

GOVERNMENT POLYTECHNIC, AMRAVATI

(An Autonomous Institute of Government of Maharashtra)

NBA Accredited Institute

Certificate



Name of Department: Computer Science and Engineering.

This is to certify that **Mr. Ayush Shashikant Bulbule** Identity Code **19CM007** has completed the practical work of the course **FC3410 GUI APPLICATION DEVELOPMENT USING .NET** during the Academic year 2020-21.

Signature of the Teacher

Date:

who taught the examinee

Head of Department

Index

S.No	Name of Experiment	Date	Page	Remarks
1	Installation, set up and use of Visual Basic.NET IDE also study the different components of .NET Framework.	20-9-2020	3	
2	1 2. i) Write a simple program to display a welcome message using msgbox() ii) Develop a program to solve Arithmetic expression.	24-9-2020	6	
3	3. Develop a program to demonstrate use of If, If Else control structure in VB.NET. 2	26-9-2020	8	
4	4. Develop a program to demonstrate use of Select Case Control structure in VB.NET. 2	28-9-2020		
5	5. Develop a program to demonstrate use of While, Do While loop in VB.NET. 2	30-9-2020		
6	6. Develop a program to demonstrate use of For, For Each Next loop in VB.NET.	2-10-2020		
7	7. Develop a program using Textbox, Label, and Button.	4-10-2020		
8	8. Develop a program using Radio Button , Check Box	6-10-2020		
9	9. Develop a program using List box, Combo box.	15-10-2020		
10	10. Write a program using Picture box, Panel.	20-10-2020		
11	11. Write a program using Tab Control and Panel.	30-10-2020		
12	12. Write a program to demonstrate use of Sub Procedure and Parameterized Sub Procedure.	2-11-2020		
13	13. Write a program to demonstrate use of Function and Parameterized Function.	4-11-2020		
14	14. Develop a program to create class Access members of its class using its object.	5-11-2020		
15	15. Develop a program to demonstrate Overloading and Overriding Methods.	20-11-2020		
16	16. Design a Program to handle a run time error by using Exception Handling.	25-11-2020		
17	17. Use multiple table design example of employee and department. 4	27-11-2020		
18	18. Design registration form of college using Auto postbackproperty.	13-12-2020		
19	19. Simple application for following function: (1) Login (2) Surfing (3) Logout taking Into considerations (Application, Session, Server object, global .asa file and their events, methods and collection) and demonstration enabling and disabling of session.)	15-12-2020		
20	20. Creation of file, entry, reading data from a file	18-12-2020		

21	21. Using components create: (1) Advertisement (using Ad rotator) (2) Book example (using Next function) (3) find capabilities of browser (Browser object capabilities) 5	22-12-2020		
22	22. Create Online application (student, employee, product, shopping mall) using dataset, data reader. 6	24-12-2020		
23	23. Create Online application (student, employee, product, shopping mall) using data grid. 6	8-01-2021		
24	24. Bind the data to data grid using properties / templates	10-01-2021		

Practical no 1.

Aim: Installation, set up and use of Visual Basic.NET IDE also study the different components of .NET Framework.

Theory:

Microsoft visual studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, webpages, web services and mobile apps. Visual studio uses Microsoft software development platforms such as Windows API, Windows Store and Windows Forms. It can produce both native code and managed code. Visual studio supports 36 languages. The most basic editor of visual studio "the community edition" is available free of charge.

Steps to Install Visual Studio:

Step 1: Open the Browser and search for "Visual Studio Download" then go on the Microsoft's official Site.

Step 2: Select and download the Community Version (It is Free)

Step 3: Run the downloaded Visual Studio setup(.exe) file.

Step 4: Let it to install properly.

Step 5: After installation Select the packages required as per your work. (Here I am Using Visual Studio for .NET)

Step 6 : Create your project.

Step 7: Run your Code!! Congrats!! You have installed Visual Studio properly!!

1. COMPONENTS OF .NET FRAMEWORK: The two major components of .NET framework are the common language runtime and .NET framework class library.
2. The common language runtime CLR is the execution engine that handles running applications. It provides services like thread management, garbage collection etc.
3. The class library provides a set of APIs and types for common functionality. It is a collection of ready-made classes that can be used for commonly needed programming tasks.

4. .NET applications are written in the C#, F#, or visual basic programming language. Code is compiled into language- agnostic CIL. Compile code is store in assemblies-files with .exe file exetention.
5. When an app runs, the CLR takes the assembly and use just-intime compiler [JIT] to run it into machine code that can execute on the specific architecture of the computer it is running on.

USE OF VB.NET: VB.NET can be used to create a wide range of applications and components, including the following 1. Windows console mode application 2. Standard windows applications 3. Web (ASP.NET) applications 4. Windows control and windows control library. 5. Web services. 6. Web control and web control library

Practical no 2.

Aim: 1 2. i) Write a simple program to display a welcome message using msgbox() ii) Develop a program to solve Arithmetic expression.

Theory:

1. MESSAGE BOX:

- A message box is a special dialog box used to display a piece of information to the user. As opposed to a regular form, the user cannot type anything in the dialog box. To support message boxes, the visual basic language provides a function named MsgBox.

- Process to create MsgBox with title: To create a msgbox with title using VB.NET follow this steps :

step-1: create a message box with MsgBox function

Step-2: specify the message display in the message box [Prompt := Promptstring]

Step-3: specify the message box title [Title := TitleString]

2. ARITHMETIC EXPRESSION:

- An arithmetic expression is one that can be evaluated to given numeric value, and combines variables, arithmetic operators and keywords. Note that all variables used in an expression that are to the right of an assignment operator must be declare and initialized before they can be used.

