Name :- Praful Kailas Patil

Roll :- 23110115

Div :- B

Sub :- Lab on C# Programming

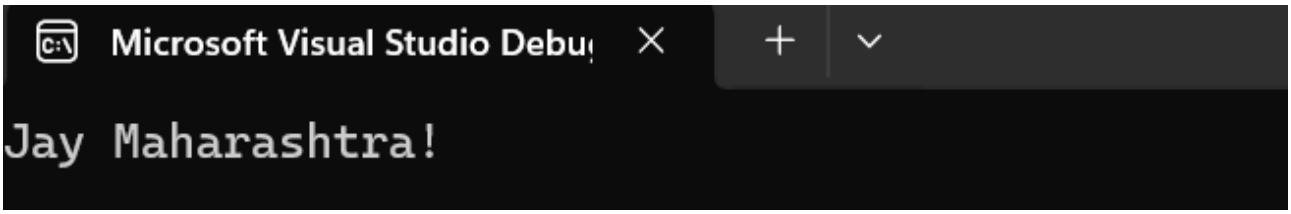
---

1. **Write a Console Application to demonstrate the structure of C# Programming.**

   - **Program :-**

```
using System;
namespace Program
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Jay Maharashtra!");
            Console.Read();
        }
    }
}
```

   - **Output :-**

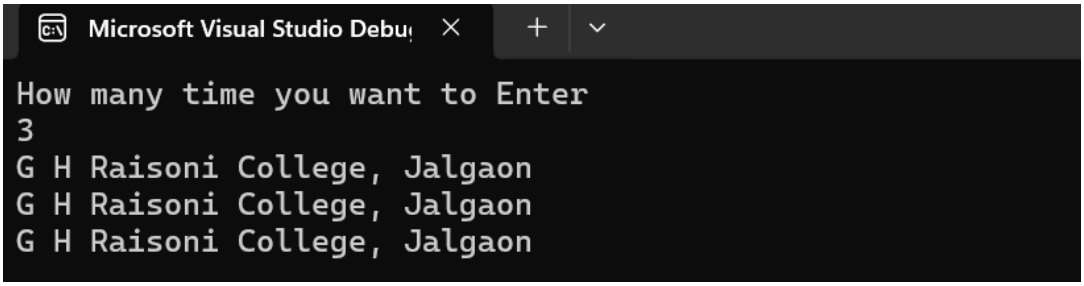


2. Write a program to print "G H Raisoni College, Jalgaon" given number of times.

- **Program :-**

```
int n;
Console.WriteLine("How many time you want to Enter");
n = int.Parse(Console.ReadLine());
for(int i = 1; i <= n; i++)
   Console.WriteLine("G H Raisoni College, Jalgaon");
```

- **Output :-**

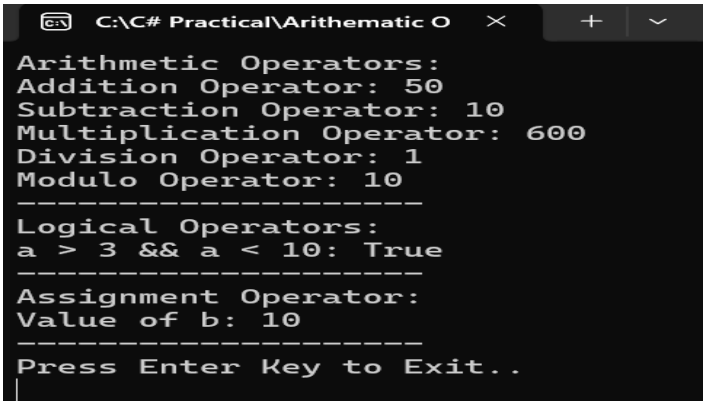


3. Write a program to show use of different operators.

- **Program :-**

```
using System;
namespace Program
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Arithmetic Operators:");
         int result;
         int x = 30, y = 20;
         result = x + y;
         Console.WriteLine("Addition Operator: " + result);
         result = x - y;
         Console.WriteLine("Subtraction Operator: " + result);
         result = x * y;
         Console.WriteLine("Multiplication Operator: " + result);
         result = x / y;
         Console.WriteLine("Division Operator: " + result);
         result = x % y;
         Console.WriteLine("Modulo Operator: " + result);
         Console.WriteLine("--------------------");

         Console.WriteLine("Logical Operators:");
         int a = 6;
         Console.WriteLine("a > 3 && a < 10: " + (a > 3 && a < 10));

         Console.WriteLine("--------------------");

         Console.WriteLine("Assignment Operator:");
         int b = 10;
         Console.WriteLine("Value of b: " + b);

         Console.WriteLine("--------------------");
         Console.WriteLine("Press Enter Key to Exit..");
         Console.ReadLine();
      }
   }
}
```

- **Output :-**

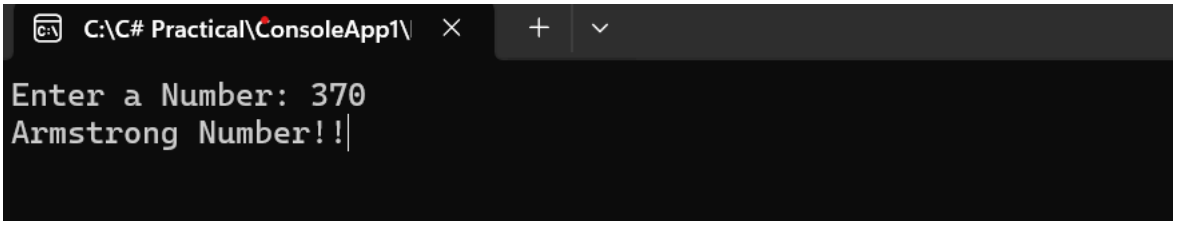


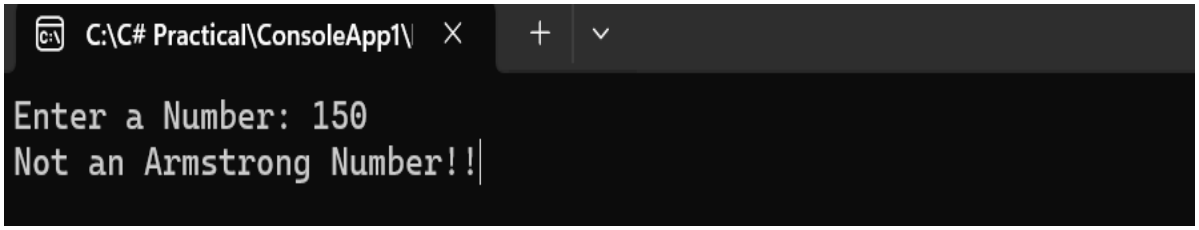
4. Write a program to reverse a string and check if it is Armstrong number.

- **Program :-**

```
using System;

namespace ConsoleApp2
{
   class Program
   {
      static void Main(string[] args)
      {
         int num, x, sum = 0, y;
         Console.Write("Enter a Number: ");
         num = int.Parse(Console.ReadLine()); y = num;
         while (num > 0)
         {
            x = num % 10;
            sum = sum + (x * x * x); num = num / 10;
         }
         if (y == sum)
            Console.Write("Armstrong Number!!");
         else
            Console.Write("Not an Armstrong Number!!");
            Console.ReadLine();
      }
   }
}
```

- **Output :-**



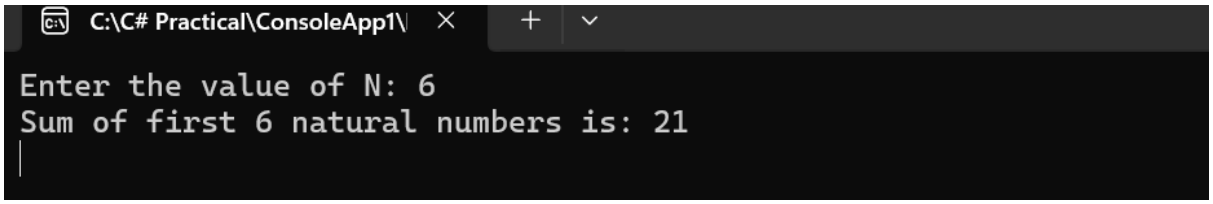


5. Write a program sum of first's N natural numbers using for loop.

- **Program :-**

```
using System;

namespace SumOfNaturalNumbers
{
    class SumCalculator
    {
        public int N;

        public int CalculateSum()
        {
            int sum = 0;
            for (int i = 1; i <= N; i++)
            {
                sum += i;
            }
            return sum;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            SumCalculator calculator = new SumCalculator();
            Console.Write("Enter the value of N: ");
            calculator.N = Convert.ToInt32(Console.ReadLine());




            int result = calculator.CalculateSum();
            Console.WriteLine("Sum of first {0} natural numbers is: {1}", calculator.N,
            result);

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

- **Output :-**



OR

5. Write a C#.NET program to display the multiplication table of a number entered by the user. Example: Output: 5 x 1 = 5

- **Program :-**

```
using System;

class Program
{
   static void Main()
   {
      Console.Write("Enter a number to show its multiplication table:"); int number =
int.Parse(Console.ReadLine());
      for (int i = 1; i <= 10; i++)
      {
         int result = number * i;
         Console.WriteLine(number + "x" + i + " =" + result); Console.ReadLine();
      }
   }
}
```

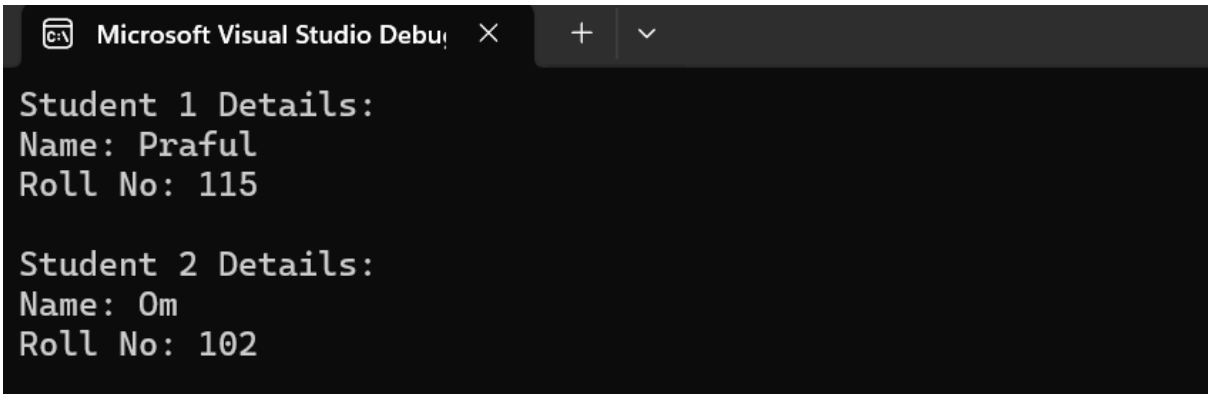
- **Output :-**

6. Write a program to show use of Constructor.

- **Program :-**

```
using System;
class Student
{
   string name; int rollNo;
   public Student(string n, int r)
   {
      name = n; rollNo = r;
   }
   public void Display()
   {
      Console.WriteLine("Name: " + name);
      Console.WriteLine("Roll No: " + rollNo);
   }
}
class Program
{
   static void Main(string[] args)
   {
      Student s1 = new Student("Praful", 115);
      Student s2 = new Student("Om", 102);
      Console.WriteLine("Student 1 Details:"); s1.Display();

      Console.WriteLine("\nStudent 2 Details:");
      s2.Display();
   }
}
```

- **Output :-**

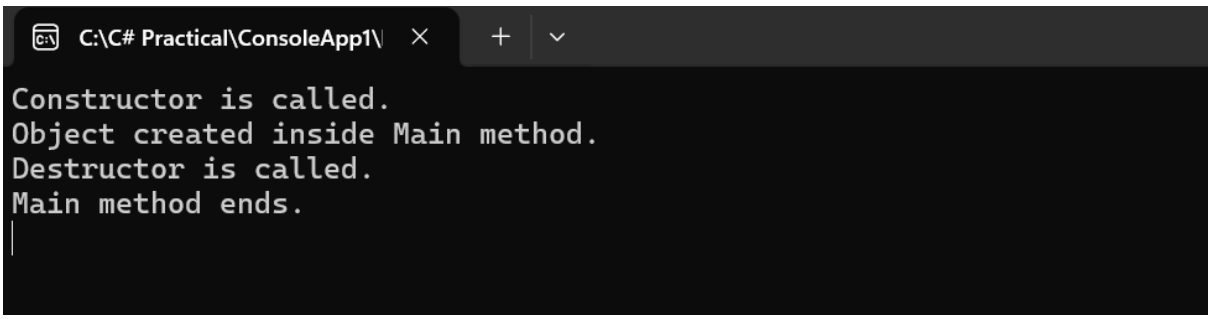


7. Write a program to show use of Destructor.

- **Program :-**

```
using System;

class Sample
{
    public Sample()
    {
        Console.WriteLine("Constructor is called.");
    }

    ~Sample()
    {
        Console.WriteLine("Destructor is called.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Sample obj = new Sample();
        Console.WriteLine("Object created inside Main method.");

        obj = null;

        GC.Collect();
        GC.WaitForPendingFinalizers();

        System.Threading.Thread.Sleep(100);
        Console.WriteLine("Main method ends.");
        Console.ReadLine();
    }
}
```

- **Output :-**

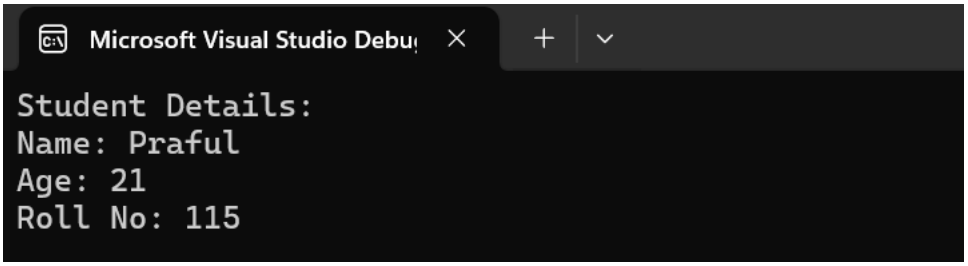


8. Write a program to demonstrate Inheritance.

- **Program :-**

```
using System;
class Person
{
   public string Name; public int Age;
   public void ShowDetails()
   {
      Console.WriteLine("Name: " + Name); Console.WriteLine("Age: " + Age);
   }
}
class Student : Person
{
   public int RollNo;
   public void ShowStudentDetails()
   {
      ShowDetails();
      Console.WriteLine("Roll No: " + RollNo);
   }
}
class Program
{
   static void Main(string[] args)
   {
      Student s1 = new Student();

      s1.Name = "Praful"; s1.Age = 21;
      s1.RollNo = 115;
      Console.WriteLine("Student Details:"); s1.ShowStudentDetails();
   }
}
```

- **Output :-**

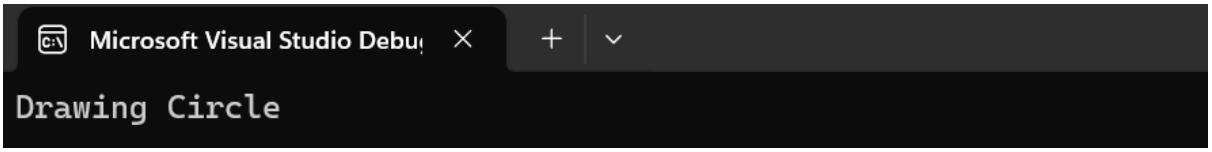


9. Write a program to demonstrate Interface in C#.

- **Program :-**

```
using System;
interface IShape
{
   void Draw();
}
class Circle : IShape
{
   public void Draw() => Console.WriteLine("Drawing Circle");
}
class Program
{
   static void Main()
   {
      IShape shape = new Circle(); shape.Draw();
   }
}
```

- **Output :-**

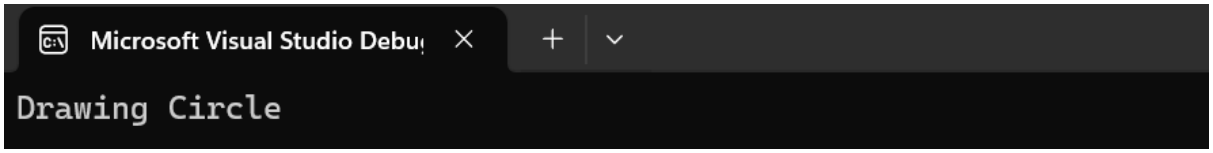


10. Write a Console Application to Demonstrate Abstract Class in C#.

➢ Program :-

```
using System;
abstract class Shape
{
   public abstract void Draw();
}
class Circle : Shape
{
   public override void Draw() => Console.WriteLine("Drawing Circle");
}
class Program
{
   static void Main()
   {
      Shape s = new Circle(); s.Draw();
   }
}
```

➢ Output :-

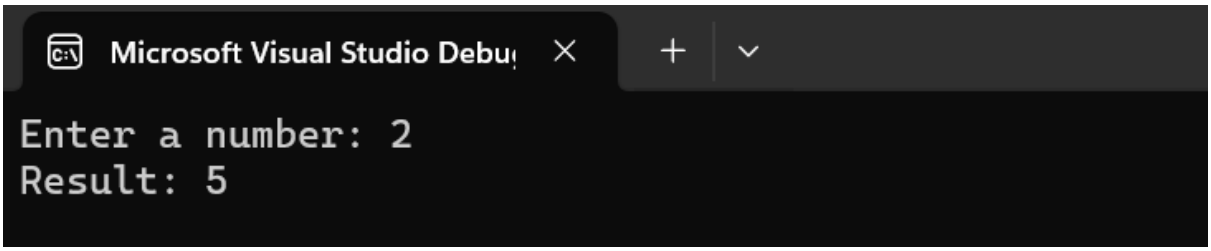


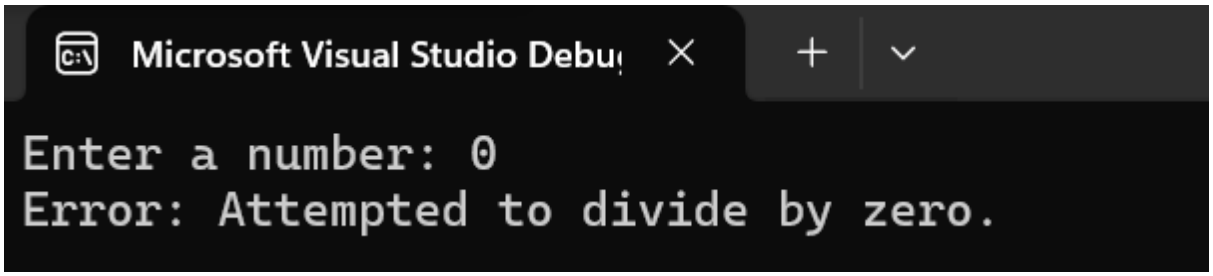
11. Write a Console Application to demonstrate the Exception Handling Mechanism in C#.

- **Program :-**

```
using System;

class Program
{
   static void Main()
   {
      try
      {
         Console.Write("Enter a number: ");
         int num = Convert.ToInt32(Console.ReadLine());
         Console.WriteLine("Result: " + (10 / num));
      }
      catch (Exception e)
      {
         Console.WriteLine("Error: " + e.Message);
      }
   }
}
```

- **Output :-**



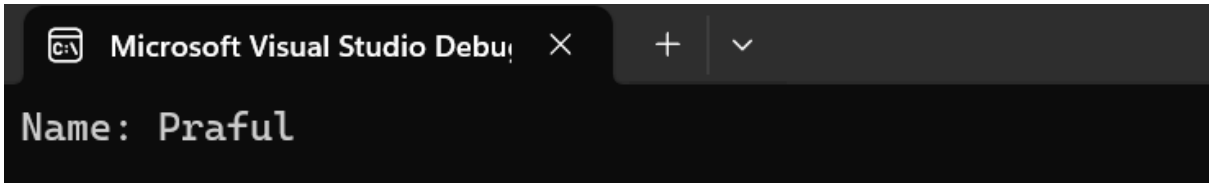


12. Write a Console Application to demonstrate the Class and object in C#.

- **Program :-**

```
using System;

class Student
{
    public string Name;
    public void Display() => Console.WriteLine("Name: " + Name);
}

class Program
{
    static void Main()
    {
        Student s = new Student(); s.Name = "Praful"; s.Display();
    }
}
```

- **Output :-**

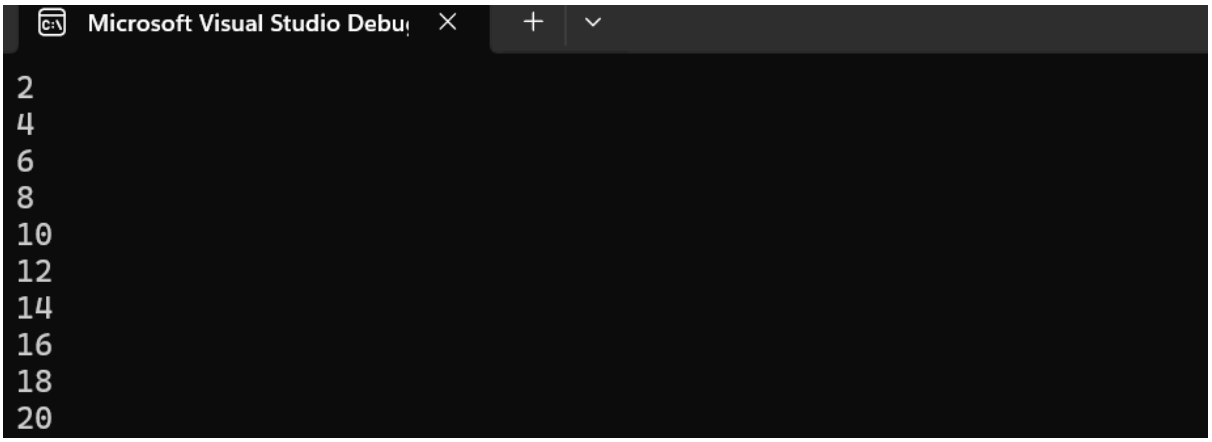


13. Write a Console Application to demonstrate the Array in C#.

- **Program :- 1D Array -**

```
using System;

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

      {
         int[] a = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };
         for (int i = 0; i < a.Length; i++)
         {
            Console.WriteLine(a[i]);
         }
         Console.Read();
      }
   }
}
```

- **Output :-**

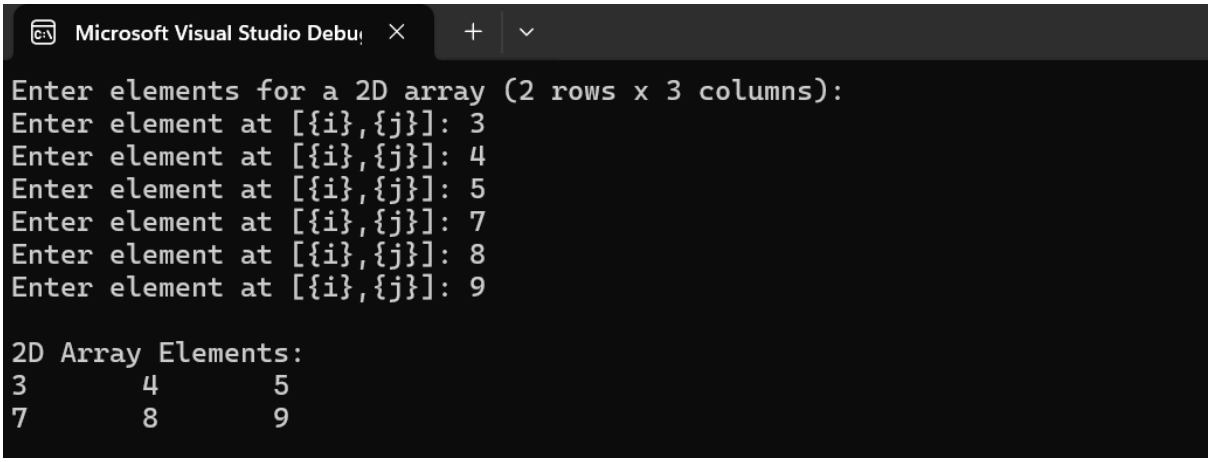


- **Program :- 2D Array –**

