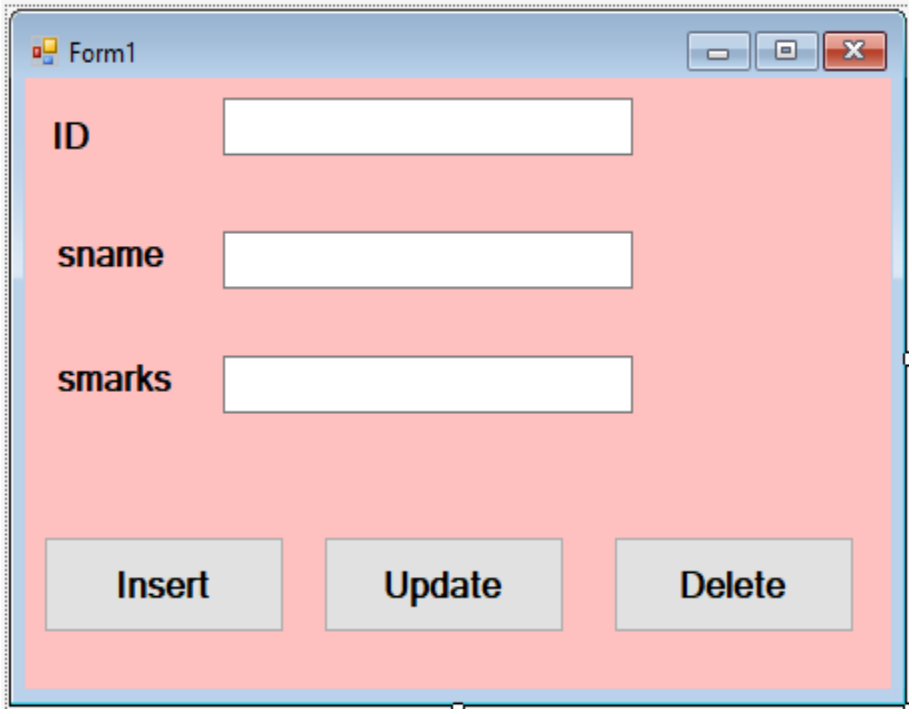
Step 3: now we need to create a table in the database. For this => double click on the data base => will open **"server explorer"** window => select **"table"** option => right click on this => **select "add new table"** => new table will be added to the database.

Step 4: If we want to change the table name then we can change as **"dbo.Student"** and will add somecolumns into that table like **"ID, SName and Smarks** etc."

Step 5: After entering these details in to table then we need to Update the table by using **"update"** => then click on **"update database".**

Step 6: We need to design the Front end application like C#.net or VB.net etc.

Step 7: we need to create one form with 3 labels, 3 textboxes and 3 button controls for entering data.



Step 8: We need to create / Establish Connection between Front end Form and Back end Database for that write the coding part.

Step 9: Double click on the form and write the coding

```
Imports  System.Data.SqlClient
Public Class Form1
Dim cn As SqlConnection
Dim cmd As SqlCommand
Private Sub Form1_Load(sender As Object, e As EventArgs) HandleMyBase.Load
cn = New SqlConnection("Connection String from properties ")
cn.Open()
        MsgBox("connected successfully")
cn.Close()
End Sub
```

Step 10: Run the Project then it display **"Connected successfully".**

Step 11: Now we insert tuples or rows in to the table.

Step 12: For inserting data in to the Table through VB.NET Coding

Double click on the **Insert button** and write the coding

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        cn.Open()
        cmd = cn.CreateCommand()
        cmd.CommandType = CommandType.Text
        cmd.CommandText = "insert into Student values(" & TextBox1.Text & " , ' " &
                TextBox2.Text &" ' , " & TextBox3.Text & ")"
        cmd.ExecuteNonQuery()
        MsgBox("record inserted successfully")
        cn.Close()
End Sub
```

Step 13: Run the Project then it display "Connected Successfully" and insert some rows in table.

Step 14: Run the query as **select * from Student;**

Step 14: Now we Update tuple or rows in table.

Step 15: For Updating Existing data in the Table through VB.NET Coding

Double click on the **Update button** and write the coding

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        cn.Open()
        cmd = cn.CreateCommand()
        cmd.CommandType = CommandType.Text
        cmd.CommandText = "update Student set sname=' " & TextBox2.Text
          & " ' , smarks=" &TextBox3.Text & " where id=" & TextBox1.Text &
          """
        cmd.ExecuteNonQuery()
        MsgBox("record updated successfully")
        cn.Close()
End Sub
```

Step 16: Run the Project then it display "Connected Successfully" and update rows.

Step 17: Now we delete data from table

Step 18: For deleting data from the Table through VB.NET Coding

Double click on the **Delete button** and write the coding

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        cn.Open()
        cmd =cn.CreateCommand()
        cmd.CommandType =CommandType.Text
        cmd.CommandText = "delete from Student where id= "  & TextBox1.Text
        & " "
```

```
        cmd.ExecuteNonQuery()
        MsgBox("record deleted successfully")
        cn.Close()
End Sub
```

****