- Syntax: variable = operand operator operand Example: a*b

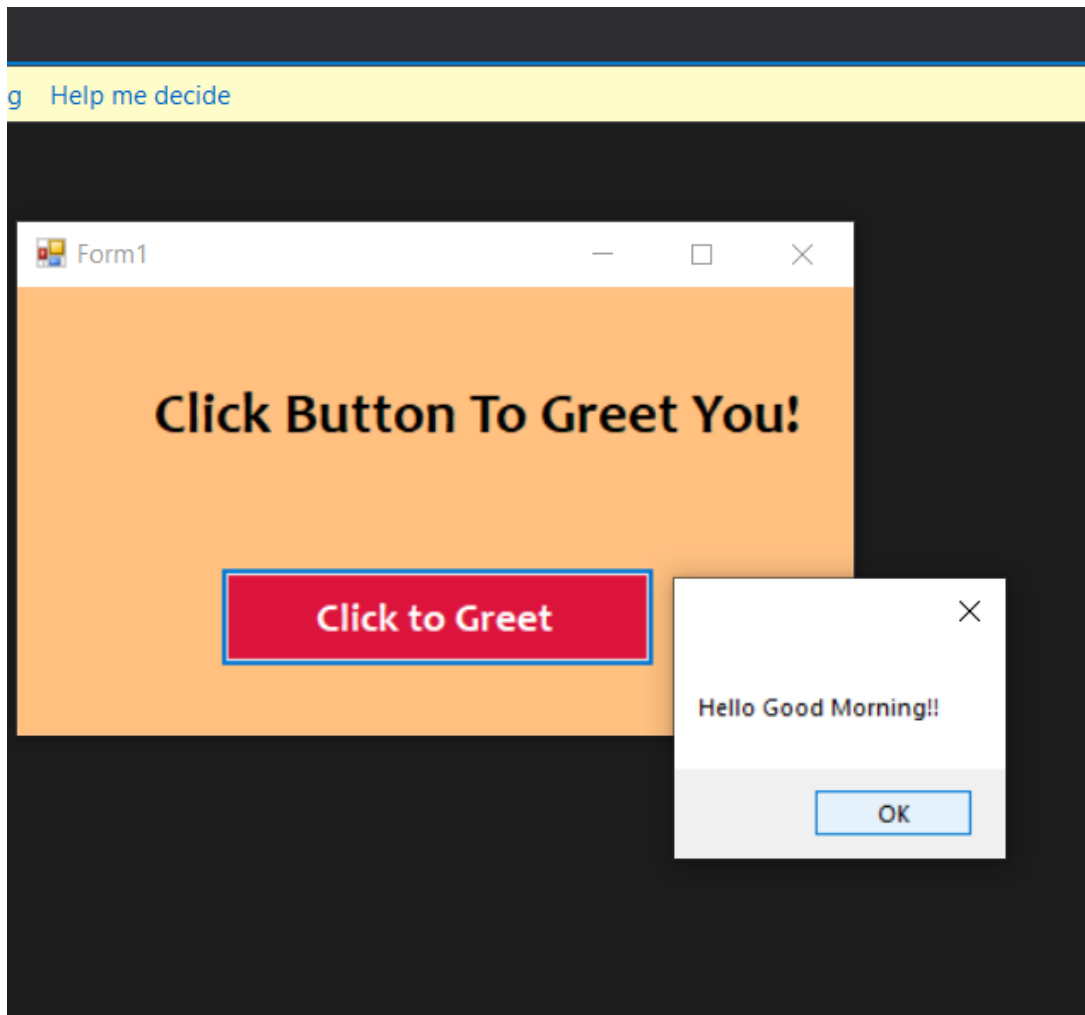
Program:

Message Box Code:

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs)
        Handles Button1.Click
            MessageBox.Show("Hello Good Morning!!")

    End Sub
End Class
```

Output:



2)Arithmetic Expression:

VB Code:

```
Public Class Form1
```

```
    Private Sub CloseBtn_Click(sender As Object, e As EventArgs)  
Handles CloseBtn.Click  
        Me.Close()  
    End Sub
```

```
    Private Sub ClearBtn_Click(sender As Object, e As EventArgs)  
Handles ClearBtn.Click  
        txtNum1.Text = ""  
        txtNum2.Text = ""  
        lblresult.Text = ""
```

```
End Sub
```

```

Private Sub Clear()
    txtNum1.Text = ""
    txtNum2.Text = ""
End Sub

Private Sub Subtract_Click(sender As Object, e As EventArgs)
Handles Subtract.Click
    Dim n1, n2 As Integer

    n1 = Val(txtNum1.Text)
    n2 = Val(txtNum2.Text)
    lblresult.Text = n1 - n2
    Clear()
End Sub

Private Sub Multiply_Click(sender As Object, e As EventArgs)
Handles Multiply.Click
    Dim n1, n2 As Integer

    n1 = Val(txtNum1.Text)
    n2 = Val(txtNum2.Text)
    lblresult.Text = n1 * n2
    Clear()
End Sub

Private Sub Divide_Click(sender As Object, e As EventArgs)
Handles Divide.Click
    Dim n1, n2 As Integer

    n1 = Val(txtNum1.Text)
    n2 = Val(txtNum2.Text)
    lblresult.Text = n1 / n2
    Clear()
End Sub

Private Sub Addition_Click(sender As Object, e As EventArgs)
Handles Addition.Click
    Dim n1, n2 As Integer

    n1 = Val(txtNum1.Text)
    n2 = Val(txtNum2.Text)
    lblresult.Text = n1 + n2
    Clear()
End Sub

End Class

```


Output:

The image shows a screenshot of a Java Swing window titled "Simple Calculator". The window has a yellow background and a title bar with standard Windows window controls (minimize, maximize, close). The title bar text is "Simple Calculator".

Inside the window, the text "Simple Calculator" is displayed in a large, bold, red font at the top center.

Below the title, there are two input fields for numbers. The first is labeled "Number 1 :" and contains the value "35". The second is labeled "Number 2 :" and contains the value "57".

Below the number inputs, there is a row of four buttons for operators: "+", "-", "×", and "÷". The text "Operator :" is to the left of these buttons.

Below the operator buttons, there is a label "Result :" followed by the value "92" in a large, bold, red font.

At the bottom of the window, there are two buttons: "Clear" and "Close".

Practical no 3.

Aim: Develop a program to demonstrate use of If, If Else control structure in VB.NET.

Theory:

An **If** statement can be followed by an optional **Else** statement, which executes when the Boolean expression is false.

Syntax:

```
If(boolean_expression)Then
    'statement(s) will execute if the Boolean expression is true
Else
    'statement(s) will execute if the Boolean expression is false
End If
```

Program:

```
Public Class Form1
```

```
    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Dim age As Integer
```

```
        age = txtage.Text
```

```
        If age >= 18 Then
            result_txt.Text = "You can Vote!!"
        Else
            result_txt.Text = "You cannot Vote!"
        End If
```

```
    End Sub
```

```
End Class
```

Output:

Form1

Detect you can vote or not?

Enter your age:

25

Detect

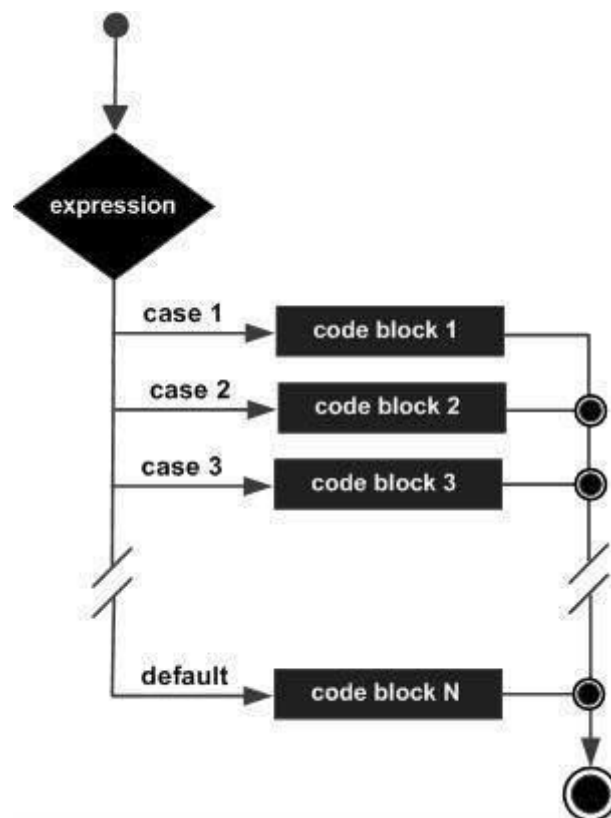
Result: You can Vote!!

Practical no 4.

Aim: Develop a program to demonstrate use of Select Case Control structure in VB.NET

Theory:

A **Select Case** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each select case.



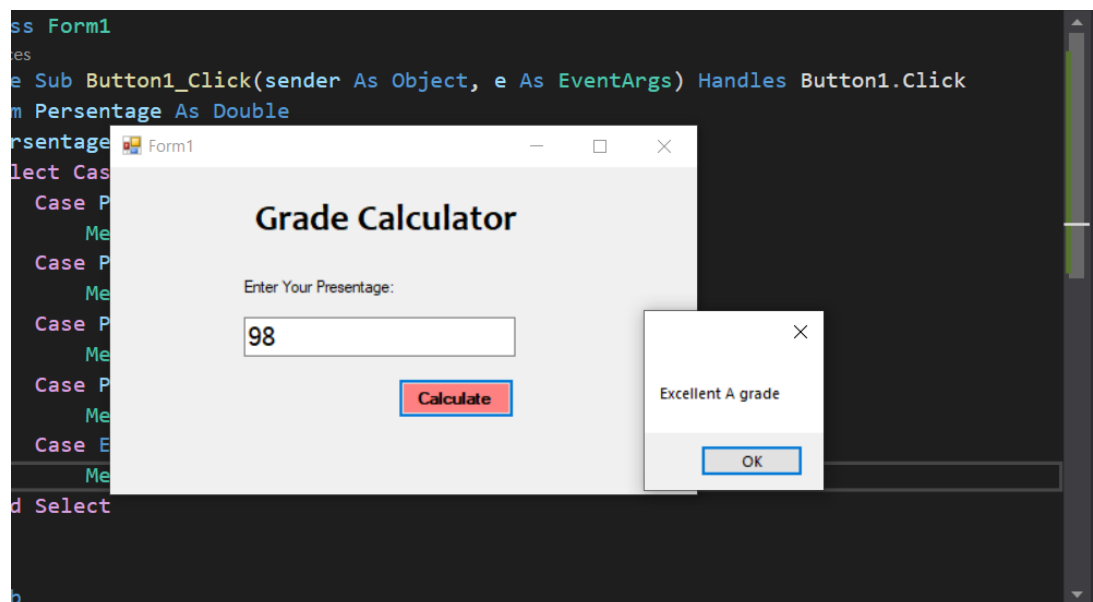
Program:

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Dim Percentage As Double
        Percentage = TextBox1.Text
        Select Case Percentage
            Case Percentage > 90
                MessageBox.Show("Excellent A grade")
            Case Percentage > 80
                MessageBox.Show("Best B grade")
```

```
Case Percentage > 65
    MessageBox.Show("Good C grade")
Case Percentage > 45
    MessageBox.Show("Needs Improvement D
grade")
Case Else
    MessageBox.Show("Try Again Fail")
End Select

End Sub
End Class
```

Output:

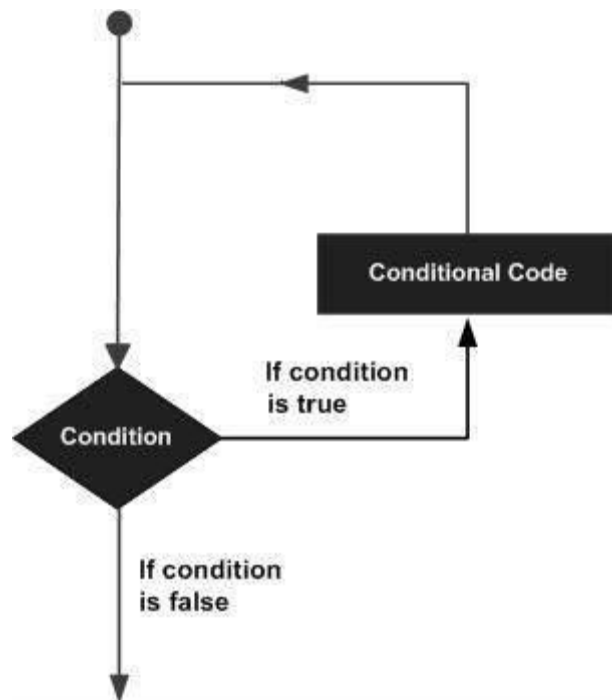


Practical no 5.