```
using System;
class Program
{
   static void Main()
   {
      int rows = 2, cols = 3;

      int[,] numbers = new int[rows, cols];

      Console.WriteLine("Enter elements for a 2D array (2 rows x 3 columns):");

      for (int i = 0; i < rows; i++)
      {
```

```
            for (int j = 0; j < cols; j++)
            {
                Console.Write("Enter element at [{i},{j}]: "); numbers[i, j] =
Convert.ToInt32(Console.ReadLine());
            }
        }

        Console.WriteLine("\n2D Array Elements:");
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                Console.Write(numbers[i, j] + "\t");
            }
            Console.WriteLine();
        }

        Console.ReadLine();
    }
}
```

- **Output :-**



14. Create a student registration form using Label, Button, and Textbox control and print the data on the same form

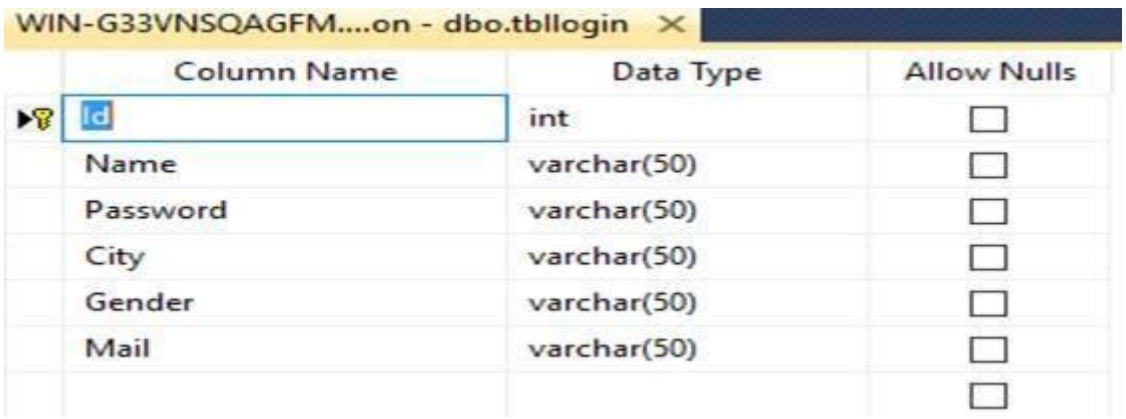
- **Program :-**

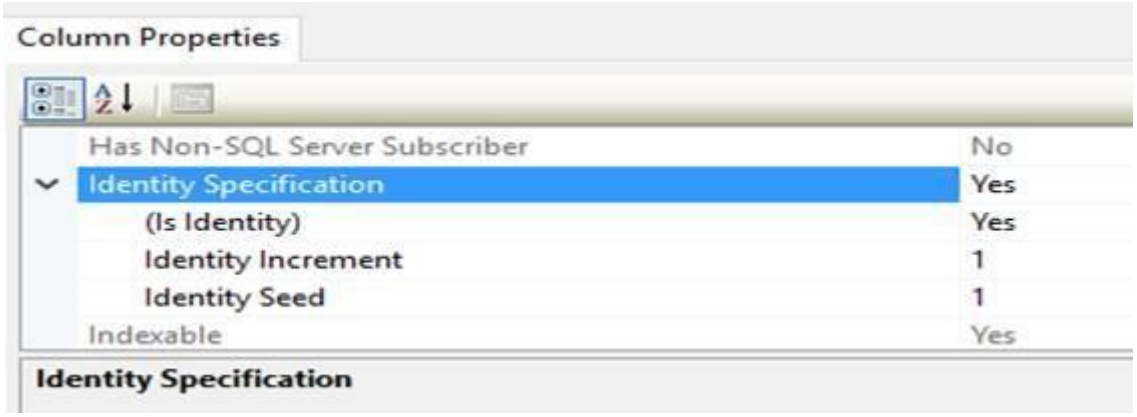
```
using System;
using System.Windows.Forms;
public class StudentForm : Form
{
   Label lblName, lblRoll;
   TextBox txtName, txtRoll;
   Button btnSubmit;
   Label lblOutput;
   public StudentForm()
   {
      this.Text = "Student Registration Form";
      this.Width = 400;
      this.Height = 300;
      lblName = new Label() { Text = "Name:", Top = 30, Left = 30 };
      txtName = new TextBox() { Top = 30, Left = 120 };
      lblRoll = new Label() { Text = "Roll No:", Top = 70, Left = 30 };
      txtRoll = new TextBox() { Top = 70, Left = 120 };
      btnSubmit = new Button() { Text = "Register", Top = 110, Left = 120 };
      btnSubmit.Click += new EventHandler(this.Submit_Click);
      lblOutput = new Label() { Top = 160, Left = 30, Width = 300 };
      this.Controls.Add(lblName);
      this.Controls.Add(txtName);
      this.Controls.Add(lblRoll);
      this.Controls.Add(txtRoll);
      this.Controls.Add(btnSubmit);
```

```
    this.Controls.Add(lblOutput);
}
private void Submit_Click(object sender, EventArgs e)
{
  lblOutput.Text = $"Student: {txtName.Text}, Roll No:
{txtRoll.Text}"; public static void Main()
{
    Application.Run(new StudentForm());
```

- **Output :-**

Student: Praful, Roll No: 115





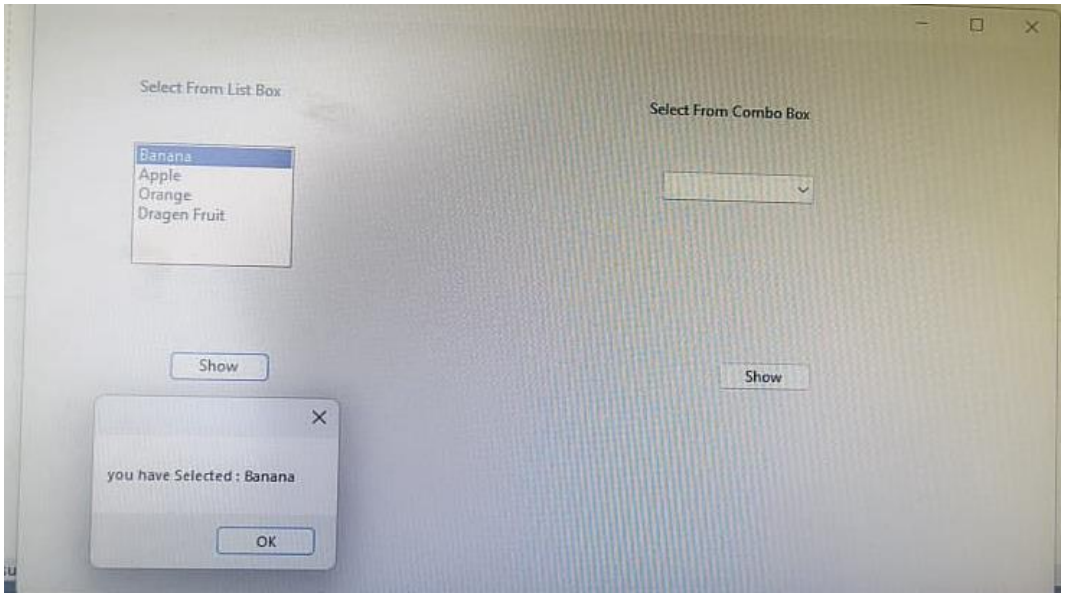

```
protected void Page_Load(object sender, EventArgs e)
{
}
0 references
protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(@"Data Source=[redacted];Initial Catalog=AspApplication;User ID =sa;Password=[redacted]");
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into tbllogin values(@name,@Pass,@city,@gender,@mail)", con);
        cmd.Parameters.AddWithValue("name", TextBox1.Text);
        cmd.Parameters.AddWithValue("Pass", TextBox2.Text);
        cmd.Parameters.AddWithValue("city", DropDownList1.SelectedValue);
        cmd.Parameters.AddWithValue("gender", RadioButtonList1.SelectedValue);
        cmd.Parameters.AddWithValue("mail", TextBox4.Text);
        cmd.ExecuteNonQuery();

        TextBox1.Text = "";
        TextBox2.Text = "";
        DropDownList1.SelectedValue = "";
        RadioButtonList1.SelectedValue = "";
        TextBox4.Text = "";
        TextBox1.Focus();
    }
}
}
```

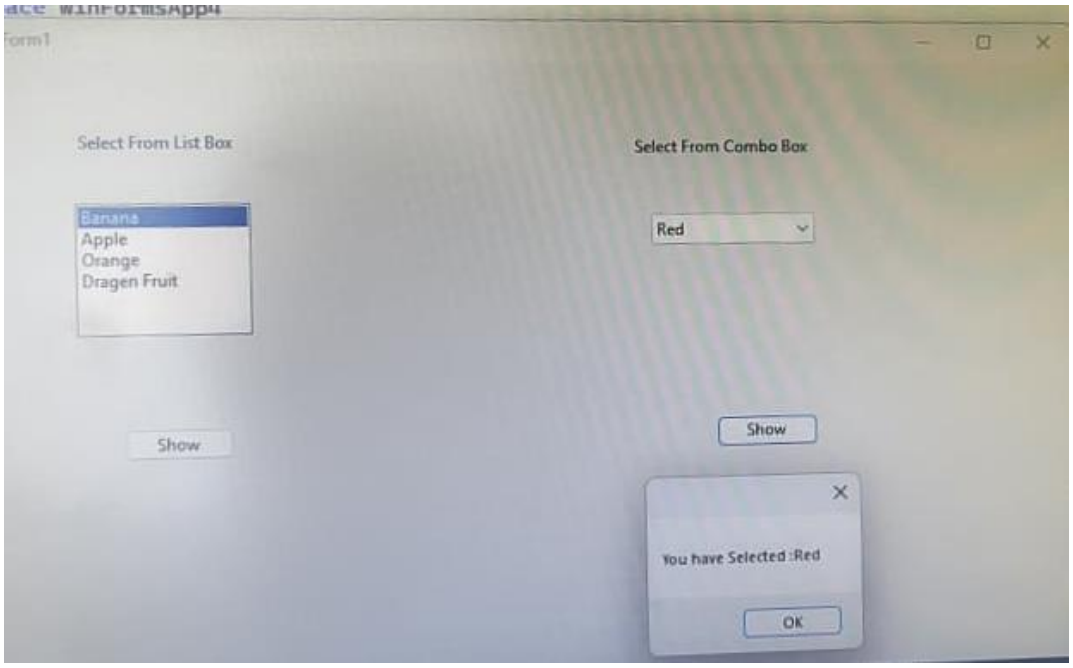
15. Create a window application to demonstrate List Box and Combo Box control.

- **Program :-**

```
namespace WinFormsApp4 {

public partial class Form1 : Form
   {
   public Form1()
   {
   InitializeComponent();
   }

private void button1_Click(object sender, EventArgs e)
      {

        foreach (object obj in listBox1.SelectedItems)
        {
             MessageBox.Show("you have Selected : " + obj.ToString());
         }

   }

      private void button2_Click(object sender, EventArgs e)
      {
         string var = comboBox1.Text; MessageBox.Show("You have Selected:" + var);
      }
      private void Form1_Load(object sender, EventArgs e)
      {

       }

}

}
```

- **Output :-**


Select From List Box
Banana
Apple
Orange
Dragen Fruit
Show
Select From Combo Box
Show
you have Selected : Banana
OK


Select From List Box
Banana
Apple
Orange
Dragen Fruit
Show
Select From Combo Box
Red
Show
You have Selected :Red
OK

16. Write a window application to change the background color randomly after every Second (Use Timer Control).

- **Program :-**

```
using System;
using System.Drawing;
using System.Windows.Forms;
public class RandomColorForm : Form
{
private Timer colorTimer;
private Random random;
public RandomColorForm()
{
InitializeComponent();
}
private void InitializeComponent()
{
this.Text = "Random Color Changer";
this.Size = new Size(400, 300);
this.StartPosition = FormStartPosition.CenterScreen; random
= new Random();
colorTimer = new Timer();
colorTimer.Interval = 1000; // 1000 milliseconds = 1 second
colorTimer.Tick += ColorTimer_Tick; // Attach the event handler
colorTimer.Start(); // Start the timer
}
private void ColorTimer_Tick(object sender, EventArgs e)
{
int r = random.Next(256); // 0-255
```

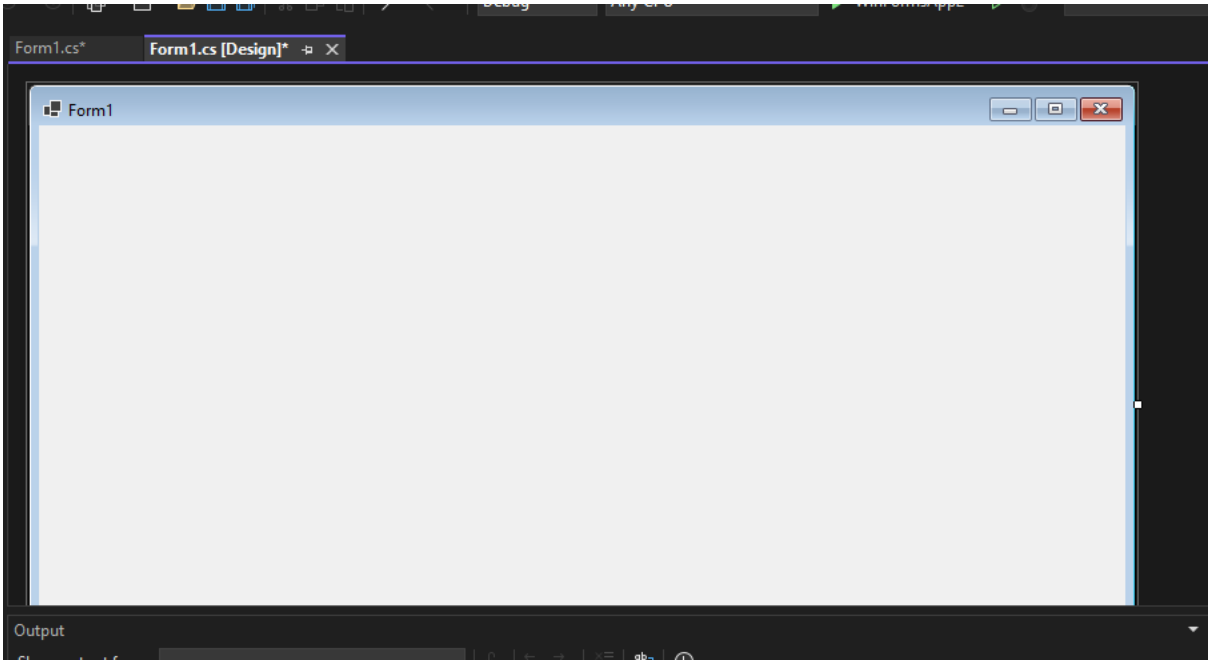
```
int g = random.Next(256); int b =
random.Next(256);
this.BackColor = Color.FromArgb(r, g, b)
[STAThread]
public static void Main()
{
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(new RandomColorForm());
}
}
```

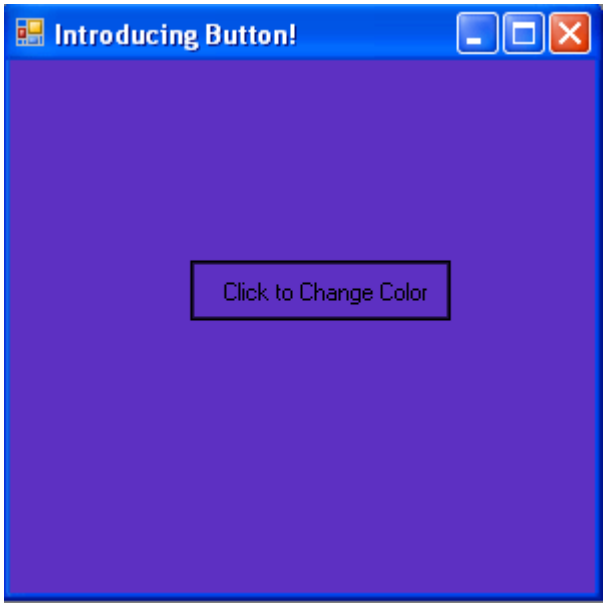
- **Output :-**

  When you run the program, the **form window background will change its color randomly every 1 second**.

  Example:

  - At 1s → Blue
  - At 2s → Green
  - At 3s → Orange
  - At 4s → Pink


Form1.cs*
Form1.cs [Design]*
Form1
Output


Introducing Button!
Click to Change Color

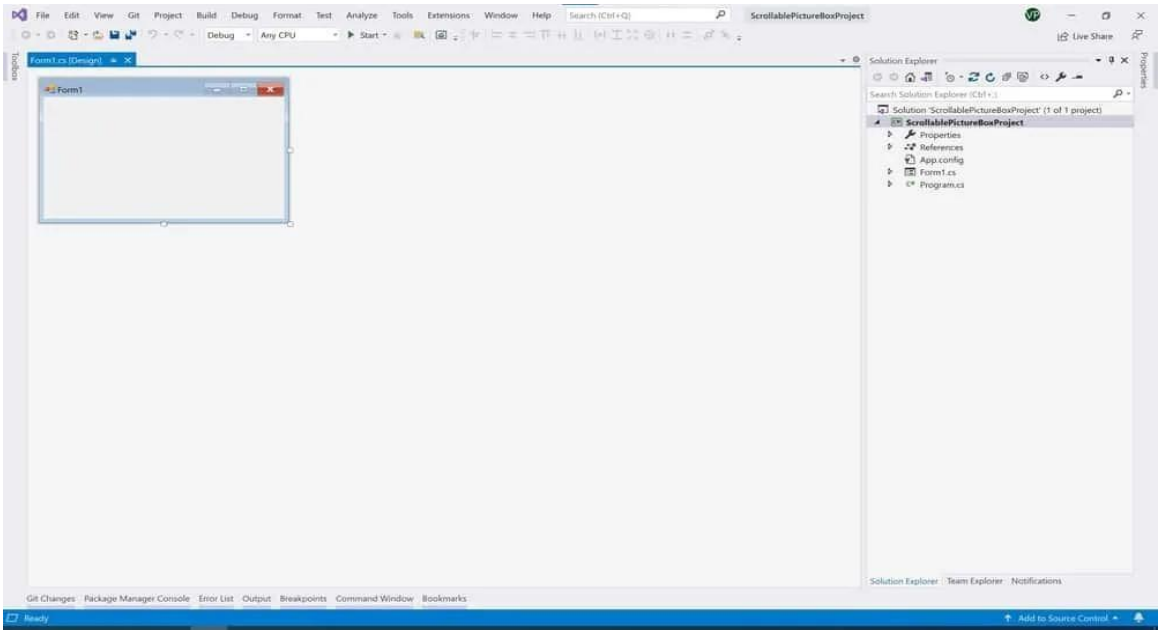
17. Create a C# application using Picture Box, Scrollbar control.

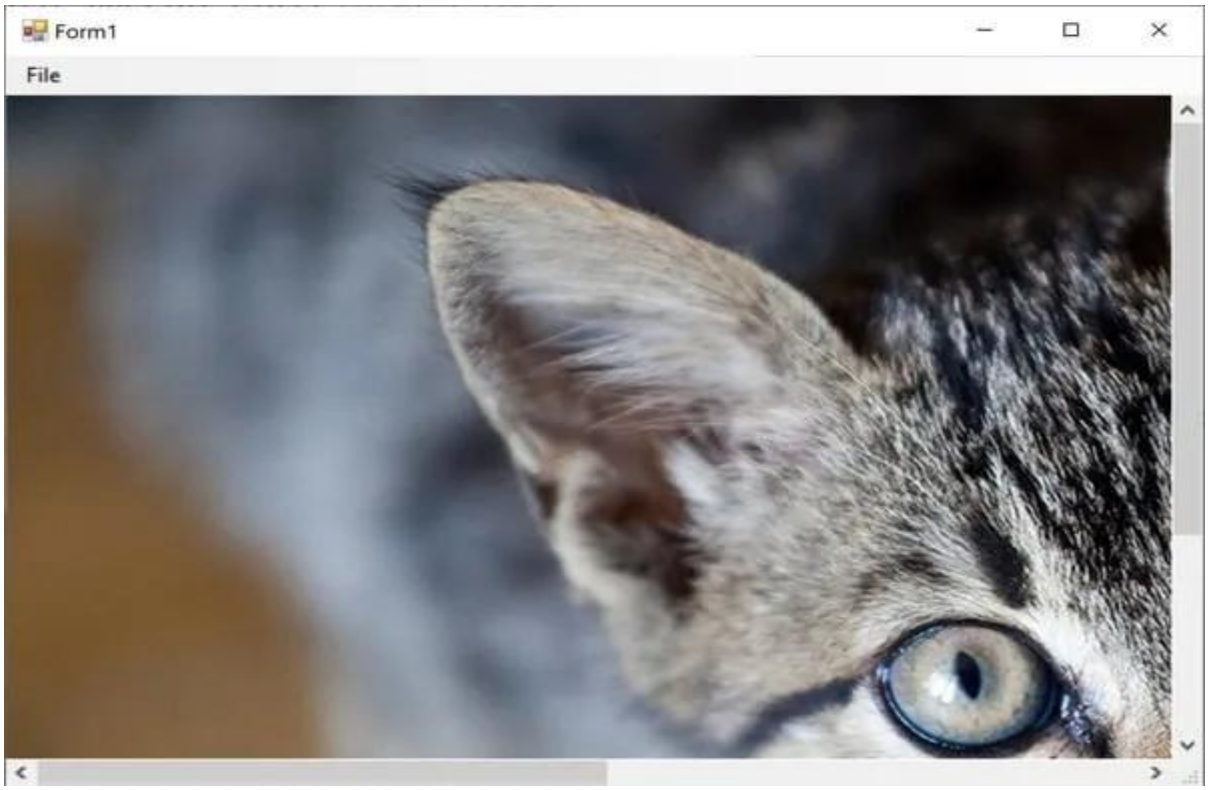
- **Program :-**

```
using System;
using System.Drawing;
using System.Windows.Forms;
namespace ScrollablePictureBoxApp
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
private void btnLoadImage_Click(object sender, EventArgs e)
{
ofdImage.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.gif;*.bmp";
ofdImage.Title = "Select an Image File";
if (ofdImage.ShowDialog() == DialogResult.OK)
{
try
{
pbDisplayImage.Image = Image.FromFile(ofdImage.FileName); pbDisplayImage.Size
= pbDisplayImage.Image.Size;
}
catch (Exception ex)
{
MessageBox.Show("Error loading image: " + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

```
    }
    }
    }
```

- **Output :-**


File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help
ScrollablePictureBoxProject
Debug
Any CPU
Start
Live Share
Form1.cs [Design]
Form1
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ScrollablePictureBoxProject' (1 of 1 project)
ScrollablePictureBoxProject
Properties
References
App.config
Form1.cs
Program.cs
Solution Explorer
Team Explorer
Notifications
Git Changes
Package Manager Console
Error List
Output
Breakpoints
Command Window
Bookmarks
Ready
Add to Source Control


Form1
File

18. Demonstrate the use of Timer control in C# Display the current Date and Time

- **Program :-**

```
using System;

namespace WinFormsApp13
{
   public partial class Forml : Form
   {
      public Form1()
      {
         InitializeComponent();

         timerl.Start();
      }
      private void timerl_Tick(object sender, EventArgs e)
      {
         label1.Text = DateTime.Now.ToString("dd-MM-yyyy hh:mm:ss tt");
      }
   }
}
```