Aim: Develop a program to demonstrate use of While, Do While loop in VB.NET

Theory:

A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages –



Program:

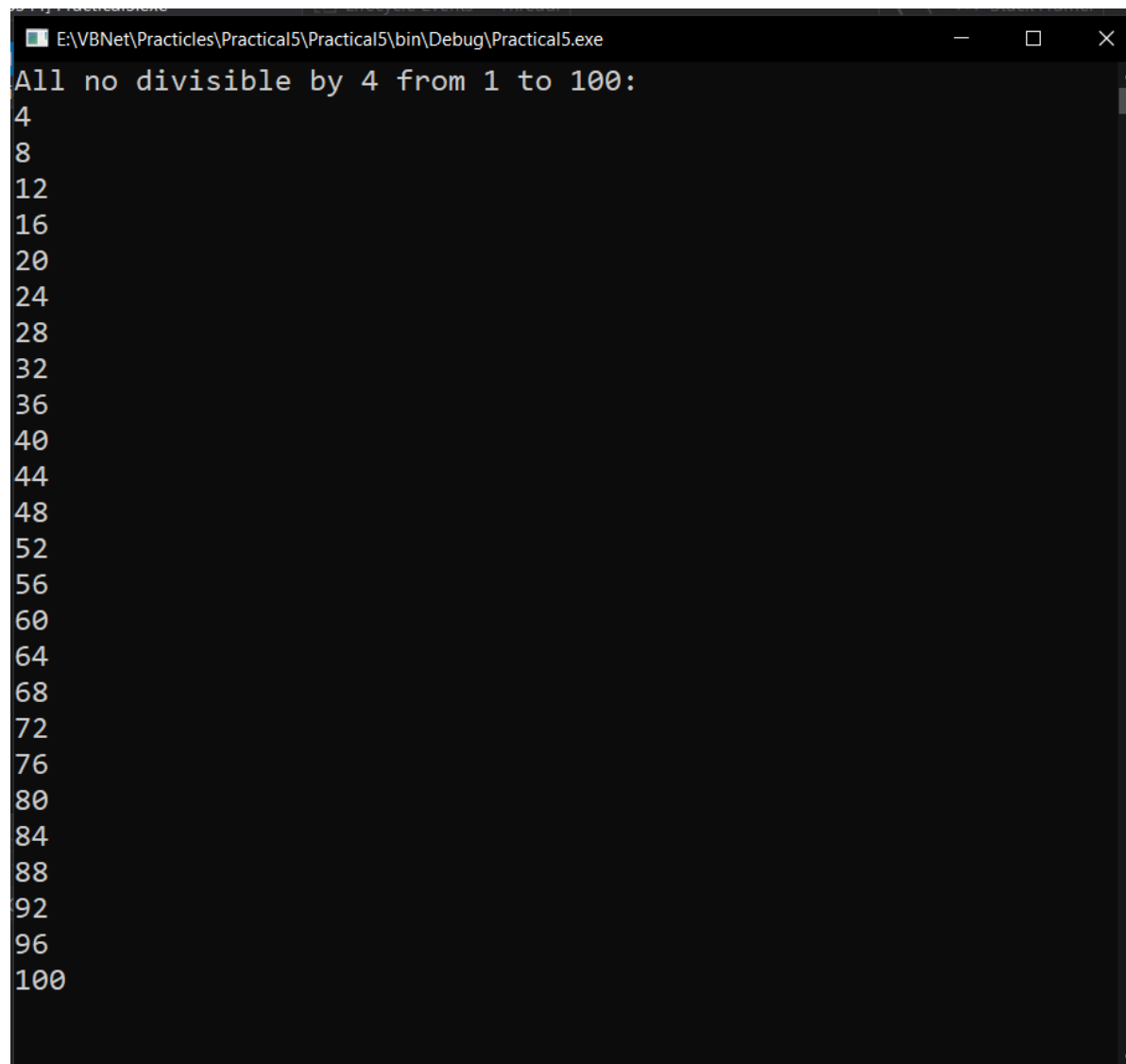
```
Module Module1
    'Develop a program to demonstrate use of While, Do
    While loop in VB.NET
    Sub Main()
        Dim n As Integer
        n = 1
        Console.WriteLine("All no divisible by 4 from 1 to
100: ")
        While (n <= 100)
            If (n Mod 4 = 0) Then
                Console.WriteLine(n)
            End If
            n += 1
        End While
    End Sub
End Module
```

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

Output:



```
E:\VBNet\Practicles\Practical5\Practical5\bin\Debug\Practical5.exe
All no divisible by 4 from 1 to 100:
4
8
12
16
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
80
84
88
92
96
100
```

Practical no 6.

Aim: Develop a program to demonstrate use of For, For Each Next loop in VB.NET.

Theory:

It repeats a group of statements for each element in a collection. This loop is used for accessing and manipulating all elements in an array or a VB.Net collection.

Program:

1) For Loop

```
Module Module1
```

```
    'Develop a program to demonstrate use of For, For Each  
    Next loop in VB.NET.
```

```
    Sub Main()
```

```
        Dim n As Integer
```

```
        Console.WriteLine("Printing all Even Numbers upto  
50: ")
```

```
        For n = 2 To 50 Step 2
```

```
            Console.WriteLine(n)
```

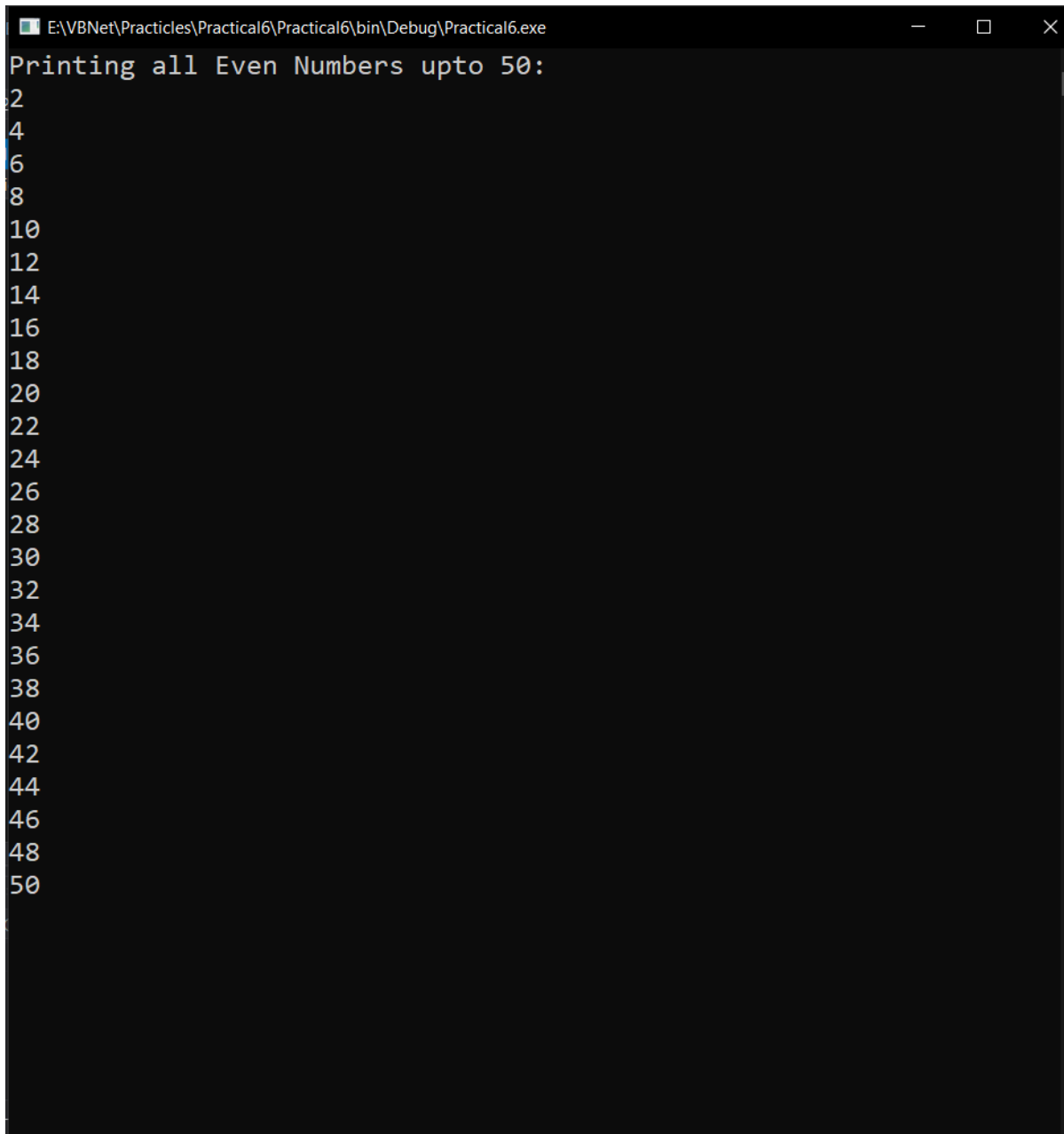
```
        Next
```

```
        Console.ReadLine()
```

```
    End Sub
```

```
End Module
```


Output:



```
E:\VBNet\Practicles\Practical6\Practical6\bin\Debug\Practical6.exe
Printing all Even Numbers upto 50:
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
```

2) For Next Loop

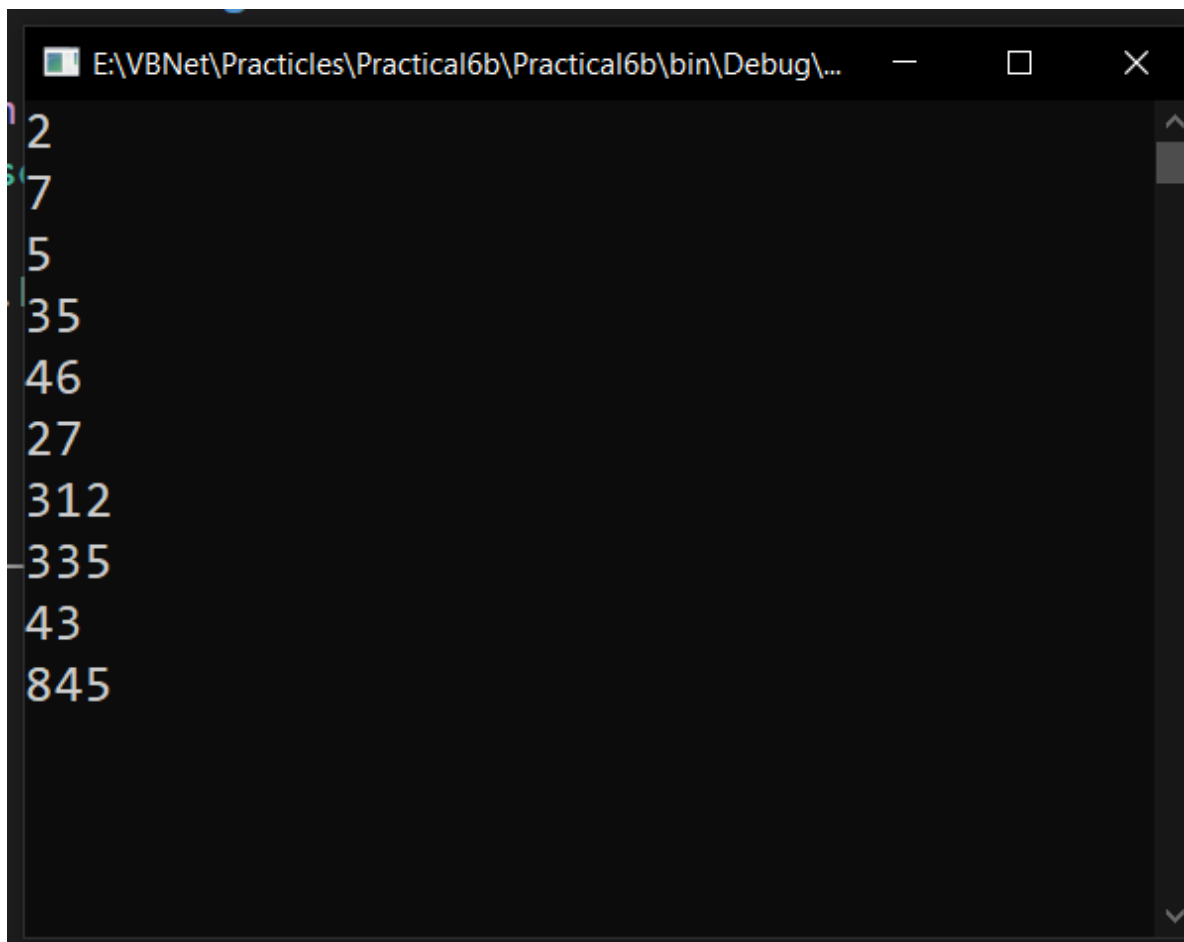
Module Module1

```
Sub Main()
    Dim Array() As Integer = {2, 7, 5, 35, 46, 27,
312, 335, 43, 845}
    Dim item As Integer
```

```
For Each item In Array
    Console.WriteLine(item)
Next
Console.ReadLine()
```

```
End Sub
```

```
End Module
```



```
E:\VBNet\Practicles\Practical6b\Practical6b\bin\Debug\...
2
7
5
35
46
27
312
335
43
845
```