- **Output :-**


Form1
20-09-2025 11:48:36 AM

19. Write a Window Application to demonstrate MDI (Multiple Document Interface)

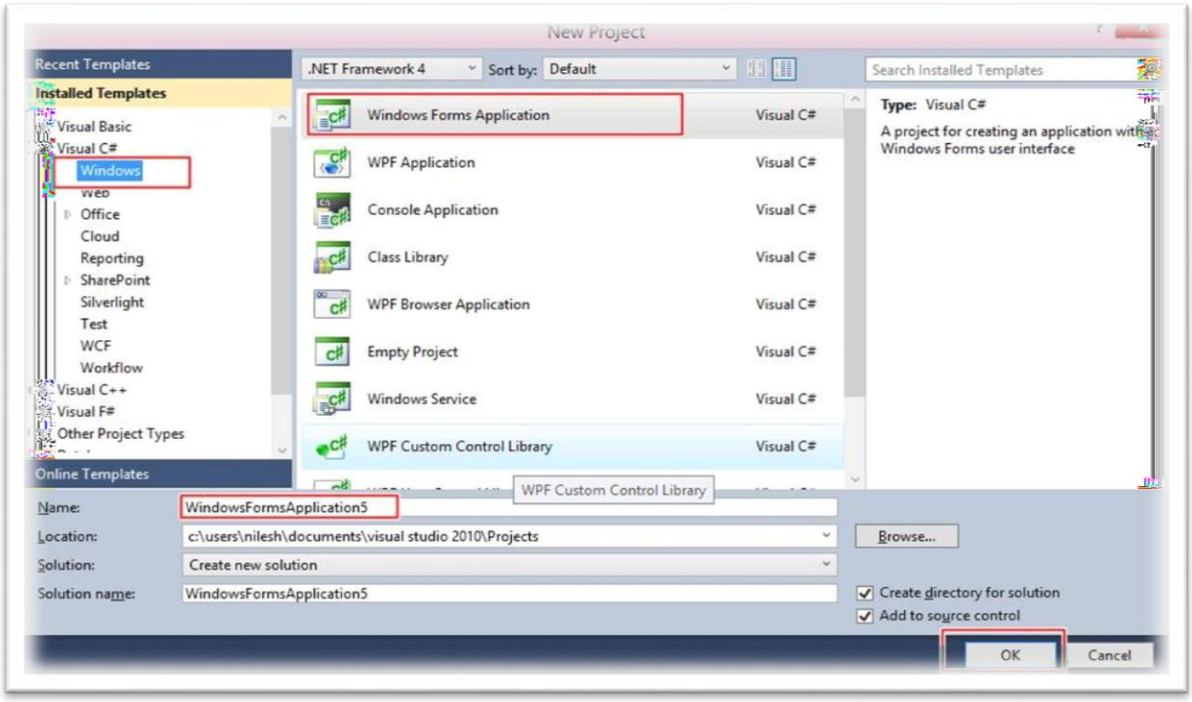
Create Main form with Menu-Form1, Form2 and Form3, and open respective form whenever user clicks on respective Menu option

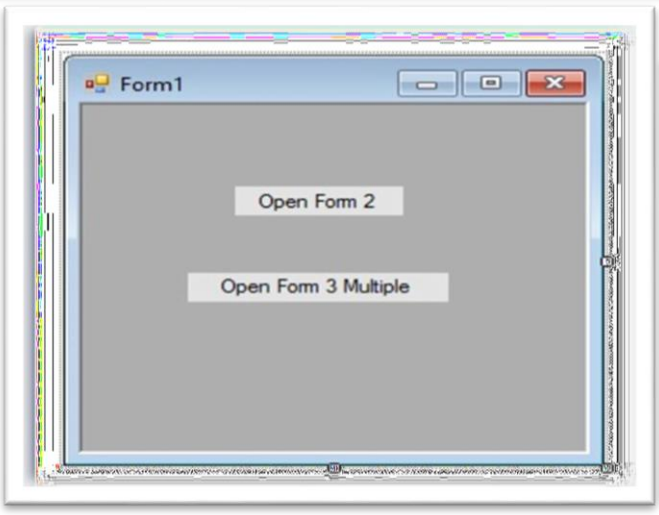
- **Program :-**

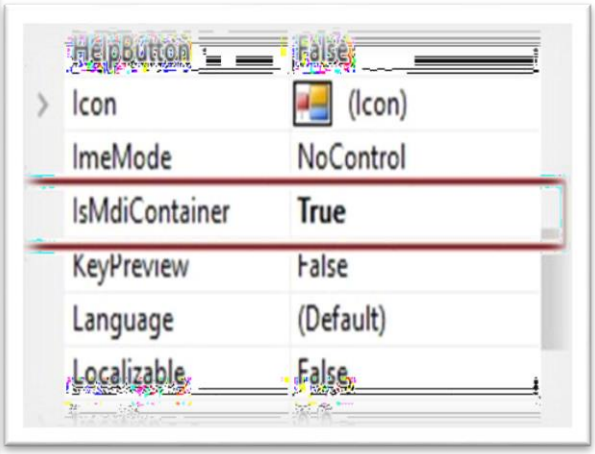
```
using System;

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Form2 frm2;

private void button2_Click(object sender, EventArgs e)

{

if (frm2==null)

frm2 = new Form2(); frm2.MdiParent = this;
frm2.Show();

}

else
```

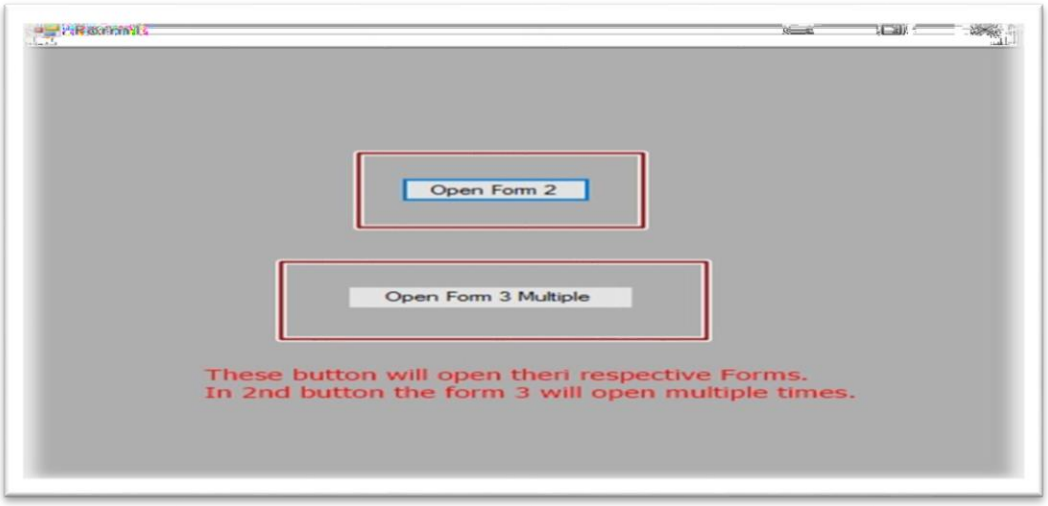
```
{

frm2.Activate();

}

}
private void button3_Click(object sender, EventArgs
e) for (int i = 0; i < 50; i++)
{

Form3 frm3 = new Form3();
frm3.MdiParent = this;
frm3.Show();
}
```

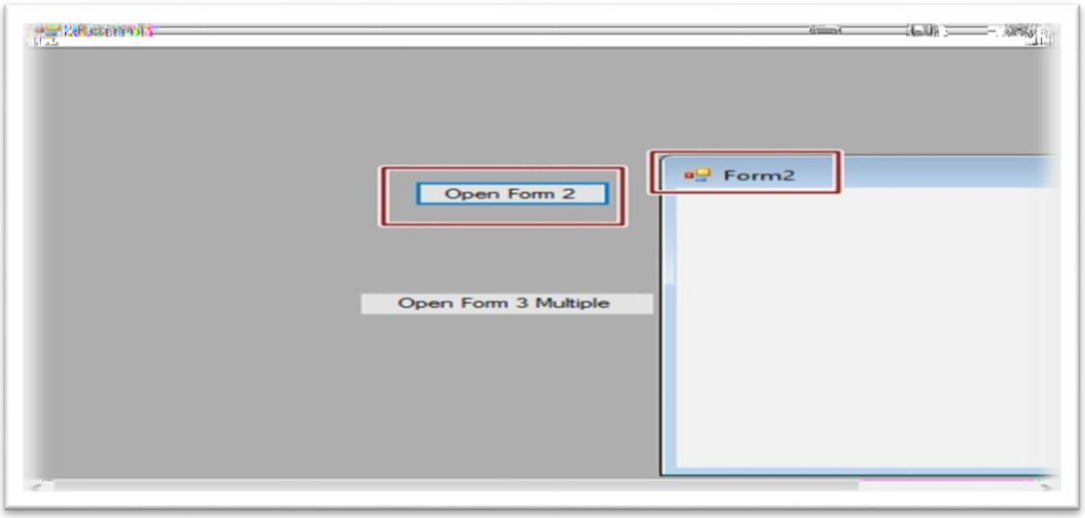
- **Output :-**

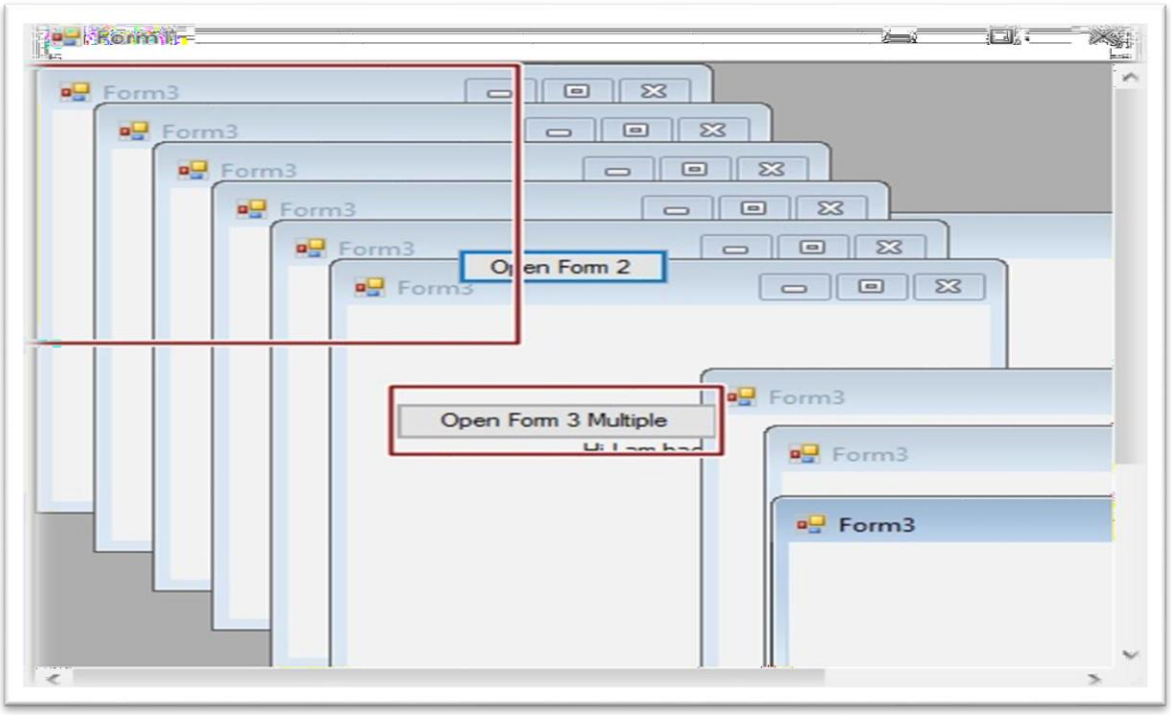







Open Form 2
Open Form 3 Multiple
These button will open theri respective Forms.
In 2nd button the form 3 will open multiple times.


Open Form 2
Form2
Open Form 3 Multiple

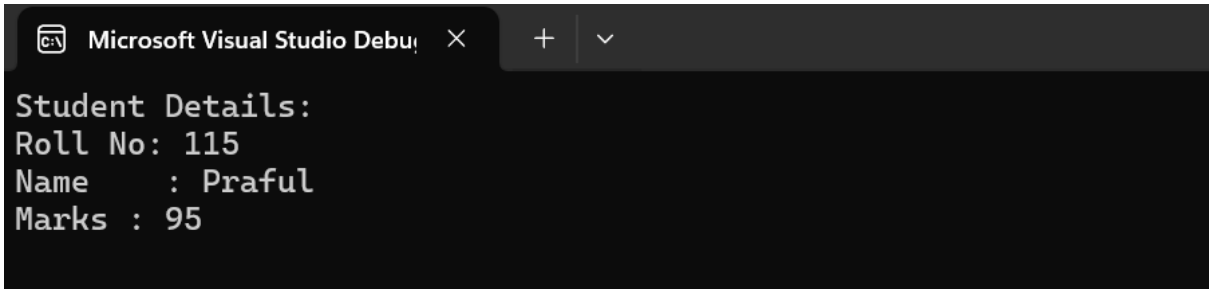

Form3
Form3
Form3
Form3
Form3
Form3
Open Form 2
Open Form 3 Multiple
Form3
Form3
Form3

20. Create a class Student with properties like Roll no, Name, and Marks. Write a method to display student details.

- **Program :-**

```
using System;
class Student
{
    public int RollNo { get; set; }
    public string Name { get; set; }
    public int Marks { get; set; }
    public void DisplayDetails()
    {
        Console.WriteLine("Student Details:");
        Console.WriteLine($"Roll No: {RollNo}");
        Console.WriteLine($"Name      : {Name}");
        Console.WriteLine($"Marks : {Marks}");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Student s1 = new Student(); s1.RollNo = 115;
        s1.Name = "Praful";

        s1.Marks = 95;


        s1.DisplayDetails(); Console.ReadLine();
    }
}
```

- **Output :-**

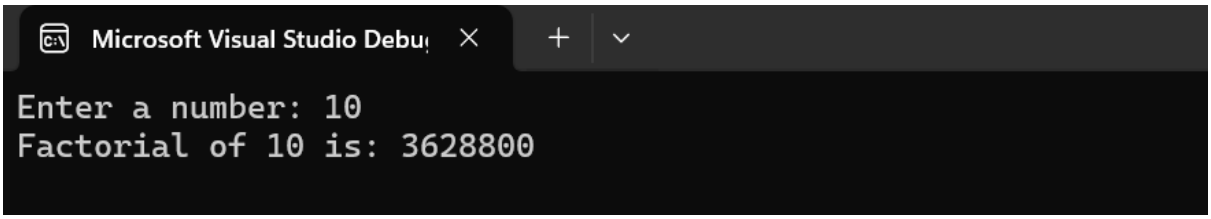


21. Create a Console Application to calculate the factorial of a number

- **Program :-**

```
using System;

class FactorialProgram
{
   static void Main(string[] args)
   {
      Console.Write("Enter a number: ");
      int num = Convert.ToInt32(Console.ReadLine());
      long factorial = 1;
      for (int i = 1; i <= num; i++)
      {
         factorial *= i;
      }
      Console.WriteLine($"Factorial of {num} is: {factorial}");

      Console.ReadLine(); // to keep console window open
   }
}
```

- **Output :-**

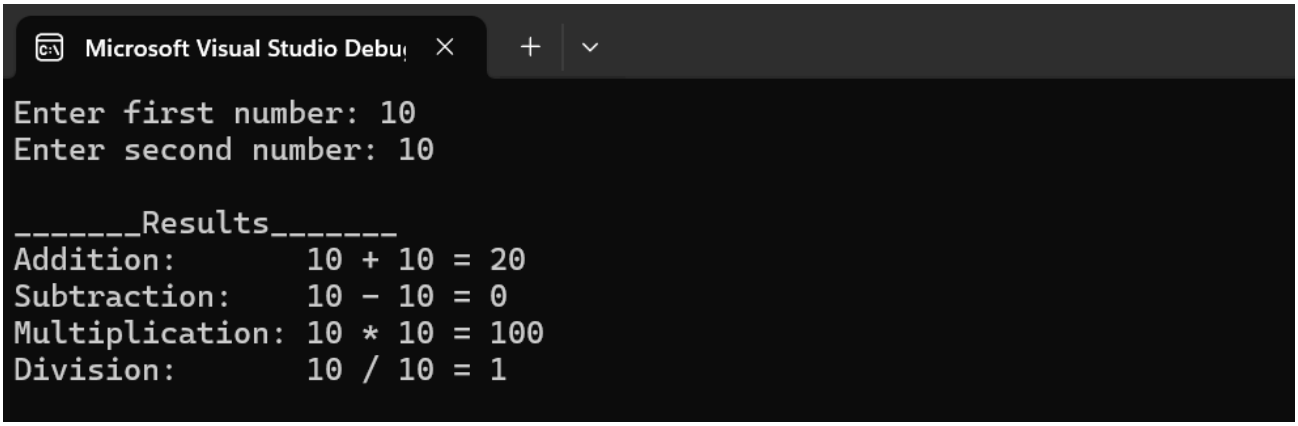


22. Create a Calculator in Console Application to Perform Add, Subtract, Multiply and Divide operation.

- **Program :-**

```
using System;

namespace CalculatorApp
{
   class Program
   {
      static void Main(string[] args)
      {
         double num1, num2;

         Console.Write("Enter first number: ");
         num1 = Convert.ToDouble(Console.ReadLine());

         Console.Write("Enter second number: ");
         num2 = Convert.ToDouble(Console.ReadLine());

         Console.WriteLine("\n_______Results_______");

         double addition = num1 + num2;
         double subtraction = num1 - num2;
         double multiplication = num1 * num2;
         double division = (num2 != 0) ? num1 / num2 : double.NaN;

         Console.WriteLine($"Addition:       {num1} + {num2} = {addition}");
         Console.WriteLine($"Subtraction:    {num1} - {num2} =
{subtraction}");
         Console.WriteLine($"Multiplication: {num1} * {num2} =
{multiplication}");

         if (num2 != 0)
            Console.WriteLine($"Division:       {num1} / {num2} =
{division}");
         else
            Console.WriteLine("Division:       Error! Division by zero is not
allowed.");
      }
   }
}
```

- **Output :-**


Microsoft Visual Studio Debu
Enter first number: 10
Enter second number: 10
_______Results_______
Addition: 10 + 10 = 20
Subtraction: 10 - 10 = 0
Multiplication: 10 * 10 = 100
Division: 10 / 10 = 1

23. Write a Program to determine eligibility for admission to professional course based on following cat

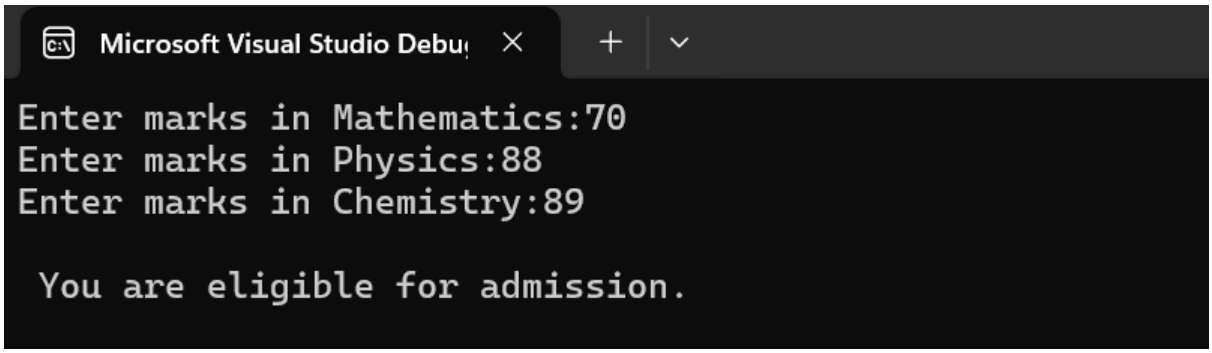
Math >= 65 , Physics >= 55 , Chemistry >= 50

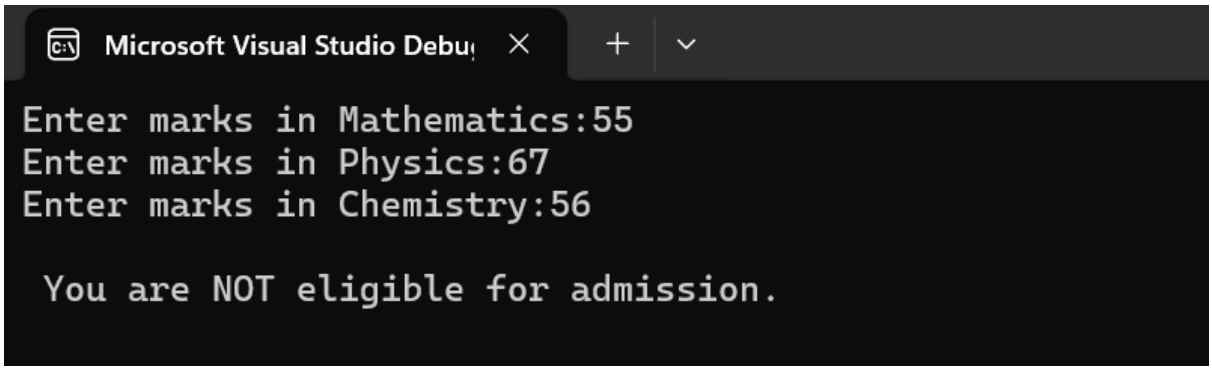
Totals on all three courses >= 180 or Total in math and course>=140

- **Program :-**

```
using System;
class Program
{
   static void Main()
   {
      int math, physics, chemistry;
      // Input marks
      Console.Write("Enter marks in Mathematics:");
      math = int.Parse(Console.ReadLine());

      Console.Write("Enter marks in Physics:");
      physics = int.Parse(Console.ReadLine());

      Console.Write("Enter marks in Chemistry:");
      chemistry = int.Parse(Console.ReadLine());

      int total = math + physics + chemistry;
      int mathPhysicsTotal = math + physics;

      if (math >= 65 && physics >= 55 && chemistry >= 50 && (total >= 180
|| mathPhysicsTotal >= 140))
      {
         Console.WriteLine("\n You are eligible for admission.");
      }
      else
      {
         Console.WriteLine("\n You are NOT eligible for admission.");
      }
      Console.ReadLine();
   }
}
```

➢ **Output :-**


Microsoft Visual Studio Debu
Enter marks in Mathematics:70
Enter marks in Physics:88
Enter marks in Chemistry:89
You are eligible for admission.

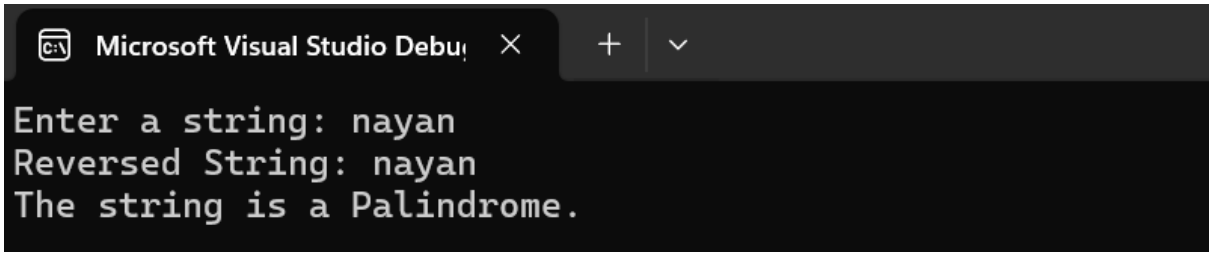

Microsoft Visual Studio Debu
Enter marks in Mathematics:55
Enter marks in Physics:67
Enter marks in Chemistry:56
You are NOT eligible for admission.

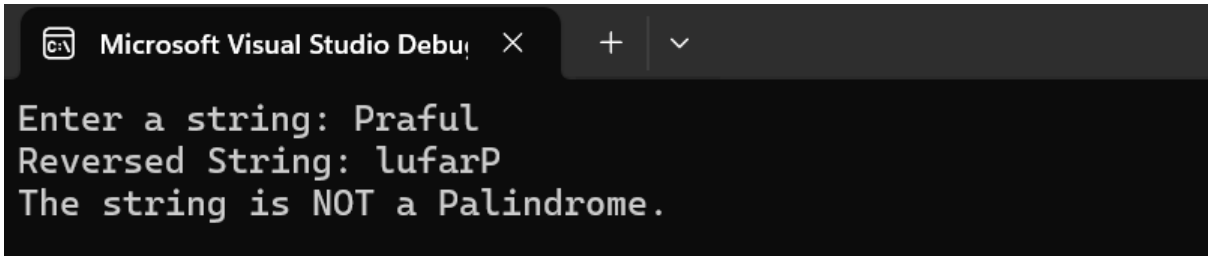
24. Write a program to reverse a string and check if it is palindrome.