Practical no 7.

Aim: Develop a program using Textbox, Label, and Button.

Theory:

1. A **label** is meant to be used beside a text box to make a user understand what is to be entered in that text box where as a text box is used normally for user input.
2. The contents of a **label** is not to be directly modified by a user where as the contents of a text box is for the user to modify.

Program:

```
Public Class Form1
    'Develop a program using Textbox, Label, and Button.

    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Dim sum, n1, n2 As Double
        n1 = txt_num.Text
        n2 = txt_num2.Text
        sum = n1 + n2
        label_res.Text = "Addition is: " + sum.ToString()

    End Sub

    Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
        txt_num.Text = Nothing
        txt_num2.Text = Nothing
    End Sub
End Class
```

Output:

Form1

Addition Calcuate

Number 1

Number 2

Addition is: 63

Practical no 8.

Aim: Develop a program using Radio Button , Check Box

Theory:

Quick sort is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays. A large array is partitioned into two arrays one of which holds values smaller than the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value.

Quicksort partitions an array and then calls itself recursively twice to sort the two resulting subarrays. This algorithm is quite efficient for large-sized data sets as its average and worst-case complexity are $O(n^2)$, respectively.

Algorithm:

Step 1 : Start.

Step 2 : Enter the number of elements.

Step 3 : Enter the number.

Step 4 : Pick an element from the array, this element is called a pivot element.

Step 5 : Divide the unsorted array of elements in two arrays with values less than the pivot come in the first sub array, while all elements with values greater than the pivot come in the second sub-array (equal values can go either way). This step is called the partition operation.

Step 6 : Recursively repeat the step 5(until the sub-arrays are sorted) to the sub-array of elements with smaller values and separately to the sub-array of elements with greater values.

Step 7 : Print the sorted elements.

Step 8 : Stop.

Program:

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Dim topping, flavour As String

        If chocolate.Checked Then
            topping = " Samosa"
```

```
ElseIf nutmeg.Checked Then
    topping += " Dosa"
ElseIf cream.Checked Then
    topping += " Fafda"
ElseIf icecream.Checked Then
    topping += " Jalebi"
End If
```

```
If cappu.Checked Then
    flavour = "Burger"
ElseIf late.Checked Then
    flavour = "Pizza"
ElseIf espre.Checked Then
    flavour = "Pasta"
ElseIf mocha.Checked Then
    flavour = "Noodles"
End If
```

```
    ordrSumm.Text = "Your order is " + flavour + " and  
Toppings are " + topping
```

```
End Sub  
End Class
```

Output:

Form1

Your Refreshment order

Customize your Order

Breakfast

- ☐ Samosa
- ☐ Dosa
- ☐ Fafda
- ☒ Jalebi

FastFood

- ☒ Pizza
- ☐ Burger
- ☐ Pasta
- ☐ Noodles

Your Order:

Your order is Pizza and
Toppings are Jalebi

Order

Practical no 9.

Aim: Develop a program using List box, Combo box.

Theory:

The list box displays the items all at once in a text area whilst combo box displays only one item initially and the user needs to click on the handle of the combo box to view the items in a drop-down list.

Program:

```
Public Class Form1
```

```
    Private Sub add_btn_Click(sender As Object, e As  
EventArgs) Handles add_btn.Click  
        Dim item As String = text_item.Text
```

```
        items_list.Items.Add(item)
```

```
    End Sub
```

```
    Private Sub ComboBox1_SelectedIndexChanged(sender As  
Object, e As EventArgs) Handles  
ComboBox1.SelectedIndexChanged  
        items_list.Items.Add(ComboBox1.Text)
```

```
    End Sub
```

```
End Class
```

Output:

Form1

— □ ×

List all Your Subjects!!

Enter an subject to add VB.net

DefaultItems ▾

Add item

VB.net

Practical no 10.

Aim: Write a program using Picture box, Panel

Theory:

The PictureBox control is used for displaying images on the form. The Image property of the control allows you to set an image both at design time or at run time.

Let's create a picture box by dragging a PictureBox control from the Toolbox and dropping it on the form.

Program:

```
Public Class Form1
    Private Sub TextBox1_TextChanged(sender As Object, e
As EventArgs) Handles TextBox1.TextChanged,
    TextBox2.TextChanged