- **Program :-**

```
using System;
class PalindromeCheck
{
   static void Main()
   {
      Console.Write("Enter a string: ");
      string input = Console.ReadLine();
      char[] charArray = input.ToCharArray();
      Array.Reverse(charArray);
      string reversed = new string(charArray);
      Console.WriteLine("Reversed String: " + reversed);
   if (input.Equals(reversed, StringComparison.OrdinalIgnoreCase))
      {
         Console.WriteLine("The string is a Palindrome.");
      }
      else
      {
         Console.WriteLine("The string is NOT a Palindrome.");
      }
   }
}
```

- **Output :-**



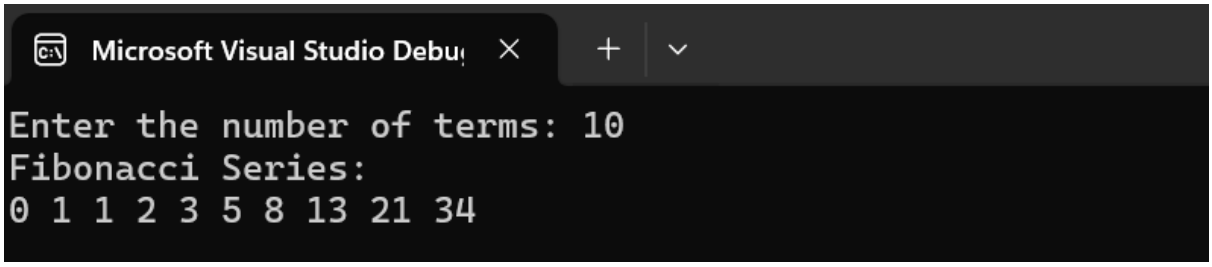


25. Write a Console Application to print a Fibonacci series.

- **Program :-**

```
using System;
class FibonacciSeries
{
   static void Main()
   {
      Console.Write("Enter the number of terms: ");
      int n = int.Parse(Console.ReadLine());
      int first = 0, second = 1, next;
      Console.WriteLine("Fibonacci Series:");
      for (int i = 1; i <= n; i++)
      {
         Console.Write(first + " "); next = first + second;
         first = second; second = next;
      }
      Console.WriteLine();
   }
}
```

- **Output :-**

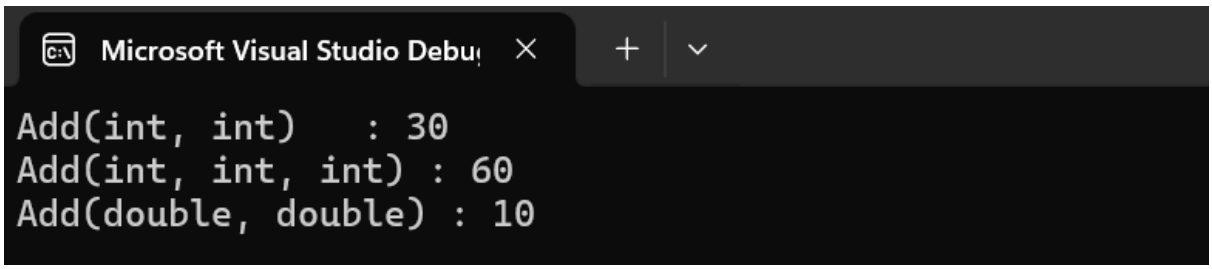


26. Write a Console Application to Demonstrate Function Overloading in C#.NET

- **Program :-**

```
using System;
class FunctionOverloading
{
   public int Add(int a, int b)
   {
      return a + b;
   }
   public int Add(int a, int b, int c)
   {
      return a + b + c;
   }
   public double Add(double a, double b)
   {
      return a + b;
   }
   static void Main()
   {
      FunctionOverloading obj = new FunctionOverloading();
      Console.WriteLine("Add(int, int)          : " + obj.Add(10, 20));
      Console.WriteLine("Add(int, int, int) : " + obj.Add(10, 20, 30));
      Console.WriteLine("Add(double, double) : " + obj.Add(5.5, 4.5));
   }
}
```

- **Output :-**



27. Create a window application to demonstrate simple database connectivity with wizard And display the data on the form.

- **Program :-**

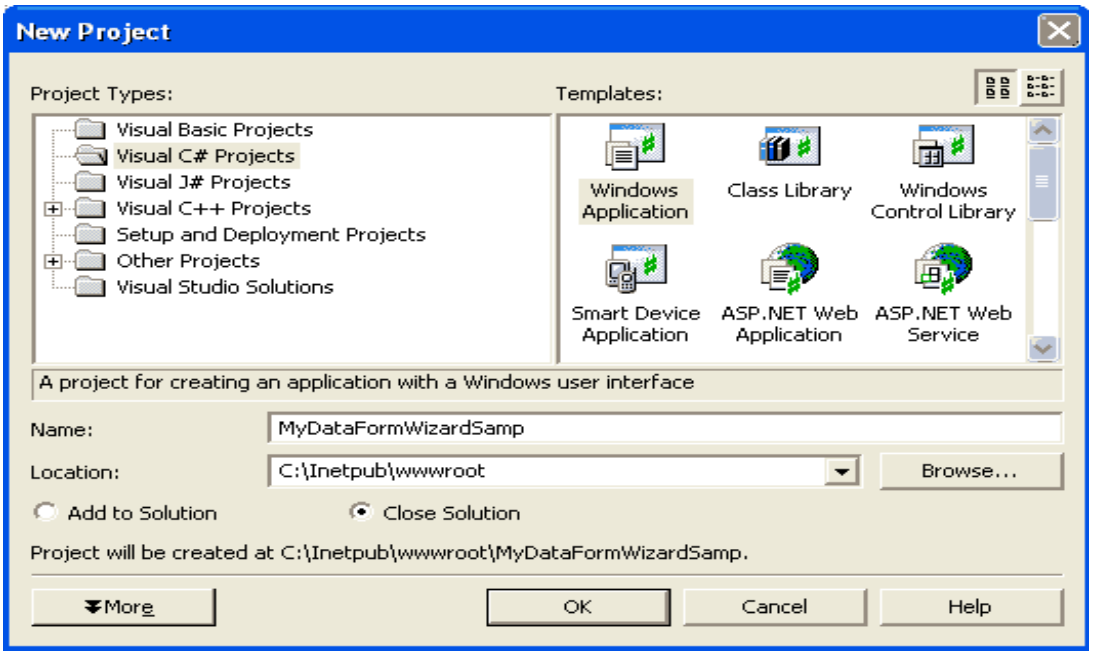
```
using System.Reflection;

public void LoadDataSet()
   {
      the call to FillDataSet.
      MyDataFormWizardSamp.MyDS objDataSetTemp;
      objDataSetTemp = new MyDataFormWizardSamp.MyDS();
try
   {
      this.FillDataSet(objDataSetTemp);
   }

catch (System.Exception eFillDataSet)
   {
      throw eFillDataSet;
   }

try
   {
      objMyDS.Clear();

      objMyDS.Merge(objDataSetTemp);
   }
catch (System.Exception eLoadMerge)
   {
      throw eLoadMerge;
   }
}
```

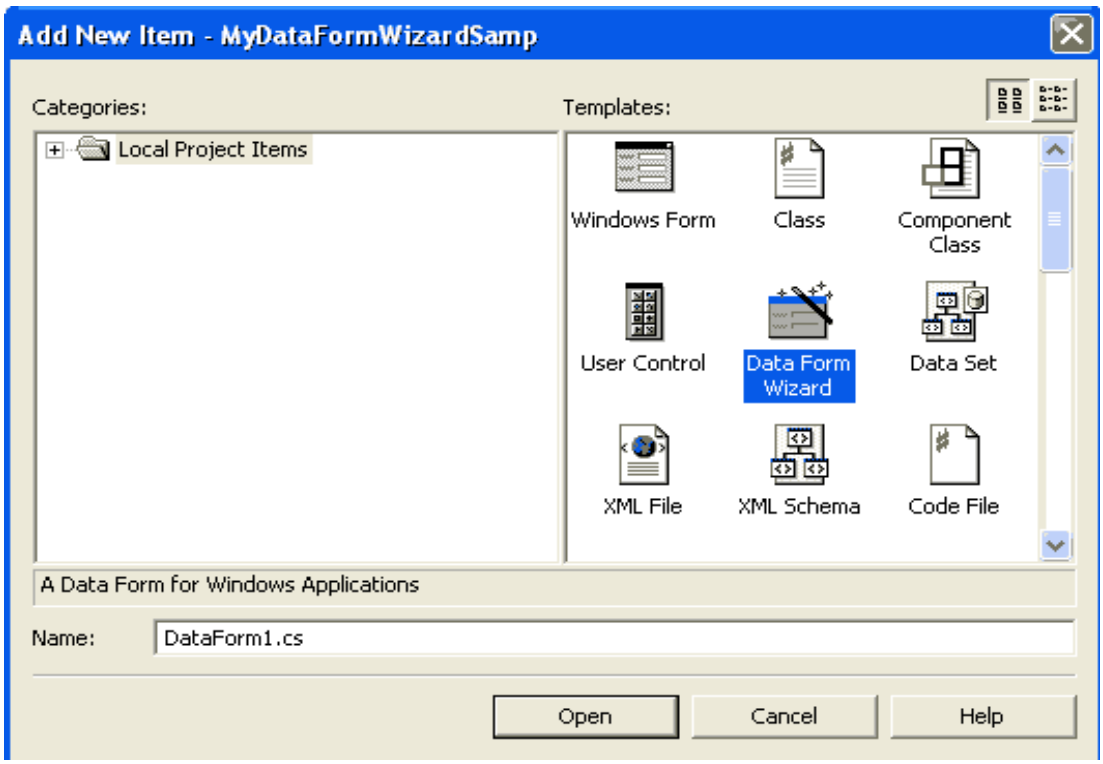
Listing 4-8. The FillDataSet method generated by the Data Form Wizard

```
public void FillDataSet(MyDataFormWizardSamp.MyDS dataSet)
{
   dataSet.EnforceConstraints = false;

   try
   {
      this.oleDbConnection1.Open();
      OleDbDataAdapter1.
      this.oleDbDataAdapter1.Fill(dataSet);
      this.oleDbDataAdapter2.Fill(dataSet);
   }

   catch (System.Exception fillException)
   {
```

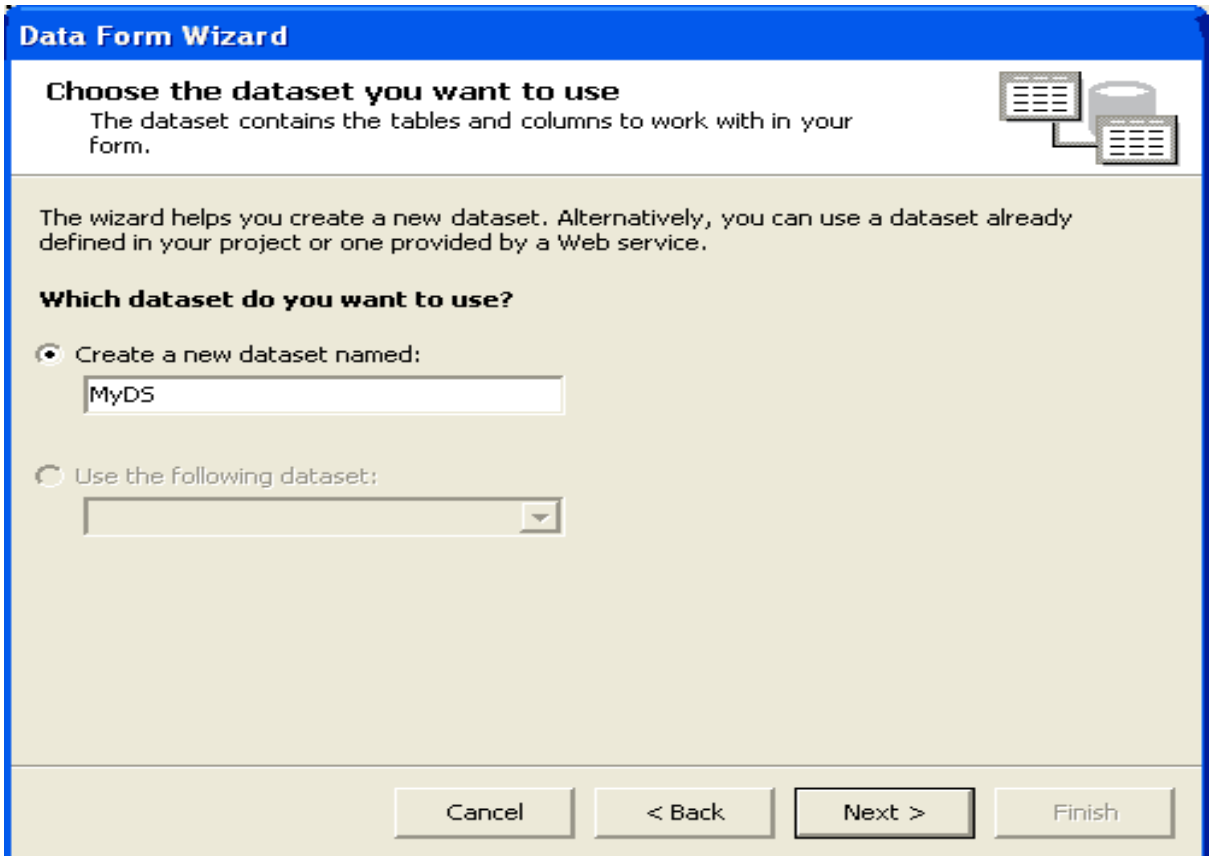
```
        throw fillException;
    }

    finally
    {
        dataSet.EnforceConstraints = true;
        this.oleDbConnection1.Close();
    }
```

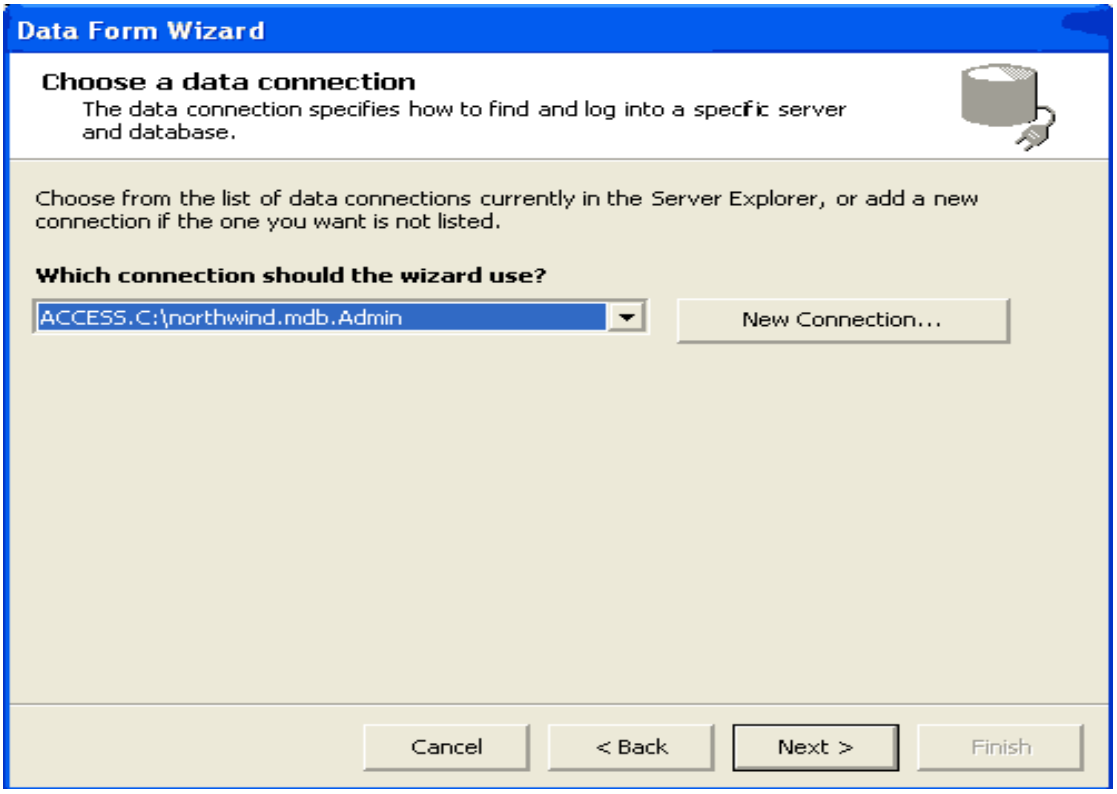
- **Output :-**

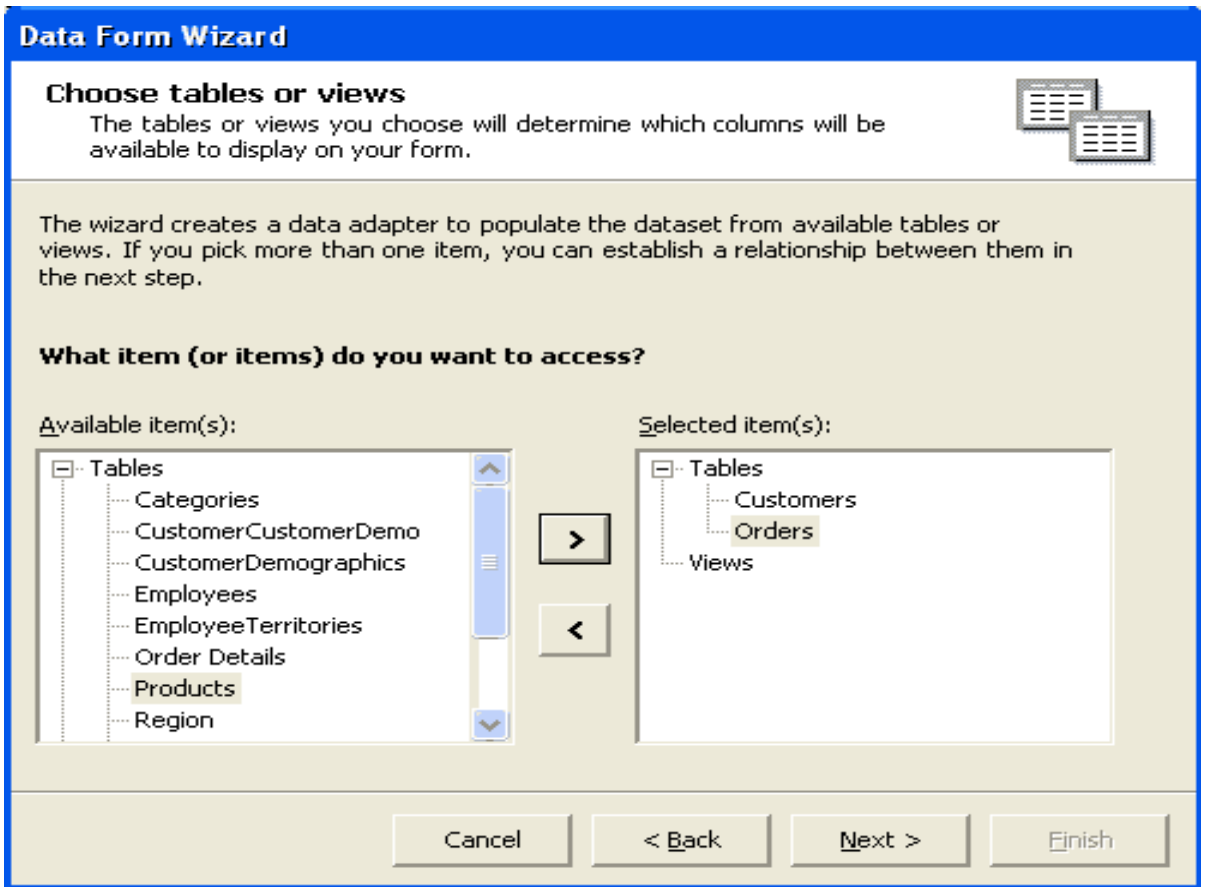


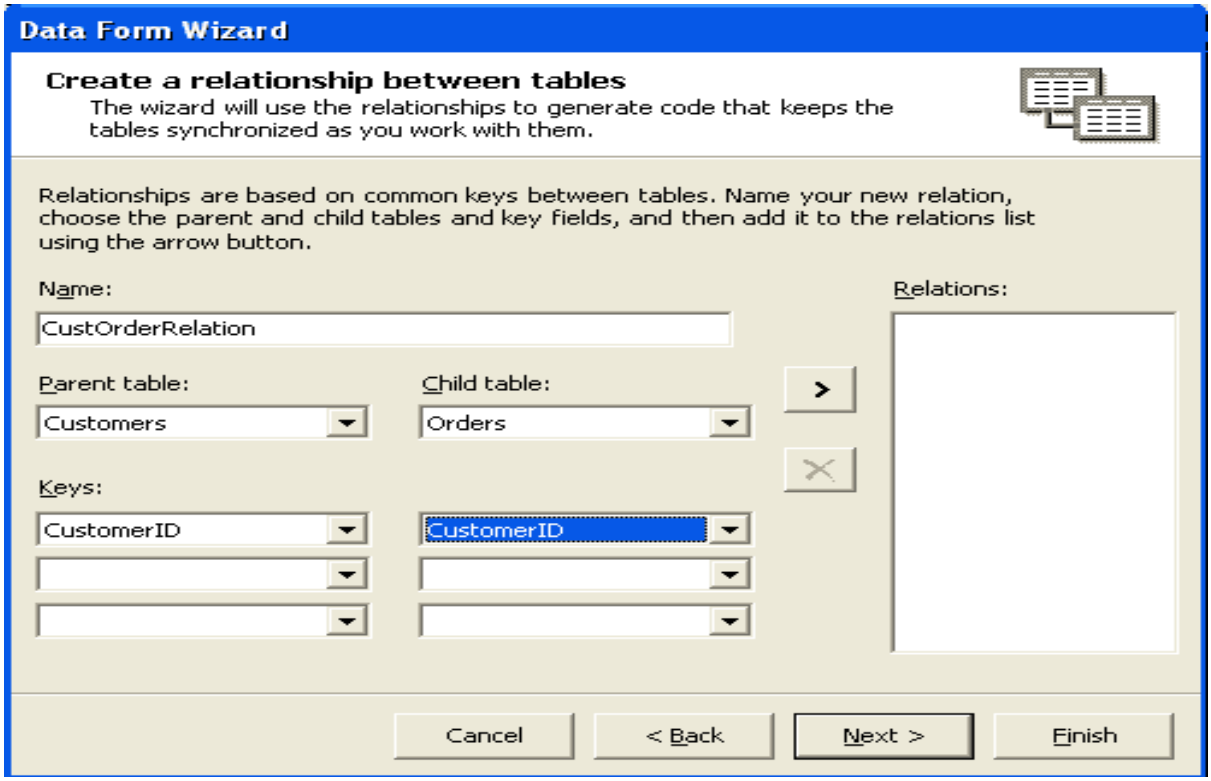


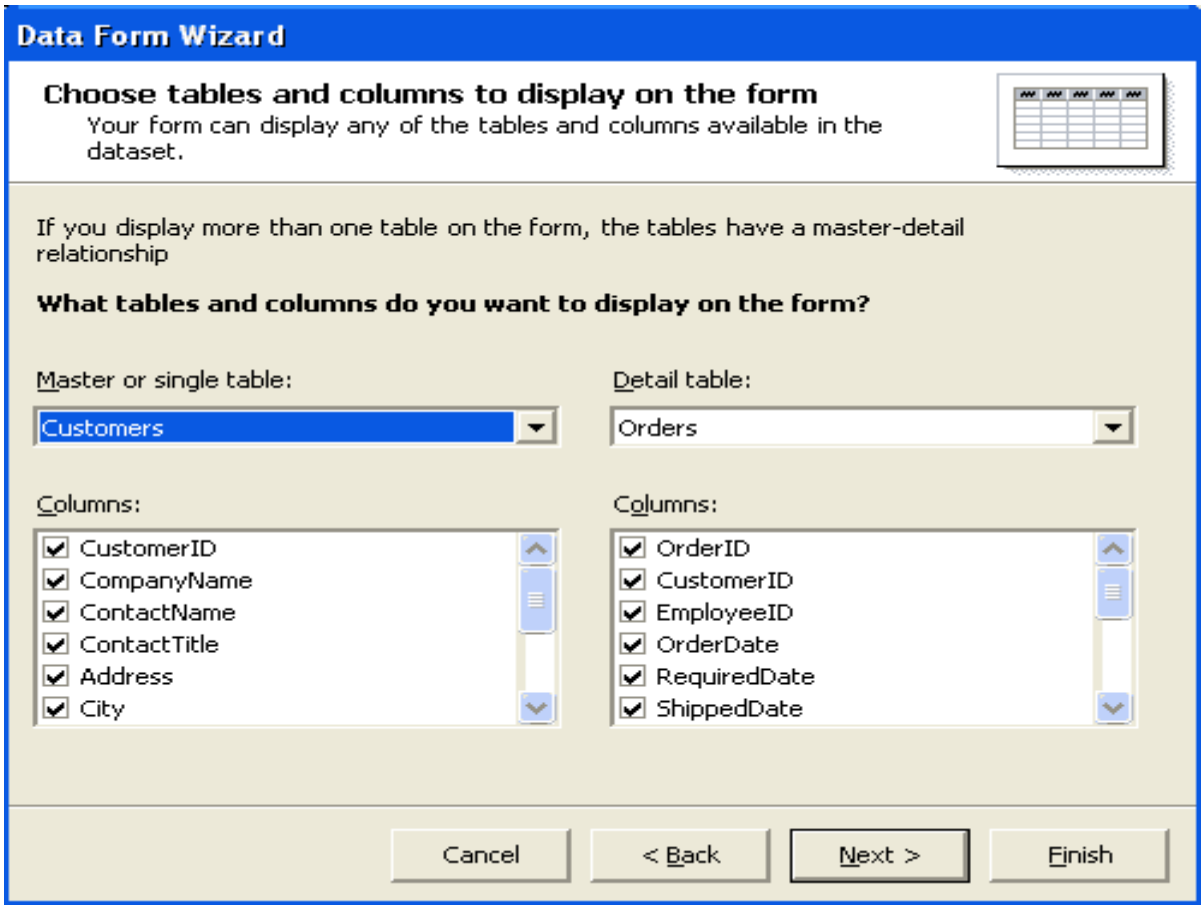

Data Form Wizard
Welcome to the DataForm Wizard
This wizard will create a new form with data-bound controls to display information from the dataset you specify.
Click Next to continue.
Cancel
< Back
Next >
Finish

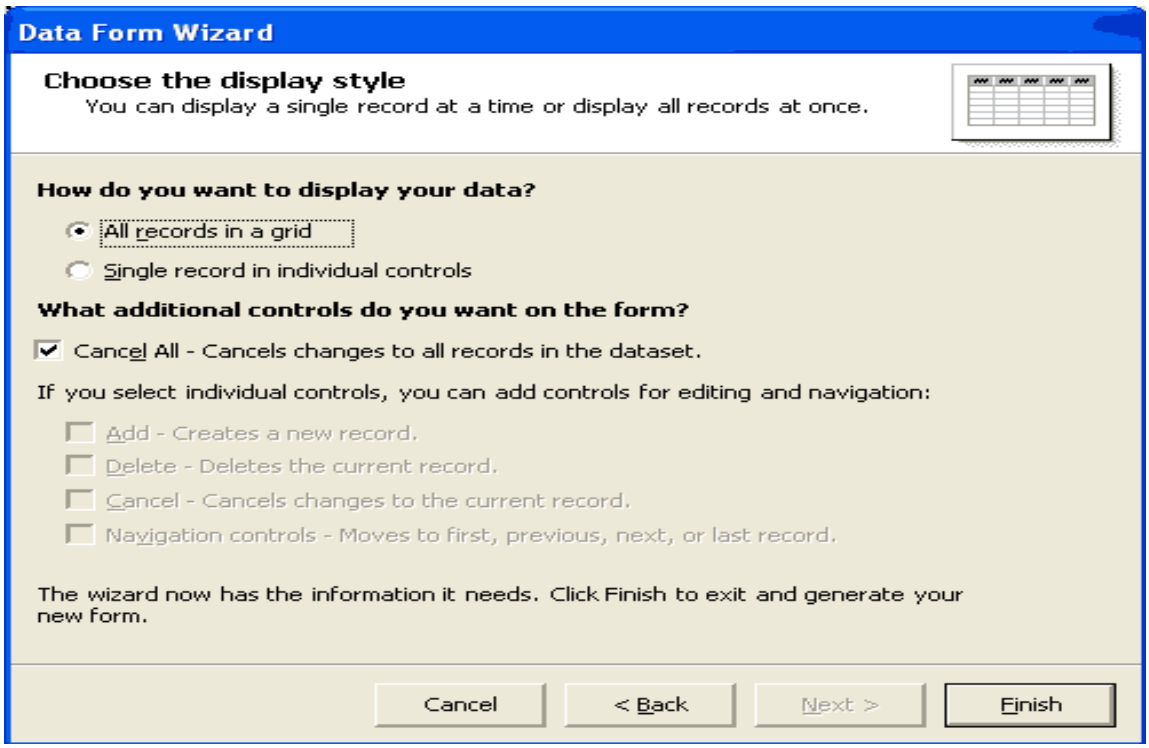

Data Form Wizard
Choose the dataset you want to use
The dataset contains the tables and columns to work with in your form.
The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.
Which dataset do you want to use?
Create a new dataset named:
MyDS
Use the following dataset:
Cancel
< Back
Next >
Finish

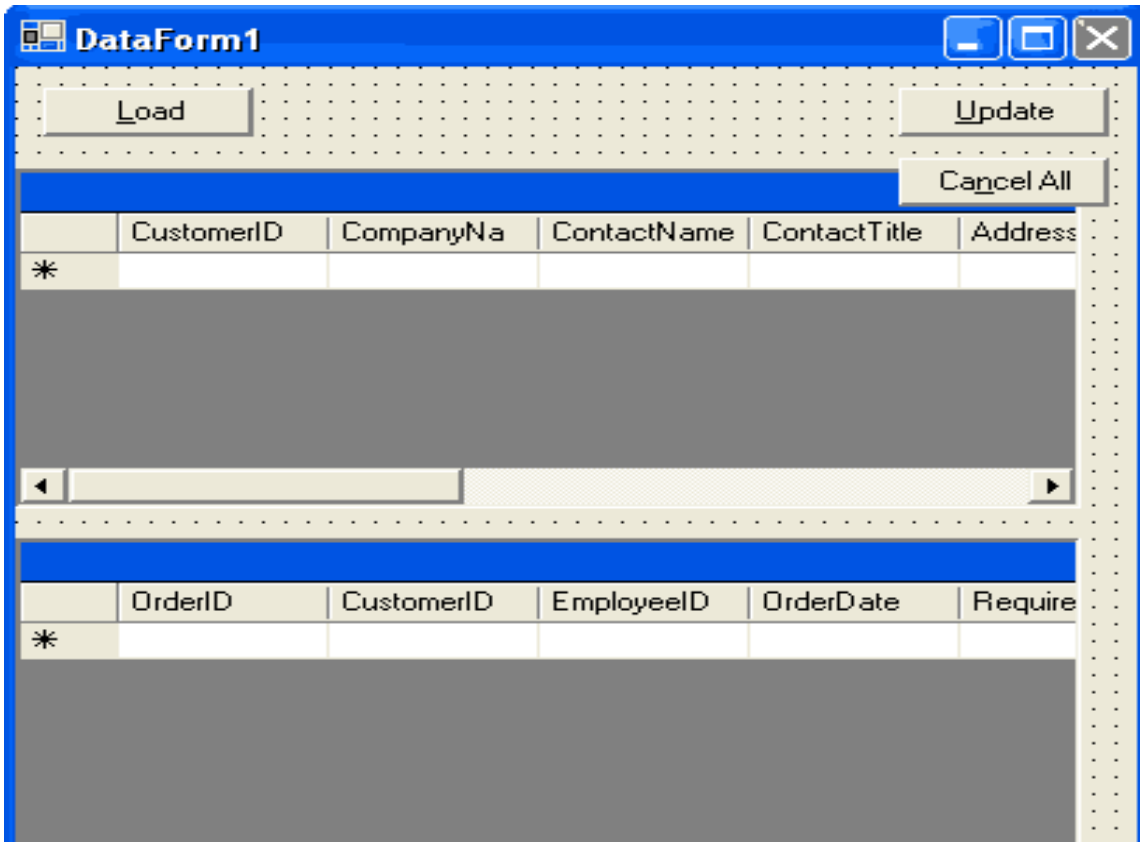

Data Form Wizard
Choose a data connection
The data connection specifies how to find and log into a specific server and database.
Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.
Which connection should the wizard use?
ACCESS.C:\northwind.mdb.Admin
New Connection...
Cancel
< Back
Next >
Finish

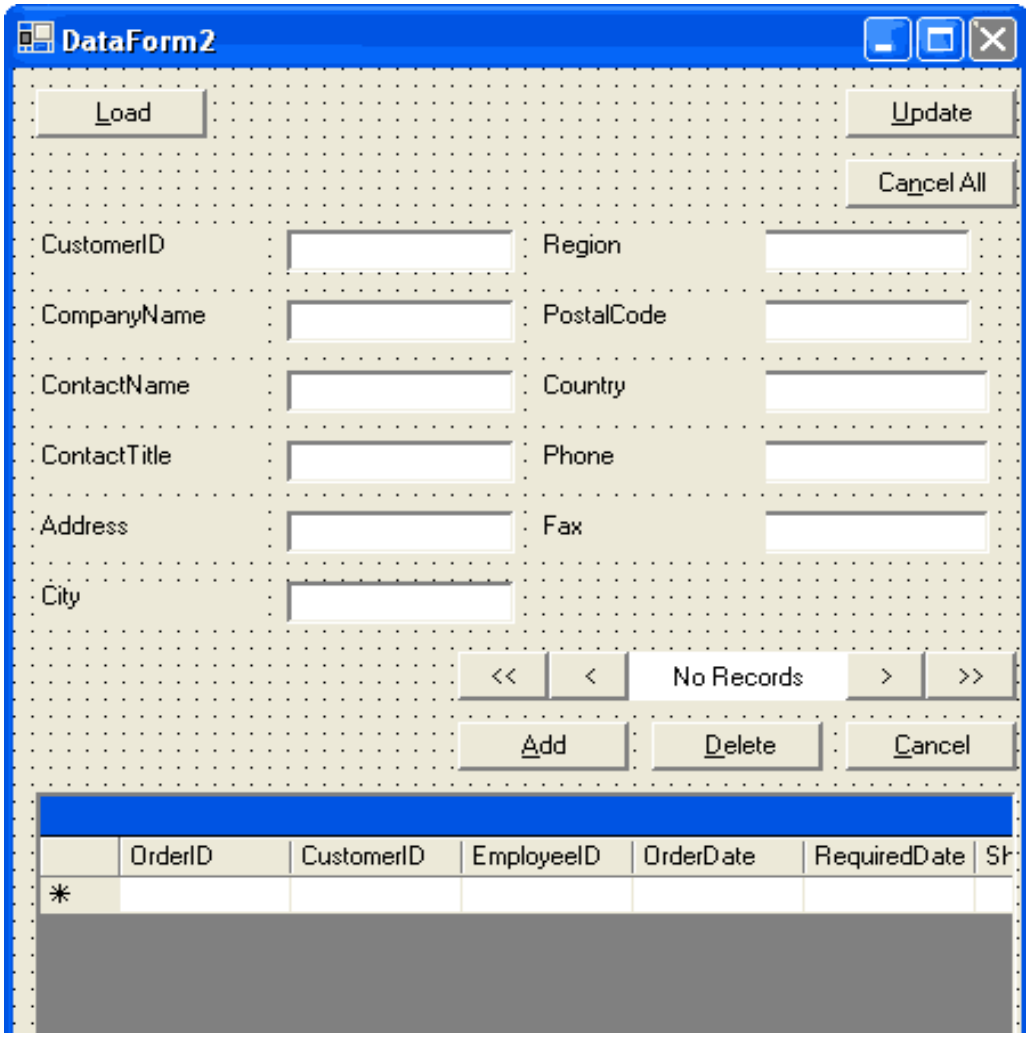

Data Form Wizard
Choose tables or views
The tables or views you choose will determine which columns will be available to display on your form.
The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.
What item (or items) do you want to access?
Available item(s):
Tables
Categories
CustomerCustomerDemo
CustomerDemographics
Employees
EmployeeTerritories
Order Details
Products
Region
>
<
Selected item(s):
Tables
Customers
Orders
Views
Cancel
< Back
Next >
Finish


Data Form Wizard
Create a relationship between tables
The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.
Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.
Name:
CustOrderRelation
Relations:
Parent table:
Customers
Child table:
Orders
>
Keys:
CustomerID
CustomerID
Cancel
< Back
Next >
Finish


Data Form Wizard
Choose tables and columns to display on the form
Your form can display any of the tables and columns available in the dataset.
If you display more than one table on the form, the tables have a master-detail relationship
What tables and columns do you want to display on the form?
Master or single table:
Customers
Detail table:
Orders
Columns:
CustomerID
CompanyName
ContactName
ContactTitle
Address
City
Columns:
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate
ShippedDate
Cancel
< Back
Next >
Finish


Data Form Wizard
Choose the display style
You can display a single record at a time or display all records at once.
How do you want to display your data?
All records in a grid
Single record in individual controls
What additional controls do you want on the form?
Cancel All - Cancels changes to all records in the dataset.
If you select individual controls, you can add controls for editing and navigation:
Add - Creates a new record.
Delete - Deletes the current record.
Cancel - Cancels changes to the current record.
Navigation controls - Moves to first, previous, next, or last record.
The wizard now has the information it needs. Click Finish to exit and generate your new form.
Cancel
< Back
Next >
Finish


DataForm1
Load
Update
Cancel All
CustomerID
CompanyNa
ContactName
ContactTitle
Address
OrderID
CustomerID
EmployeeID
OrderDate
Require

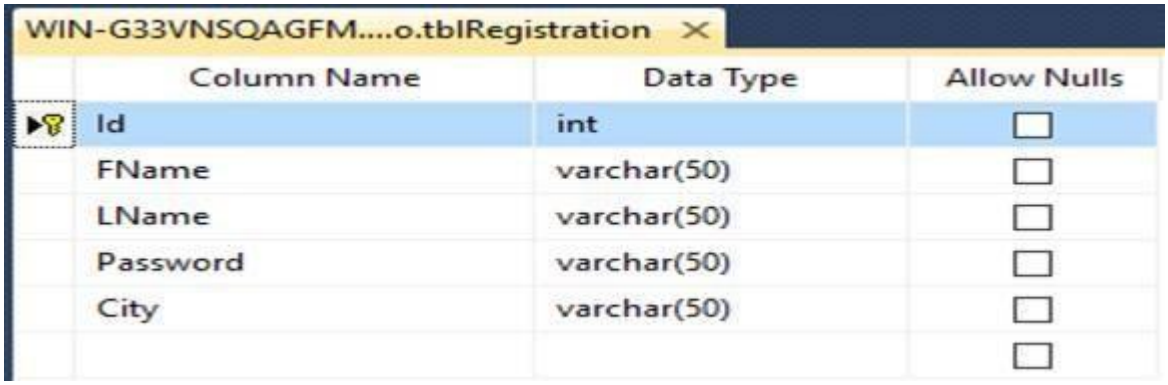

DataForm2
Load
Update
Cancel All
CustomerID
Region
CompanyName
PostalCode
ContactName
Country
ContactTitle
Phone
Address
Fax
City
<<
<
No Records
>
>>
Add
Delete
Cancel
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate

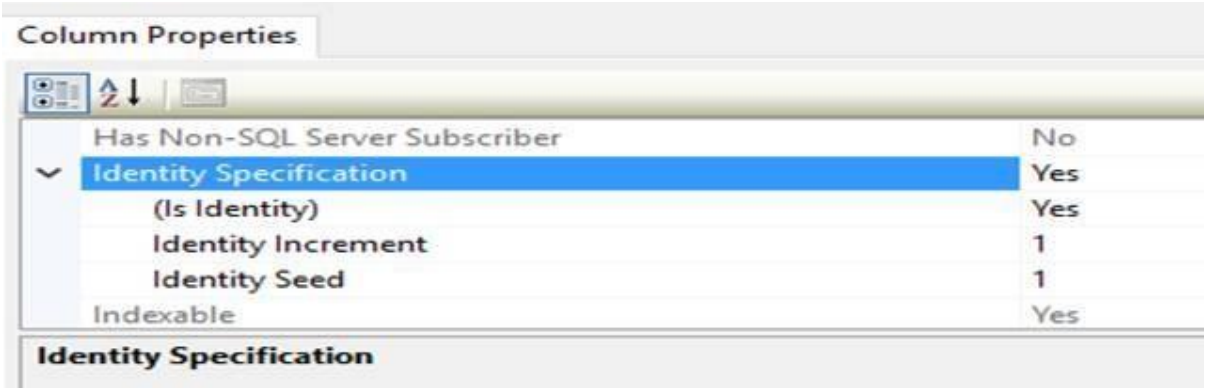
28. Create simple student's registration form and perform the operation like insert, update And delete.

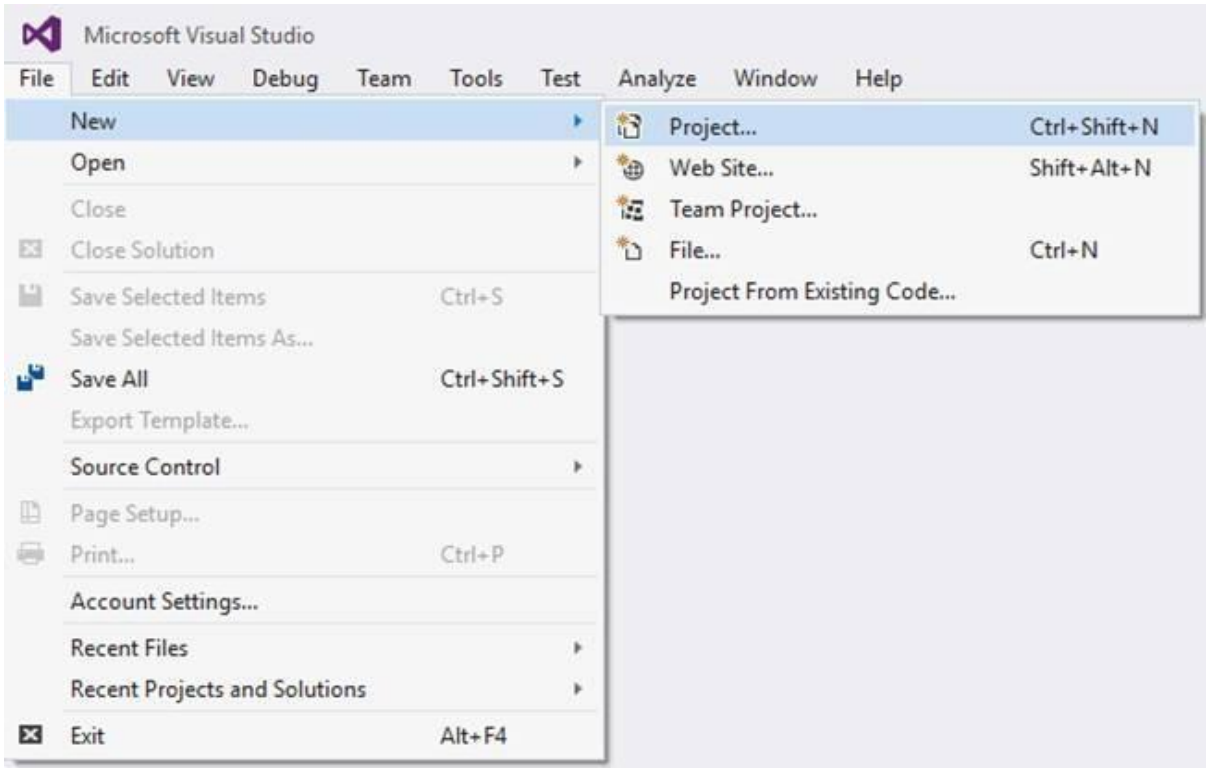
- **Program :-**

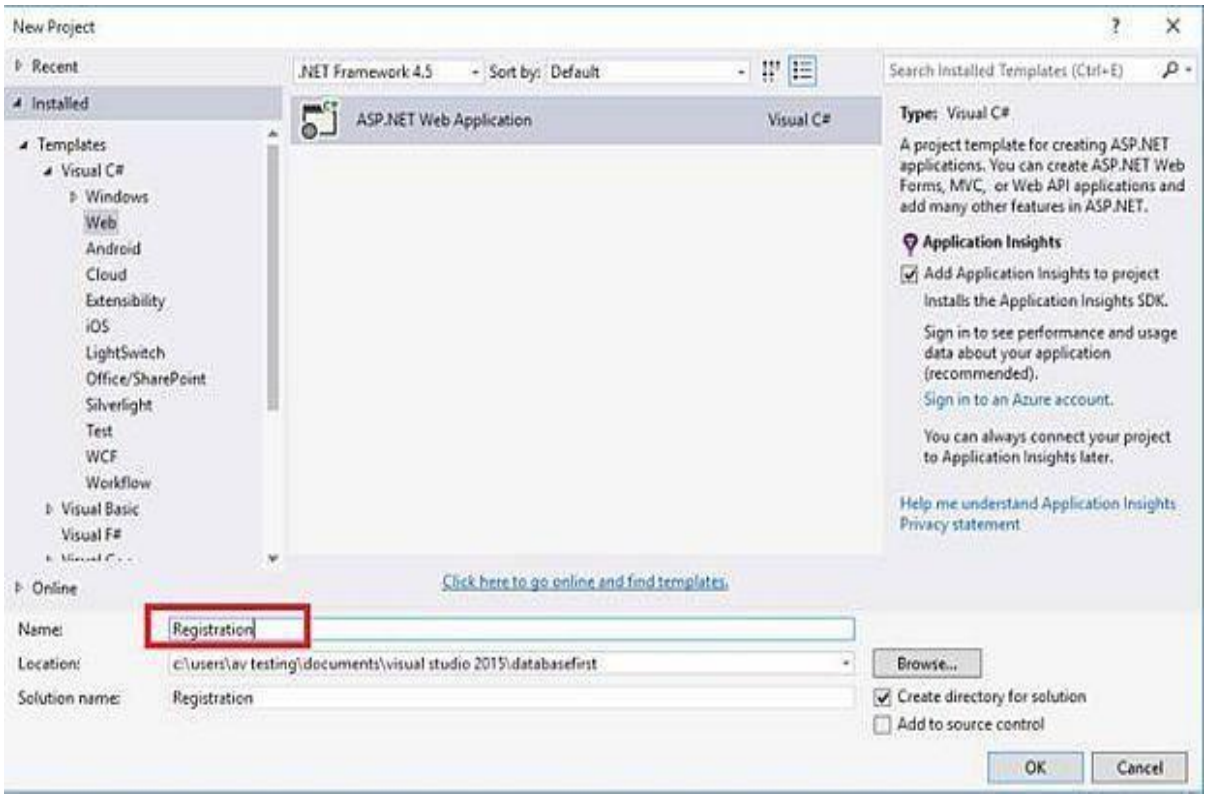
```
@model Registration.Models.tblRegistration
@{
   Layout = null;
}
< !DOCTYPE html >
< html >
< head >
   < meta name = "viewport" content = "width=device-width" />
   < title > Index </ title >
</ head >
< body >

@using(Html.BeginForm())
{
   @Html.AntiForgeryToken()
   < div class= "form-horizontal" >
      < h4 > tblRegistration </ h4 >
      < hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
   < div class= "form-group" >
      @Html.LabelFor(model => model.FName, htmlAttributes: new { @class
= "control- label col-md-2" })
       < div class= "col-md-10" >
          @Html.EditorFor(model => model.FName, new { htmlAttributes =
new { @class = "form-control" } }
          @Html.ValidationMessageFor(model => model.FName, "", new{
@class = "text-danger" })
       </ div >
   </ div >
   < div class= "form-group" >
   @Html.LabelFor(model => model.LName, htmlAttributes: new { @class =
"control- label col-md-2" })
   < div class= "col-md-10" >
   @Html.EditorFor(model => model.LName, new { htmlAttributes = new {
@class = "form-control" } })
@Html.ValidationMessageFor(model => model.LName, "", new {@class =
"text-danger" })
   </ div >
</ div >
< div class= "form-group" >
   @Html.LabelFor(model => model.Password, htmlAttributes: new { @class
= "control-label col-md-2" })

   < div class= "col-md-10" >
```