        End Sub

    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click

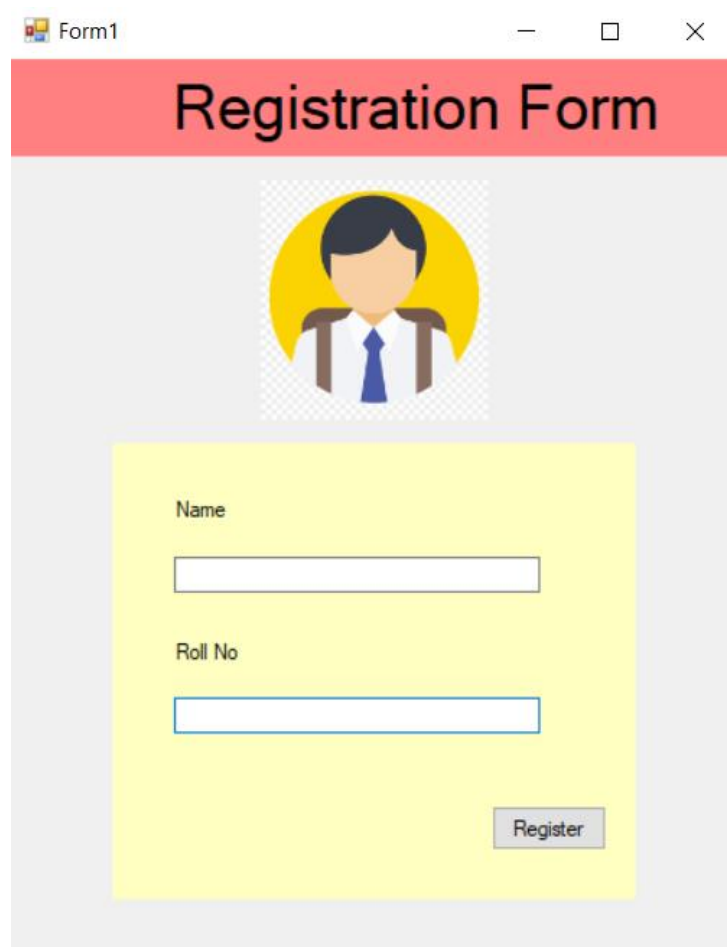
        MessageBox.Show("Registered Successfully. This
Program uses Pane and PictureBox")

    End Sub

    Private Sub PictureBox1_Click(sender As Object, e As
EventArgs) Handles PictureBox1.Click

    End Sub
End Class
```

Output:



The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area has a light gray background. At the top, there is a red header bar with the text "Registration Form" in white. Below the header, there is a placeholder for a profile picture, represented by a yellow circle with a black and white icon of a person in a suit. Below the profile picture, there is a yellow rectangular area containing two text input fields. The first field is labeled "Name" and the second field is labeled "Roll No". Both fields are empty. At the bottom right of the yellow area, there is a gray button labeled "Register".

Practical no 11.

Aim: Write a program using Tab Control and Panel.

Theory:

Windows Forms Tab Control displays multiple tabs, like dividers in a notebook or labels in a set of folders in a filing cabinet. The tabs can contain pictures and other controls. You can use the tab control to produce the kind of multiple-page dialog box that appears many places in the Windows operating system, such as the Control Panel Display Properties. Additionally, the Tab Control can be used to create property pages, which are used to set a group of related properties.

Program:

```
Public Class Form1
    Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
        PanelFrm.Show()
        Me.Hide()
    End Sub

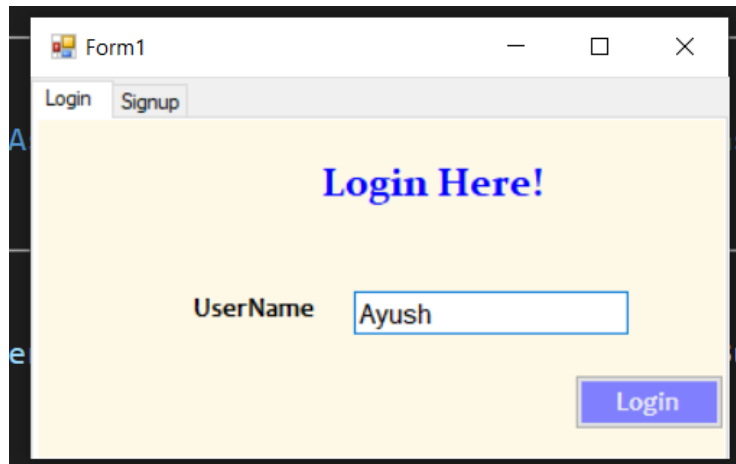
    Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        PanelFrm.Show()
        Me.Hide()
    End Sub
End Class
```

Output:

LoginTab



Form1

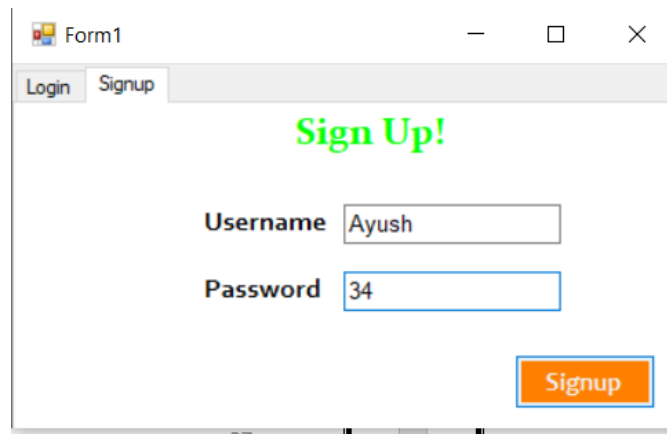
Login Signup

Login Here!

UserName Ayush

Login

Sign up tab



Form1

Login Signup

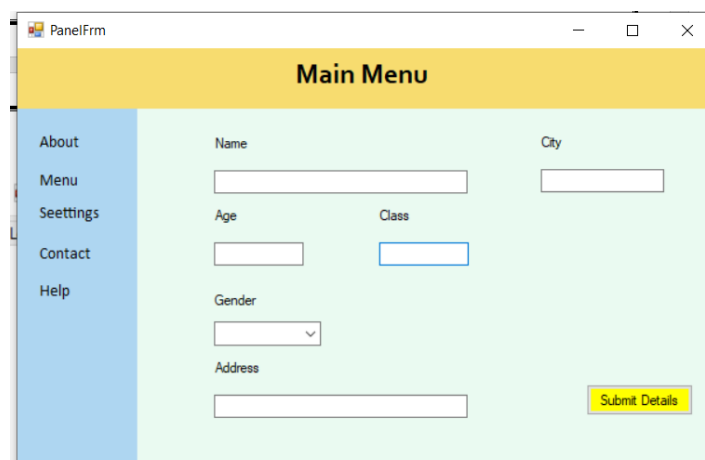
Sign Up!

Username Ayush

Password 34

Signup

Panel Form:



PanelFrm

Main Menu

About
Menu
Seettings
Contact
Help

Name City

Age Class

Gender

Address

Submit Details

Practical no 12.

Aim Write a program to demonstrate use of Sub Procedure and Parameterized Sub Procedure

Theory:

The **Sub** statement is used to declare the name, parameter and the body of a sub procedure. The syntax for the Sub statement is