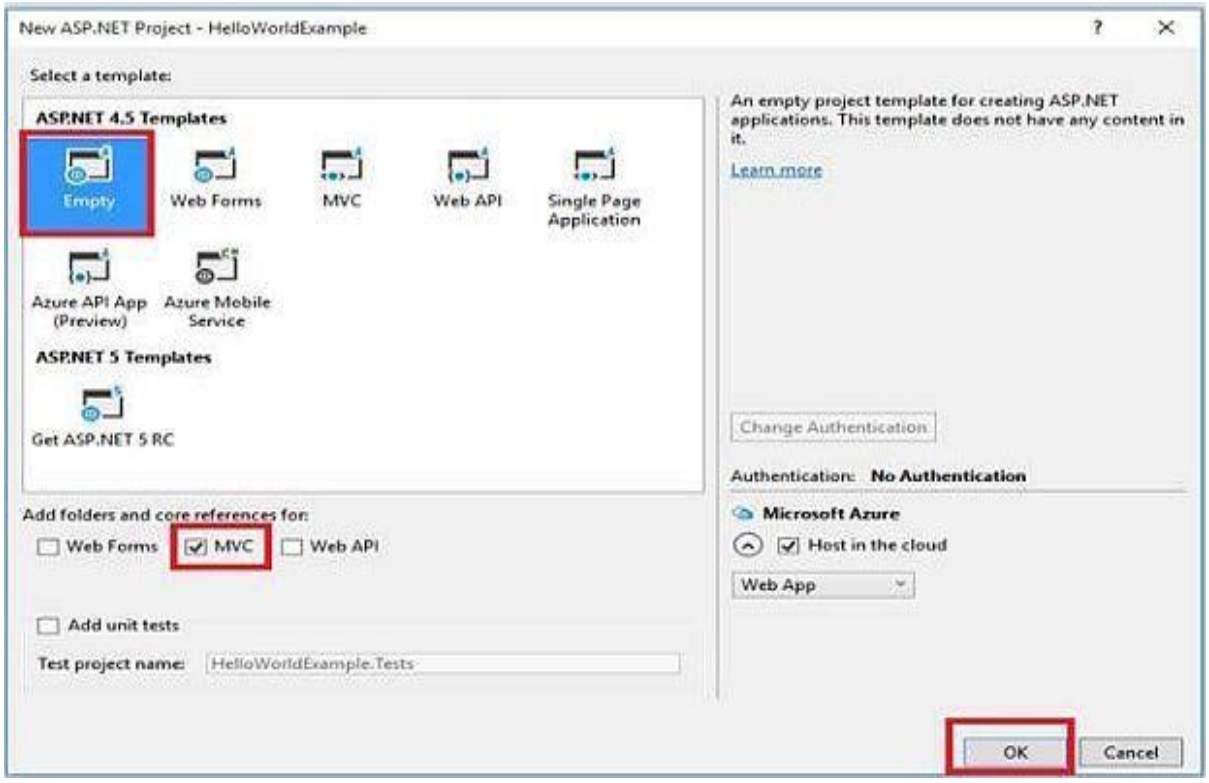
```
        @Html.EditorFor(model => model.Password, new { htmlAttributes =
new { @class = "form-control"}})
        @Html.ValidationMessageFor(model => model.Password, "", new {
@class = "text-danger" })
    </ div >
</ div >
< div class= "form-group" >
    @Html.LabelFor(model => model.City, htmlAttributes: new { @class =
"control-label col-md-2" })
    < div class= "col-md-10" >
        @Html.EditorFor(model => model.City, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.City, "", new { @class =
"text-danger" })
    </ div >
</ div >
< div class= "form-group" >
    < div class= "col-md-offset-2 col-md-10" >
        < input type = "submit" value = "Create" class= "btn btn-default" />
    </ div >
    </ div >
    }
    </ div >
</ body >
</ html >
```

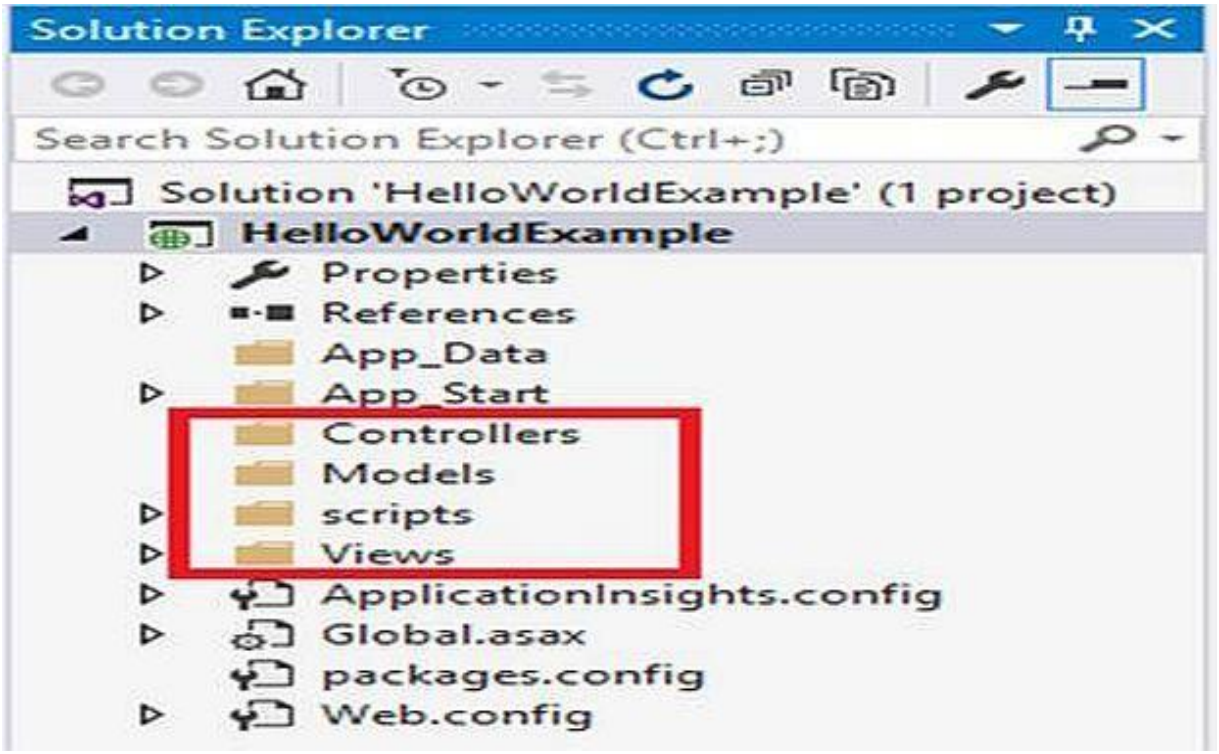


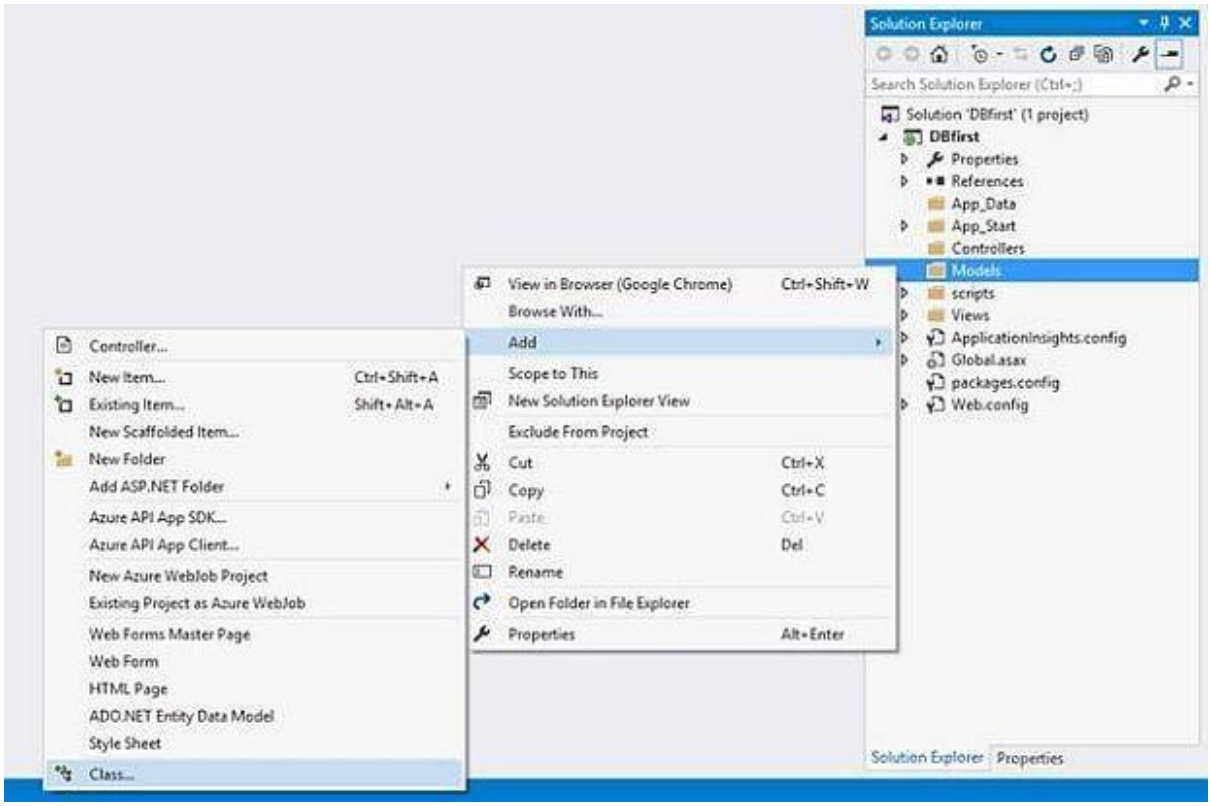


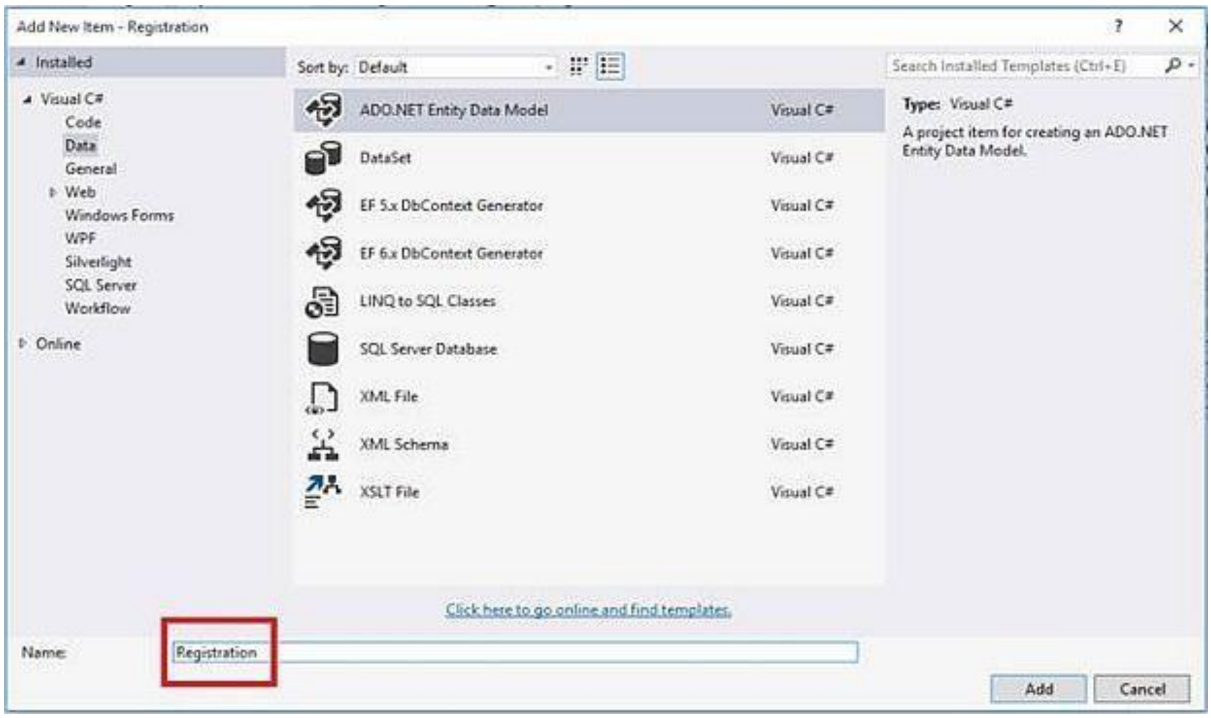

Microsoft Visual Studio
File Edit View Debug Team Tools Test Analyze Window Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Source Control
Page Setup...
Print... Ctrl+P
Account Settings...
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
File... Ctrl+N
Project From Existing Code...


New Project
Recent
Installed
Templates
Visual C#
Windows
Web
Android
Cloud
Extensibility
iOS
LightSwitch
Office/SharePoint
Silverlight
Test
WCF
Workflow
Visual Basic
Visual F#
Online
.NET Framework 4.5
Sort by: Default
ASP.NET Web Application
Visual C#
Search Installed Templates (Ctrl+E)
Type: Visual C#
A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Application Insights
Add Application Insights to project
Installs the Application Insights SDK.
Sign in to see performance and usage data about your application (recommended).
Sign in to an Azure account.
You can always connect your project to Application Insights later.
Help me understand Application Insights
Privacy statement
Click here to go online and find templates.
Name: Registration
Location: c:\users\av testing\documents\visual studio 2015\databasefirst
Solution name: Registration
Browse...
Create directory for solution
Add to source control
OK
Cancel

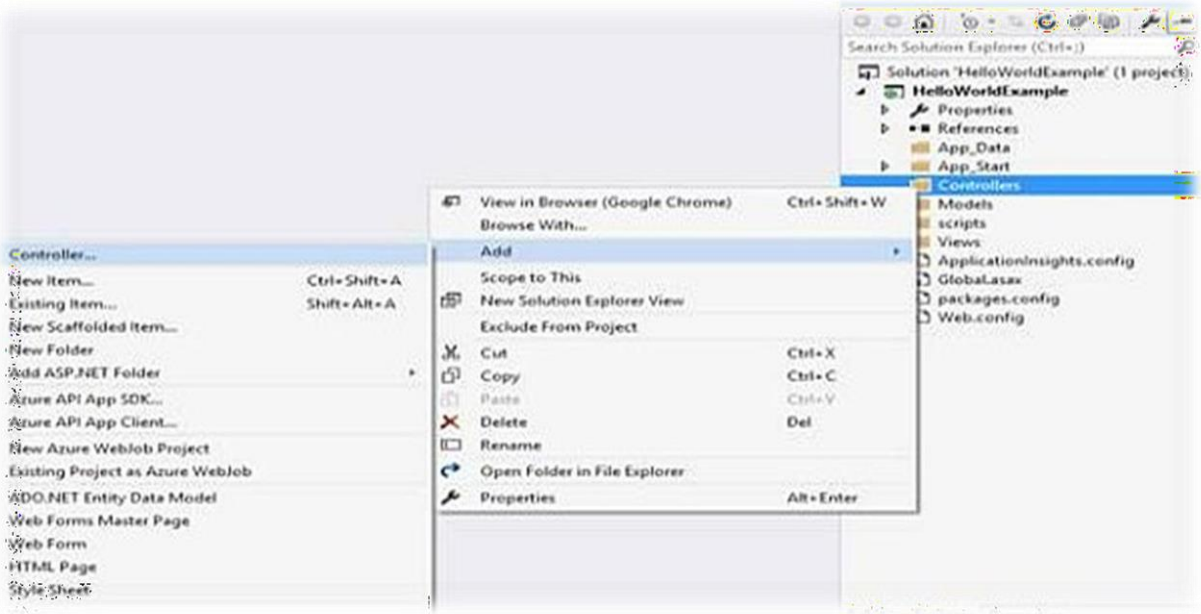

New ASP.NET Project - HelloWorldExample
Select a template:
ASP.NET 4.5 Templates
Empty
Web Forms
MVC
Web API
Single Page Application
Azure API App (Preview)
Azure Mobile Service
ASP.NET 5 Templates
Get ASP.NET 5 RC
An empty project template for creating ASP.NET applications. This template does not have any content in it.
Learn more
Change Authentication
Authentication: No Authentication
Microsoft Azure
Host in the cloud
Web App
Add folders and core references for:
Web Forms
MVC
Web API
Add unit tests
Test project name: HelloWorldExample.Tests
OK
Cancel

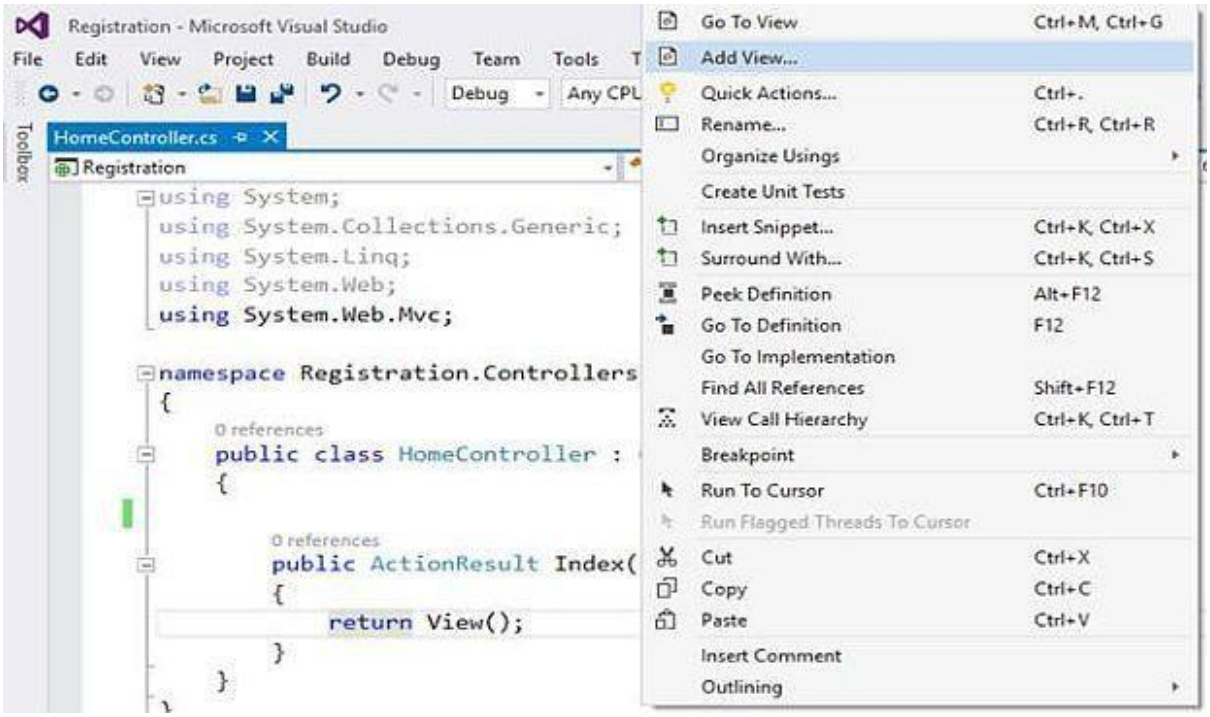

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HelloWorldExample' (1 project)
HelloWorldExample
Properties
References
App_Data
App_Start
Controllers
Models
scripts
Views
ApplicationInsights.config
Global.asax
packages.config
Web.config

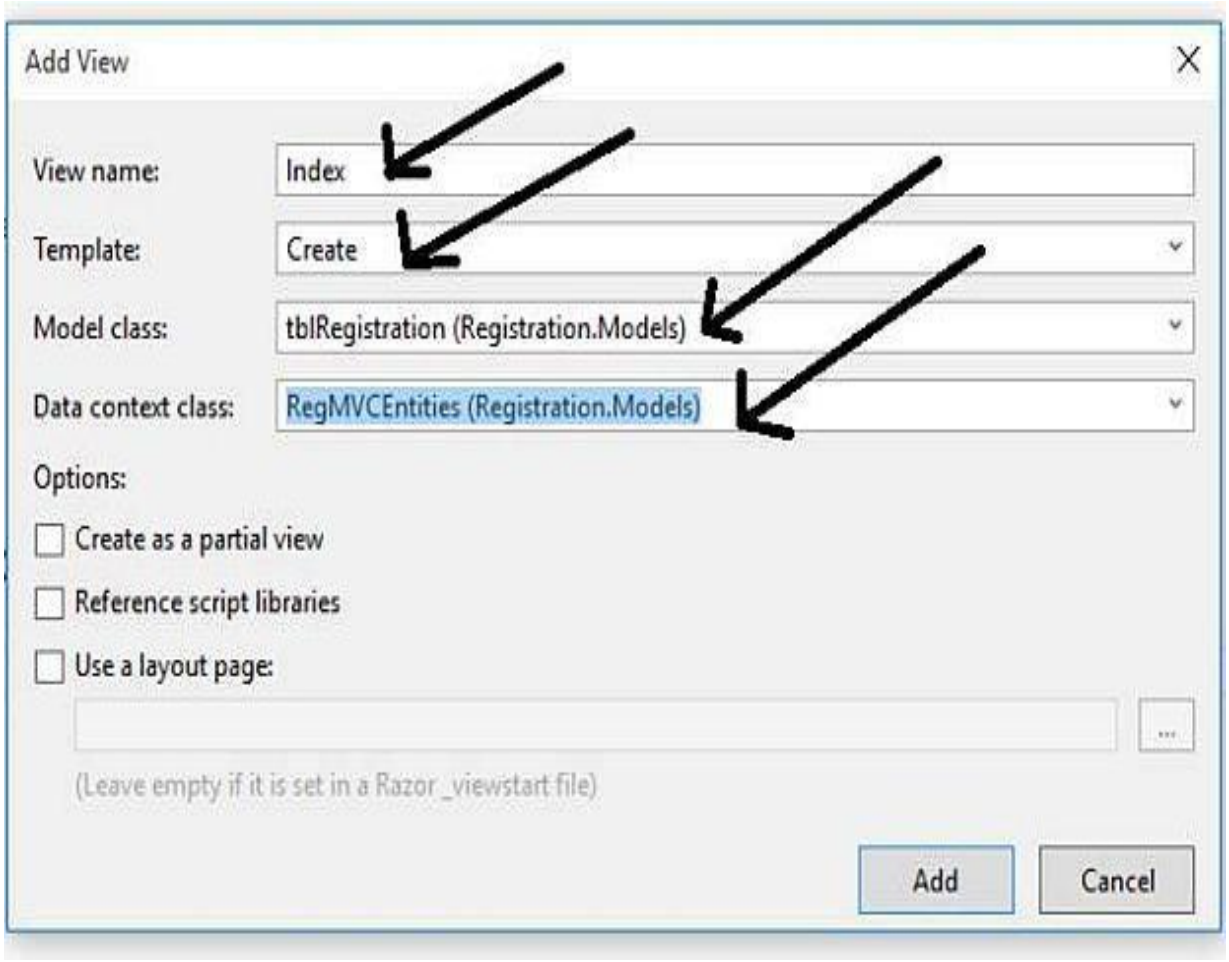

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DBfirst' (1 project)
DBfirst
Properties
References
App_Data
App_Start
Controllers
Models
scripts
Views
ApplicationInsights.config
Global.asax
packages.config
Web.config
Solution Explorer
Properties
View in Browser (Google Chrome) Ctrl+Shift+W
Browse With...
Add
Scope to This
New Solution Explorer View
Exclude From Project
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename
Open Folder in File Explorer
Properties Alt+Enter
Controller...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Scaffolded Item...
New Folder
Add ASP.NET Folder
Azure API App SDK...
Azure API App Client...
New Azure WebJob Project
Existing Project as Azure WebJob
Web Forms Master Page
Web Form
HTML Page
ADO.NET Entity Data Model
Style Sheet
Class...


Add New Item - Registration
Installed
Visual C#
Code
Data
General
Web
Windows Forms
WPF
Silverlight
SQL Server
Workflow
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
ADO.NET Entity Data Model Visual C#
DataSet Visual C#
EF 5.x DbContext Generator Visual C#
EF 6.x DbContext Generator Visual C#
LINQ to SQL Classes Visual C#
SQL Server Database Visual C#
XML File Visual C#
XML Schema Visual C#
XSLT File Visual C#
Type: Visual C#
A project item for creating an ADO.NET Entity Data Model.
Click here to go online and find templates.
Name: Registration
Add
Cancel


```
namespace Registration.Models
{
    using System;
    using System.Collections.Generic;

    1 reference
    public partial class tblRegistration
    {
        0 references
        public int Id { get; set; }
        0 references
        public string FName { get; set; }
        0 references
        public string LName { get; set; }
        0 references
        public string Password { get; set; }
        0 references
        public string City { get; set; }
    }
}
```




Registration - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools
Debug
HomeController.cs
Registration
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace Registration.Controllers
{
0 references
public class HomeController :
{
0 references
public ActionResult Index(
{
return View();
}
}
Go To View Ctrl+M, Ctrl+G
Add View...
Quick Actions... Ctrl+.
Rename... Ctrl+R, Ctrl+R
Organize Usings
Create Unit Tests
Insert Snippet... Ctrl+K, Ctrl+X
Surround With... Ctrl+K, Ctrl+S
Peek Definition Alt+F12
Go To Definition F12
Go To Implementation
Find All References Shift+F12
View Call Hierarchy Ctrl+K, Ctrl+T
Breakpoint
Run To Cursor Ctrl+F10
Run Flagged Threads To Cursor
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Insert Comment
Outlining


Add View
View name: Index
Template: Create
Model class: tblRegistration (Registration.Models)
Data context class: RegMVCEntities (Registration.Models)
Options:
Create as a partial view
Reference script libraries
Use a layout page:
(Leave empty if it is set in a Razor _viewstart file)
Add
Cancel

- **Output :-**

WIN-G33VNSQAGFM....o.tblRegistration X

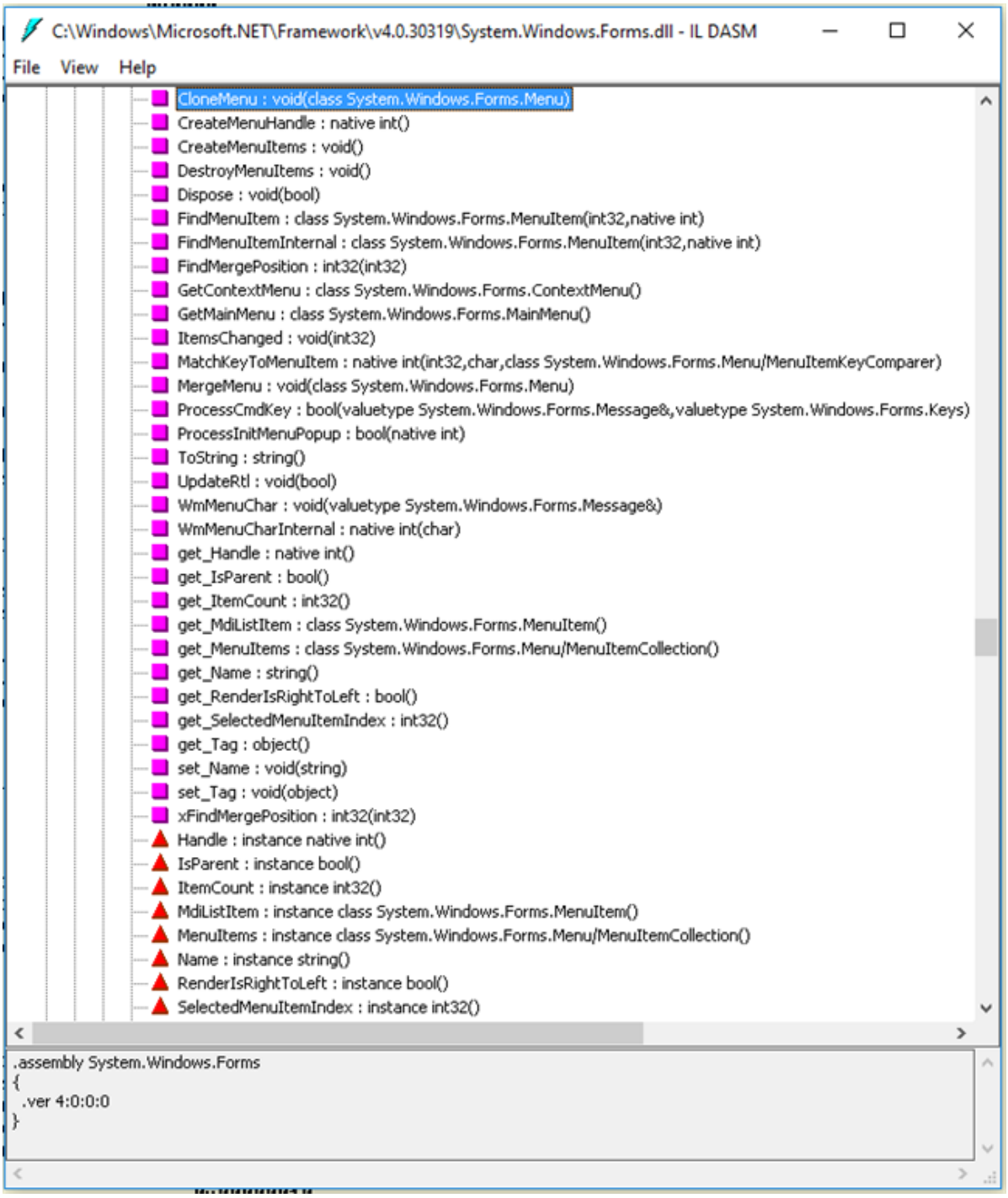
	Id	FName	LName	Password	City
▶	1	chetan	Nargund	12345	Bangalore
*	NULL	NULL	NULL	NULL	NULL

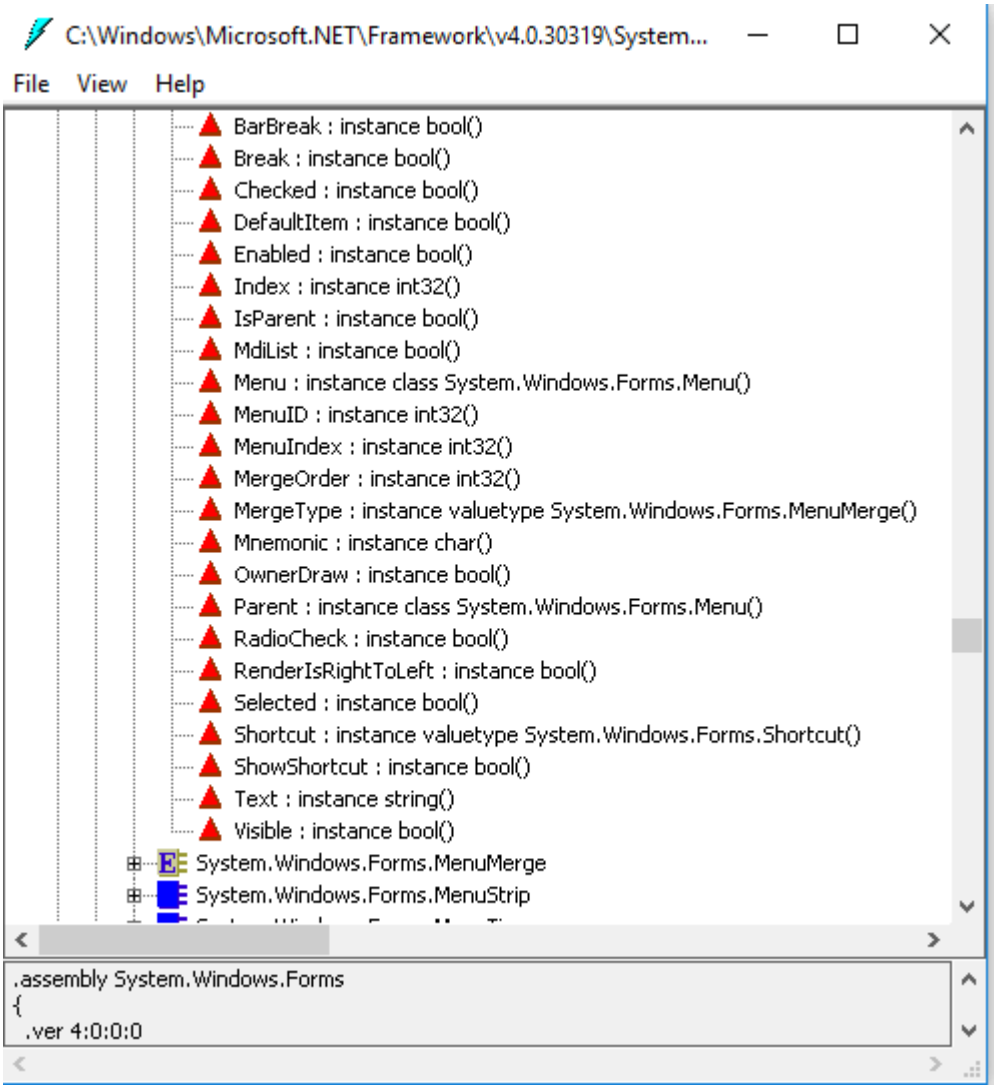
29. Demonstrate the use of Menu control in C#.

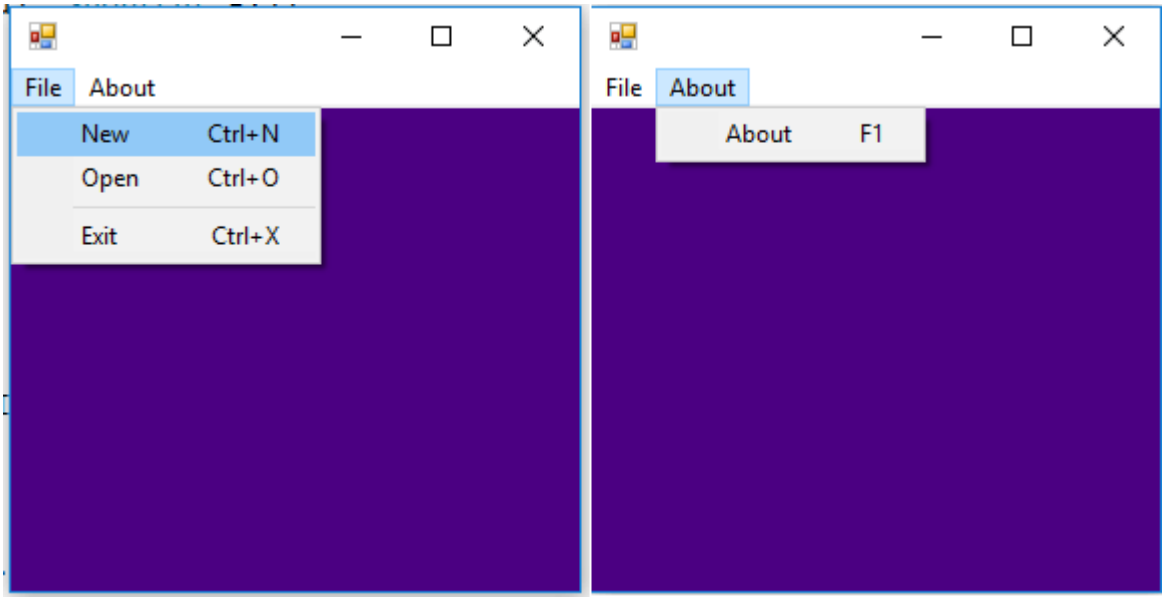
- **Program :-**

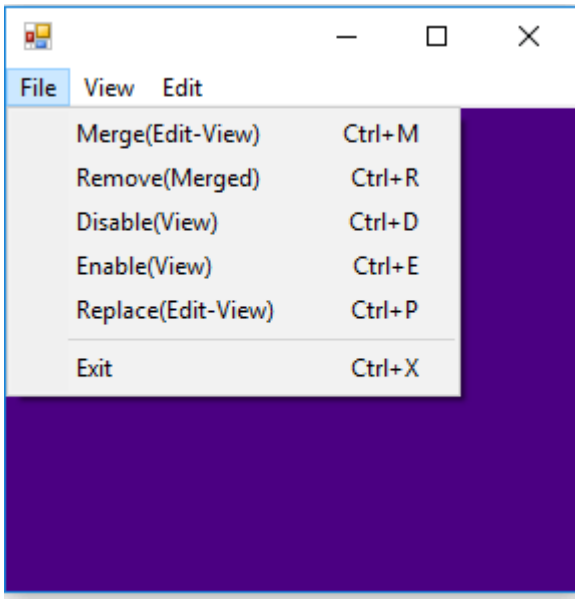
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace MenuTest
{
   public partial class MenuTest1 : Form
   {
      private MainMenu mainMenu;


      public MenuTest1()
      {
         InitializeComponent();
         mainMenu = new MainMenu();
         MenuItem File = mainMenu.MenuItems.Add("&File");
         File.MenuItems.Add(new MenuItem("&New"));
         File.MenuItems.Add(new MenuItem("&Open"));
         File.MenuItems.Add(new MenuItem("&Exit"));

         this.Menu = mainMenu;
         MenuItem About = mainMenu.MenuItems.Add("&About");
         About.MenuItems.Add(new MenuItem("&About"));
         this.Menu = mainMenu;
         mainMenu.GetForm().BackColor = Color.Indigo;
      }
   }
}
```

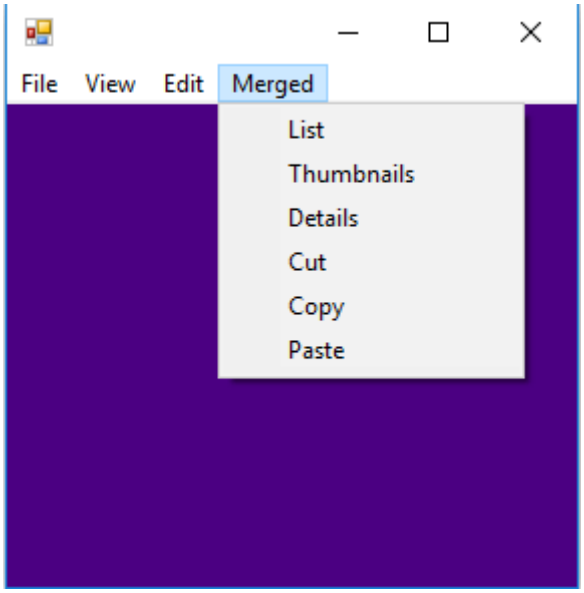
- **Output**

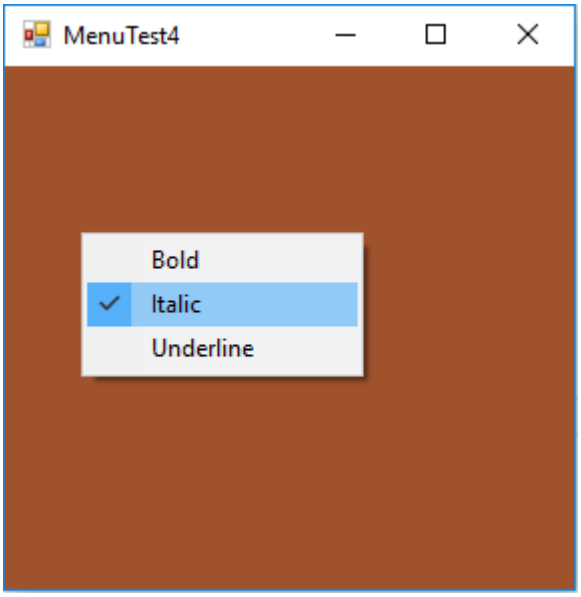

C:\Windows\Microsoft.NET\Framework\v4.0.30319\System.Windows.Forms.dll - IL DASM
File View Help
CloneMenu : void(class System.Windows.Forms.Menu)
CreateMenuHandle : native int()
CreateMenuItems : void()
DestroyMenuItems : void()
Dispose : void(bool)
FindMenuItem : class System.Windows.Forms.MenuItem(int32,native int)
FindMenuItemInternal : class System.Windows.Forms.MenuItem(int32,native int)
FindMergePosition : int32(int32)
GetContextMenu : class System.Windows.Forms.ContextMenu()
GetMainMenu : class System.Windows.Forms.MainMenu()
ItemsChanged : void(int32)
MatchKeyToMenuItem : native int(int32,char,class System.Windows.Forms.Menu/MenuItemKeyComparer)
MergeMenu : void(class System.Windows.Forms.Menu)
ProcessCmdKey : bool(valuetype System.Windows.Forms.Message&,valuetype System.Windows.Forms.Keys)
ProcessInitMenuPopup : bool(native int)
ToString : string()
UpdateRtl : void(bool)
WmMenuChar : void(valuetype System.Windows.Forms.Message&)
WmMenuCharInternal : native int(char)
get_Handle : native int()
get_IsParent : bool()
get_ItemCount : int32()
get_MdiListItem : class System.Windows.Forms.MenuItem()
get_MenuItems : class System.Windows.Forms.Menu/MenuItemCollection()
get_Name : string()
get_RenderIsRightToLeft : bool()
get_SelectedMenuItemIndex : int32()
get_Tag : object()
set_Name : void(string)
set_Tag : void(object)
xFindMergePosition : int32(int32)
Handle : instance native int()
IsParent : instance bool()
ItemCount : instance int32()
MdiListItem : instance class System.Windows.Forms.MenuItem()
MenuItems : instance class System.Windows.Forms.Menu/MenuItemCollection()
Name : instance string()
RenderIsRightToLeft : instance bool()
SelectedMenuItemIndex : instance int32()
.assembly System.Windows.Forms
{
.ver 4:0:0:0
}


C:\Windows\Microsoft.NET\Framework\v4.0.30319\System...
File View Help
BarBreak : instance bool()
Break : instance bool()
Checked : instance bool()
DefaultItem : instance bool()
Enabled : instance bool()
Index : instance int32()
IsParent : instance bool()
MdiList : instance bool()
Menu : instance class System.Windows.Forms.Menu()
MenuID : instance int32()
MenuIndex : instance int32()
MergeOrder : instance int32()
MergeType : instance valuetype System.Windows.Forms.MenuMerge()
Mnemonic : instance char()
OwnerDraw : instance bool()
Parent : instance class System.Windows.Forms.Menu()
RadioCheck : instance bool()
RenderIsRightToLeft : instance bool()
Selected : instance bool()
Shortcut : instance valuetype System.Windows.Forms.Shortcut()
ShowShortcut : instance bool()
Text : instance string()
Visible : instance bool()
System.Windows.Forms.MenuMerge
System.Windows.Forms.MenuStrip
.assembly System.Windows.Forms
{
.ver 4:0:0:0


File About
New Ctrl+N
Open Ctrl+O
Exit Ctrl+X
File About
About F1

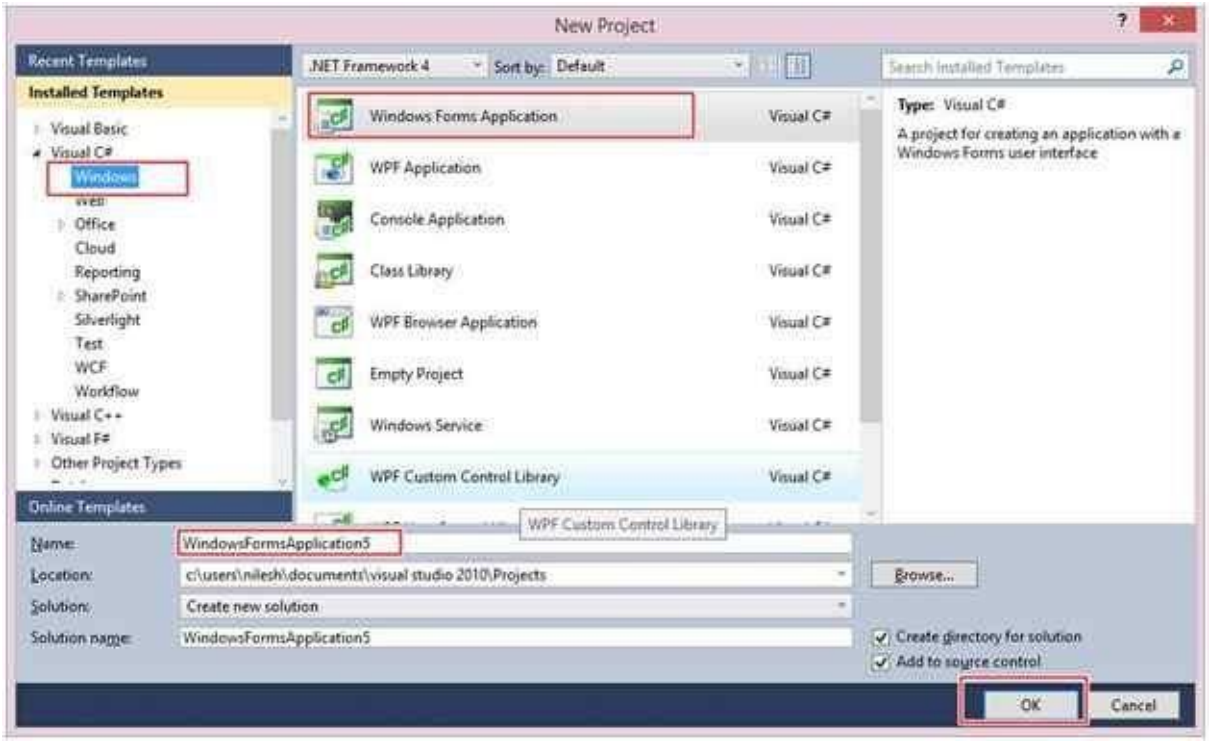

File
View
Edit
Merge(Edit-View) Ctrl+M
Remove(Merged) Ctrl+R
Disable(View) Ctrl+D
Enable(View) Ctrl+E
Replace(Edit-View) Ctrl+P
Exit Ctrl+X

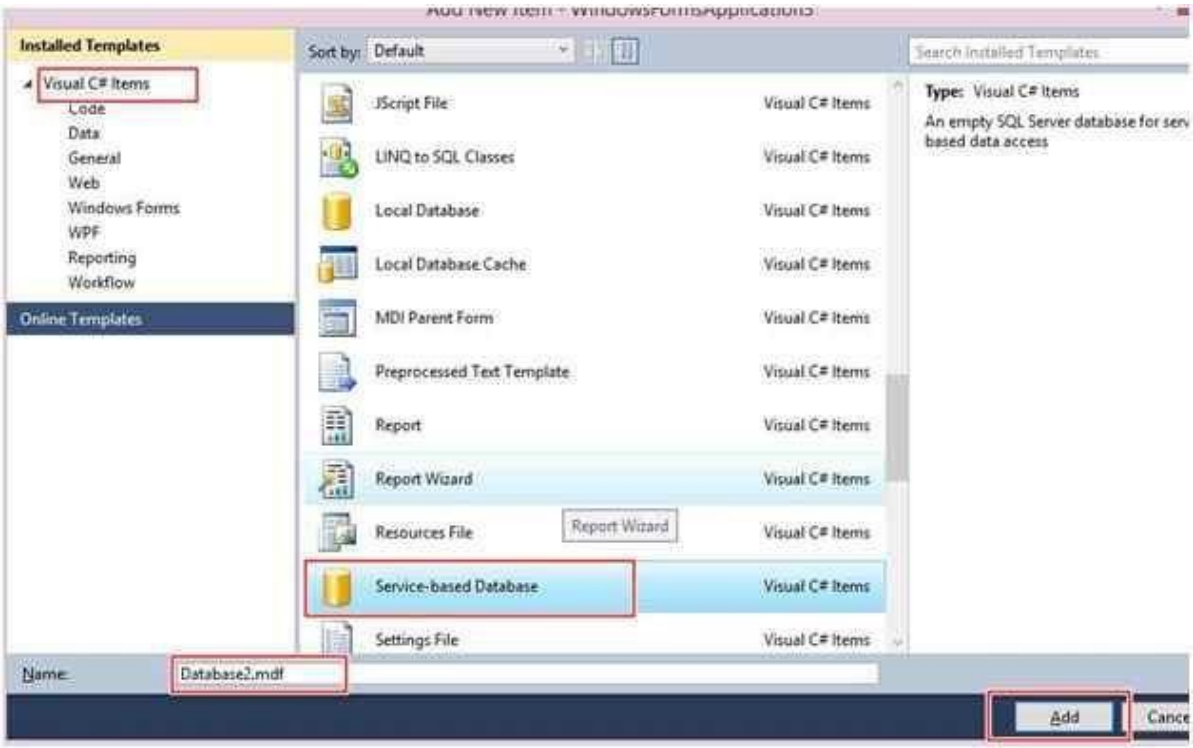

File
View
Edit
Merged
List
Thumbnails
Details
Cut
Copy
Paste


MenuTest4
Bold
Italic
Underline

30. Create a Window Application to generate the crystal report.

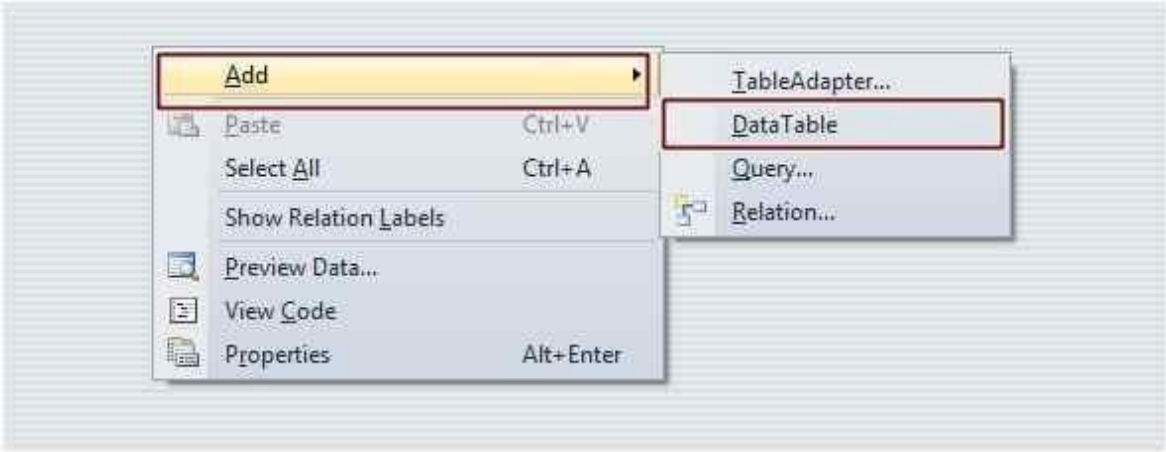
- **Program :-**

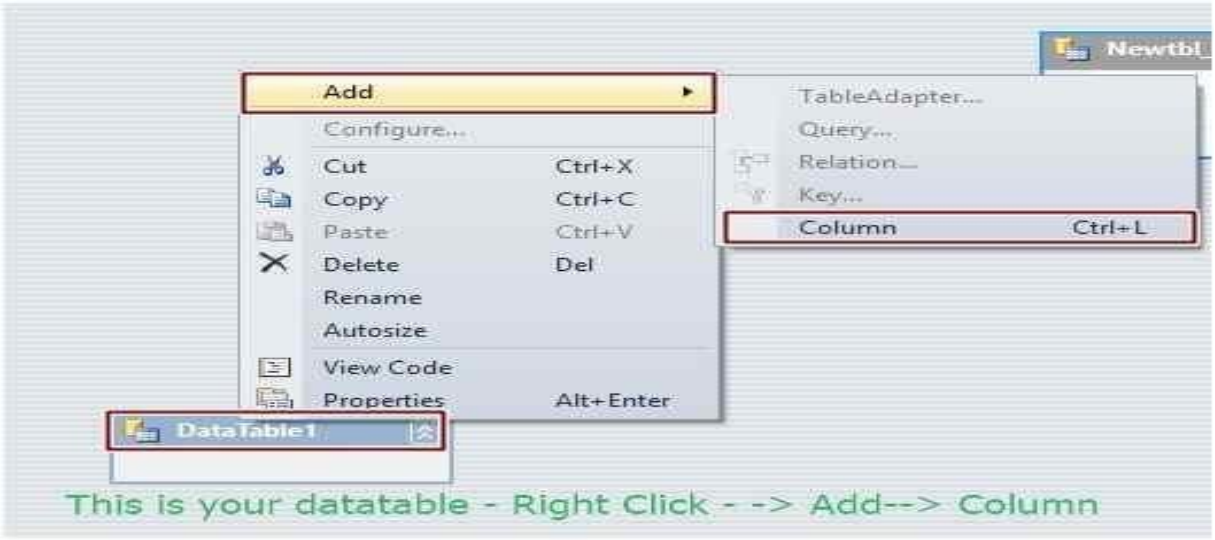




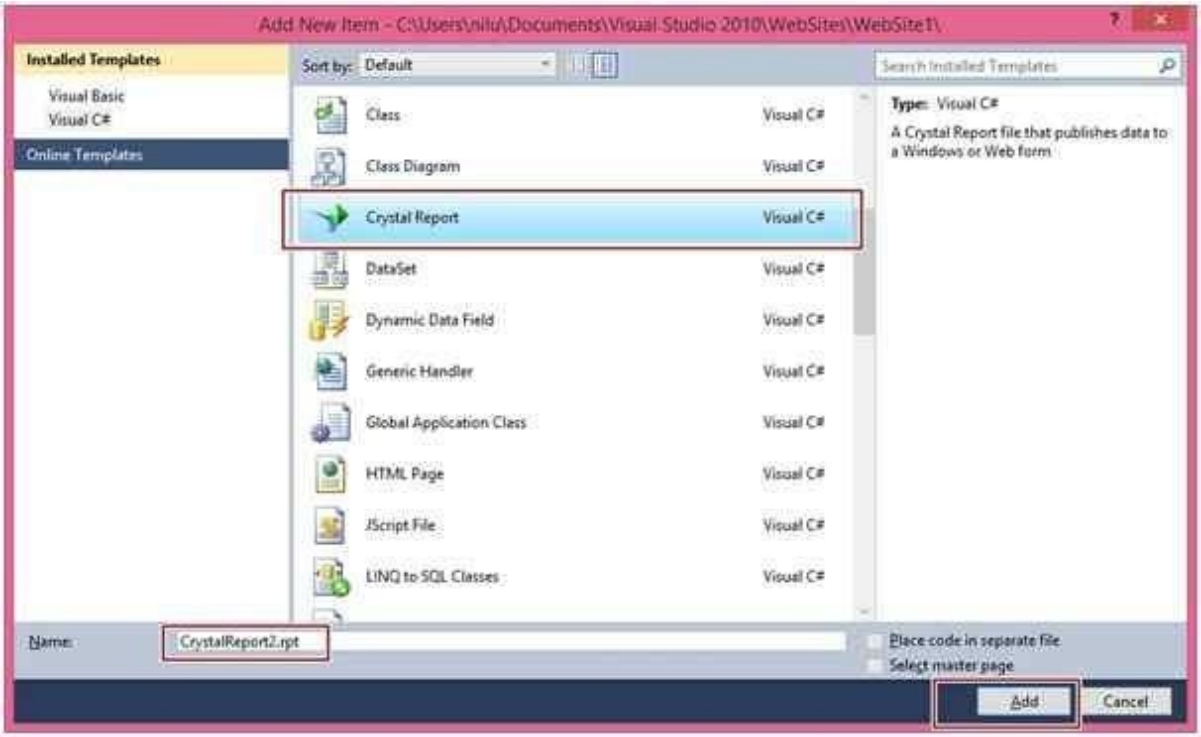
Column Name	Data Type	Allow Nulls
[product id]	int	☐
[product name]	varchar(50)	☑
quantity	varchar(50)	☑
[unit price]	varchar(50)	☑
		☐

```
ALTER PROCEDURE sp_select
    /*
    (
    @parameter1 int = 5,
    @parameter2 datatype OUTPUT
    )
    */
AS

    select * from tbl_data

        RETURN
```



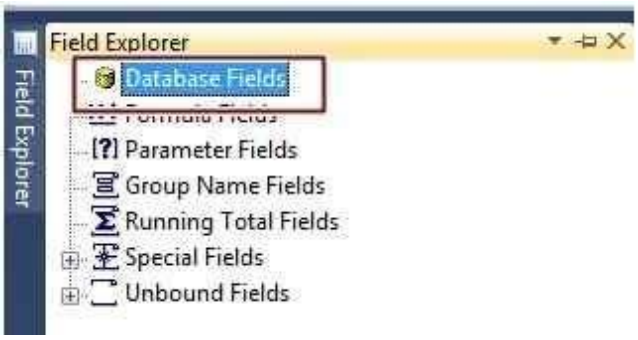


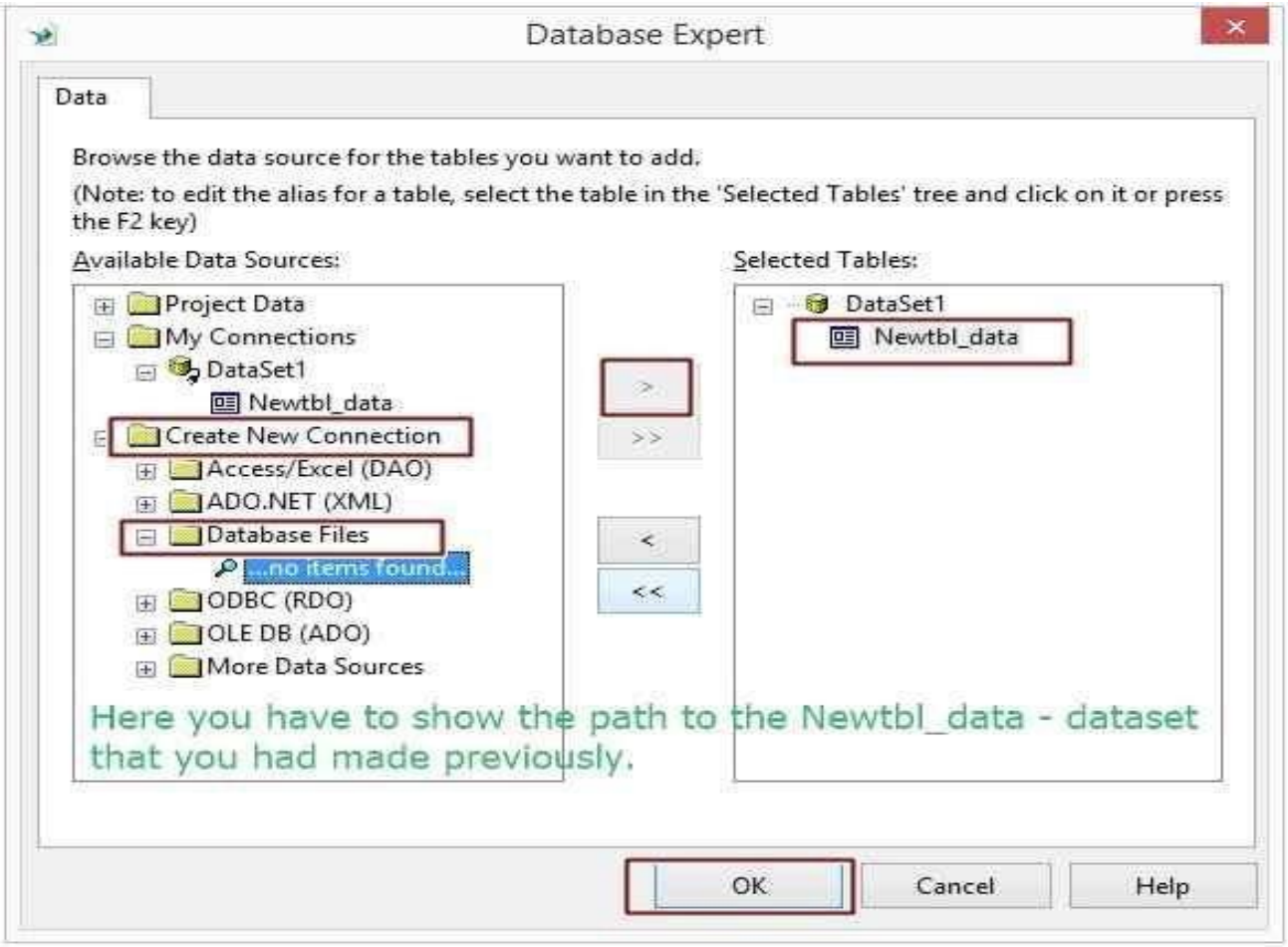

Newtbl_data
product id
product name
quantity
unit price

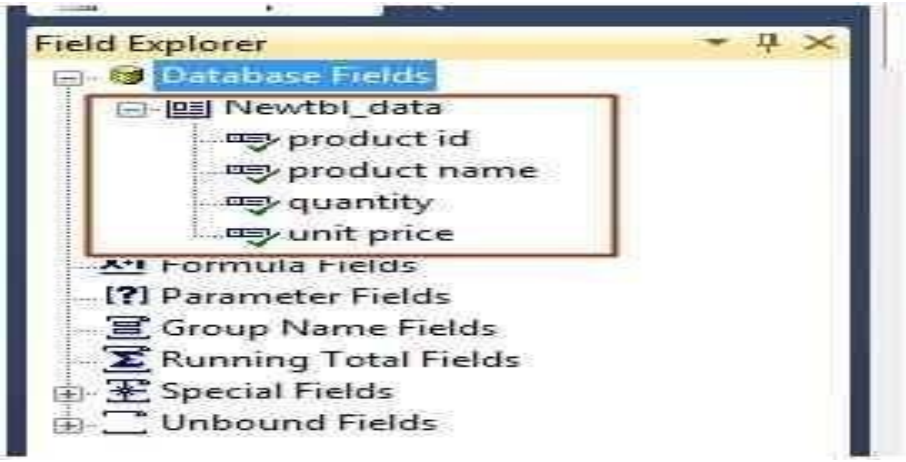

Add New Item - C:\Users\nilu\Documents\Visual Studio 2010\WebSites\WebSite1\
Installed Templates
Visual Basic
Visual C#
Online Templates
Sort by: Default
Search Installed Templates
Class
Class Diagram
Crystal Report
DataSet
Dynamic Data Field
Generic Handler
Global Application Class
HTML Page
JScript File
LINQ to SQL Classes
Visual C#
Type: Visual C#
A Crystal Report file that publishes data to a Windows or Web form
Name:
CrystalReport2.rpt
Place code in separate file
Select master page
Add
Cancel

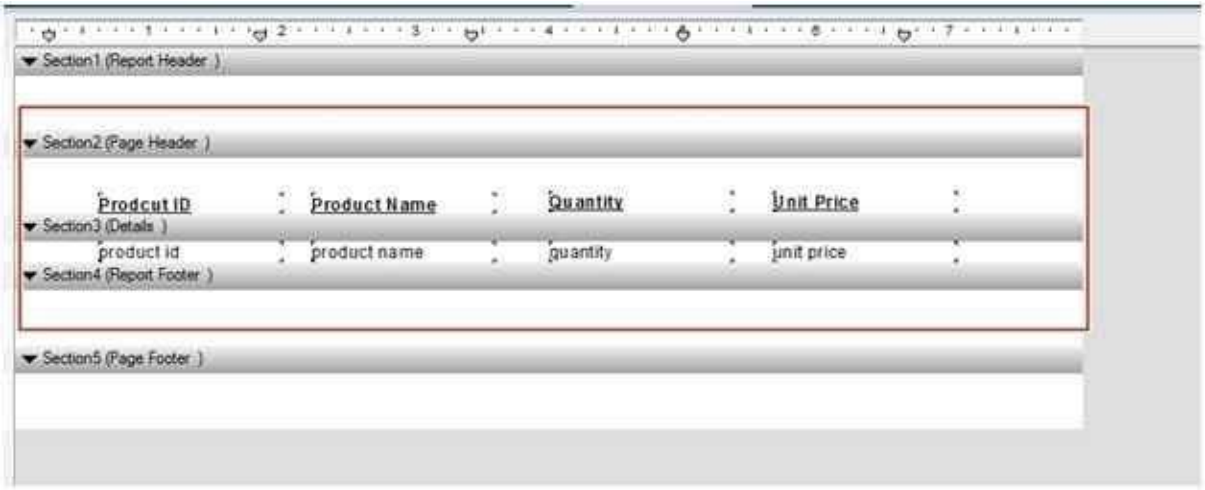

Crystal Reports Gallery
Create a New Crystal Report Document
Using the Report Wizard
As a Blank Report
From an Existing Report
Choose an Expert
Standard
Cross-Tab
Mail Label
Guides the creation of a typical report.
OK
Help

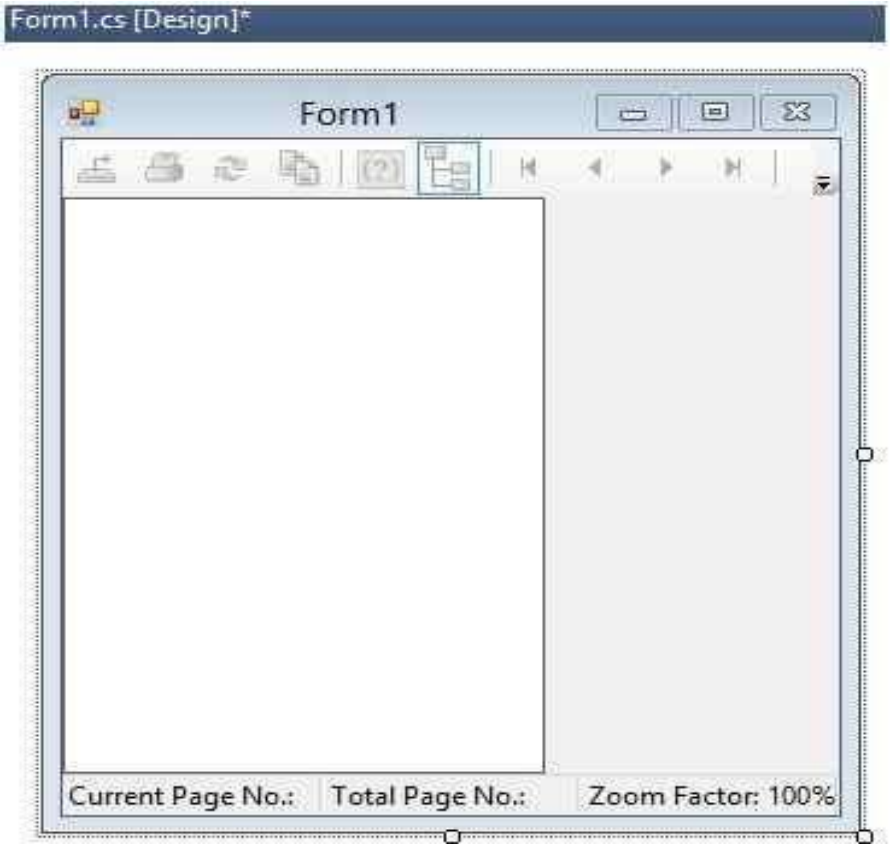

Section1 (Report Header)
Section2 (Page Header)
Section3 (Details)
Section4 (Report Footer)
Section5 (Page Footer)


Field Explorer
Database Fields
Parameter Fields
Group Name Fields
Running Total Fields
Special Fields
Unbound Fields


Database Expert
Data
Browse the data source for the tables you want to add.
(Note: to edit the alias for a table, select the table in the 'Selected Tables' tree and click on it or press the F2 key)
Available Data Sources:
Project Data
My Connections
DataSet1
Newtbl_data
Create New Connection
Access/Excel (DAO)
ADO.NET (XML)
Database Files
...no items found...
ODBC (RDO)
OLE DB (ADO)
More Data Sources
Selected Tables:
DataSet1
Newtbl_data
Here you have to show the path to the Newtbl_data - dataset that you had made previously.
OK
Cancel
Help

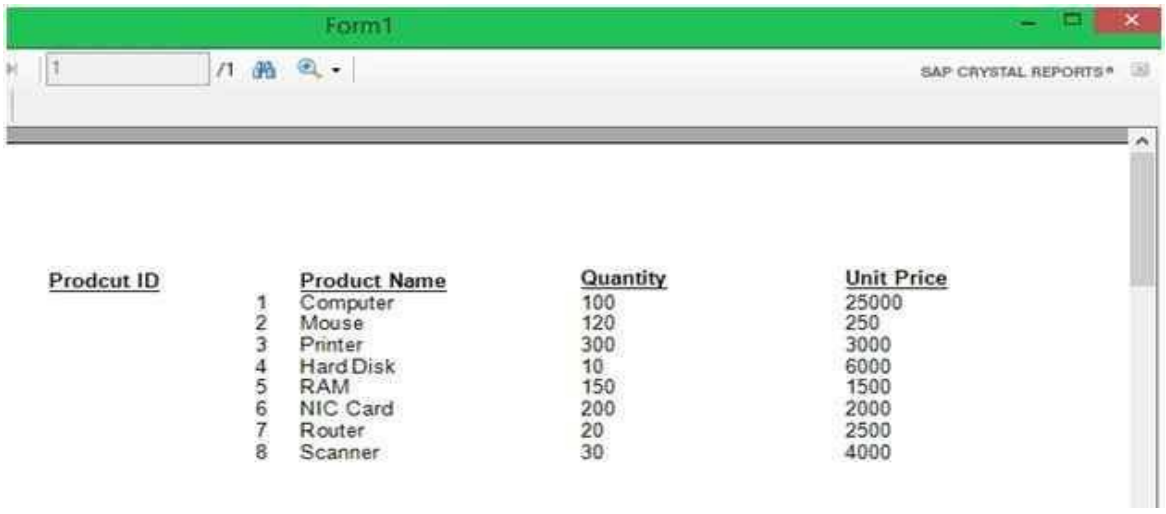

Field Explorer
Database Fields
Newtbl_data
product id
product name
quantity
unit price
Formula Fields
Parameter Fields
Group Name Fields
Running Total Fields
Special Fields
Unbound Fields


Section1 (Report Header)
Section2 (Page Header)
Prodcut ID
Product Name
Quantity
Unit Price
Section3 (Details)
product id
product name
quantity
unit price
Section4 (Report Footer)
Section5 (Page Footer)


Form1.cs [Design]*
Form1
Current Page No.:
Total Page No.:
Zoom Factor: 100%

```
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 System.Data.SqlClient;
using CrystalDecisions.CrystalReports.Engine;
namespace WindowsFormsApplication5
{
   public partial class Form1 : Form
   {
      SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPR
ESS;AttachDbFilename=|DataDirectory|\Database1.mdf;Integrated Secu
rity=True;User Instance=True");
      ReportDocument rprt = new ReportDocument();
      public Form1()
      {
         InitializeComponent();
      }
      private void Form1_Load(object sender, EventArgs e)
      {
         rprt.Load(@"C:\Users\Nilesh\Documents\visual studio 2010\Projects\

WindowsFormsApplication5\WindowsFormsApplication5\CrystalReport1.rp t");
         SqlCommand cmd = new SqlCommand("sp_select", con);
         cmd.CommandType = CommandType.StoredProcedure;
         SqlDataAdapter sda = new SqlDataAdapter(cmd);
         DataSet ds = new DataSet();
         sda.Fill(ds, "Newtbl_data");
         rprt.SetDataSource(ds);
         crystalReportViewer1.ReportSource = rprt;
      }
   }
}
```

- **Output :-**



Prodcut ID		Product Name	Quantity	Unit Price
	1	Computer	100	25000
	2	Mouse	120	250
	3	Printer	300	3000
	4	Hard Disk	10	6000
	5	RAM	150	1500
	6	NIC Card	200	2000
	7	Router	20	2500
	8	Scanner	30	4000