```
[Modifiers] Sub SubName [(ParameterList)]  
    [Statements]  
End Sub
```

Where,

Modifiers – specify the access level of the procedure; possible values are
- Public, Private, Protected, Friend, Protected Friend and information regarding overloading, overriding, sharing, and shadowing.

SubName – indicates the name of the Sub

SParameterList – specifies the list of the parameters

Program:

```
Imports System.Math  
Public Class Form1  
    Private Sub Button1_Click(sender As Object, e As  
EventArgs) Handles Button1.Click  
        convertci()  
    End Sub  
  
    Private Sub convertci()  
  
        Dim tempCelci As Double  
        Dim tempFahrenheit As Integer  
  
        If txtCelci.Text <> "" Then  
            tempCelci = txtCelci.Text  
            tempFahrenheit = (tempCelci * 9 / 5) + 32  
            txtFahrenheit.Text = tempFahrenheit  
        End If  
    End Sub  
End Class
```

```
labelResult.Text = tempFahrenh
```

```
ElseIf txtFern.Text <> "" Then  
    tempFahrenh = txtFern.Text  
    tempCelci = (tempFahrenh - 32) * 5 / 9  
    txtCelci.Text = tempCelci  
    labelResult.Text = tempCelci
```

```
End If
```

```
End Sub  
End Class
```

Output:

Form1

Compound Intrest Calculator

Principal Amount : 2000

Intrest Rate : 10

Total No of Years: 35

Calculate

Compound Intrest: 56204.8736961287

Practical no 13.

Aim: Write a program to demonstrate use of Function and Parameterized Function.

Theory:

A function is a block of visual basic statements inside functions and End function statements. Function returns values. There are two basic types of functions Built in functions and user define functions. The built-in functions are part of the visual basic language.

Syntax:

```
[modifiers] Function Function_Name [Parameter list] as  
Return type    [statements]
```

End Function.

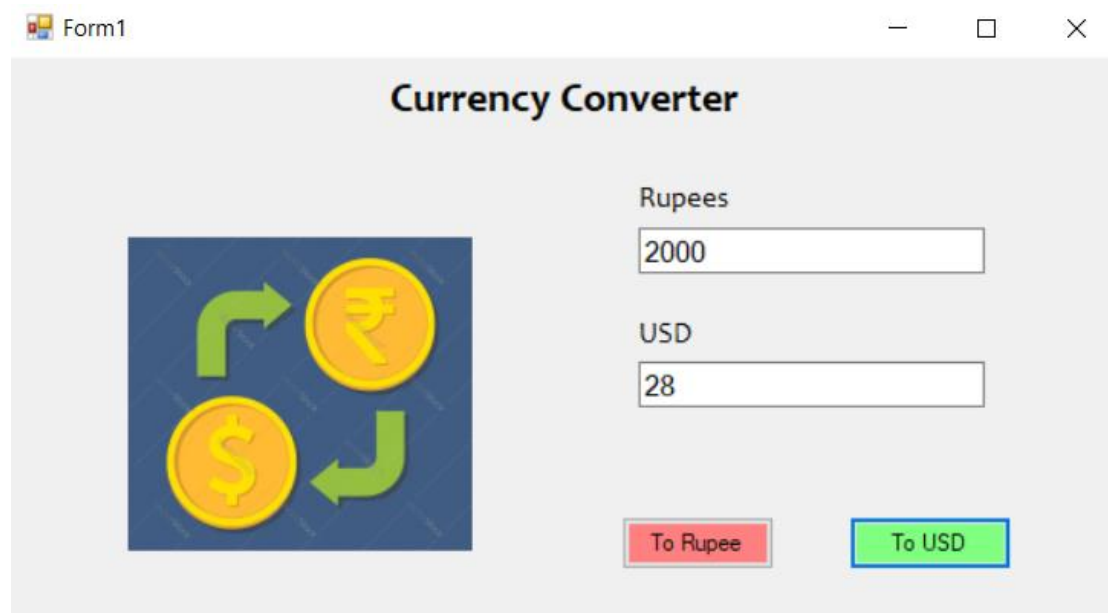
Program:

```
Public Class Form1  
    Private Function getUsd(ByVal r As Double) As Double  
        Dim usd As Double  
        usd = r * 0.014  
        Return usd  
    End Function  
    Private Function getRupees(ByVal usd As Double) As  
Double  
        Dim r As Double  
        r = usd / 0.014  
        Return r  
    End Function  
  
    Private Sub btntousd_Click(sender As Object, e As  
EventArgs) Handles btntousd.Click  
        Dim usd, r As Double  
        r = txtRupee.Text  
        usd = getUsd(r)  
        txtUSD.Text = usd  
    End Sub
```



```
Private Sub btntorupee_Click(sender As Object, e As  
EventArgs) Handles btntorupee.Click  
    Dim usd, r As Double  
    usd = txtUSD.Text  
    r = getRupees(usd)  
    txtRupee.Text = r  
End Sub  
End Class
```

Output:



Practical no 14.

Aim: Develop a program to create class Access members of its class using its object.

Theory:

Visual basic provides full support for object-oriented programming including encapsulation, inheritance and polymorphism. •

CLASS AND OBJECTS: The term class and object are sometimes used to inherchangly, but in fact, classes describe the type of objects, while object are usable instance of classes. So, the act of creating an object is called instantiation. Using the blueprint analogy, a class is a blueprint, and an object is a building made form that blueprint

- **CLASS MEMBERS:** Each class can have different class members that includes properties that describe class data, methods that define class behavior, and events that provides communication between different classes and objects.

- **SYNTAX:**

[attribute list] [access modifiers] [shadows] [MustInherit] [NotInherite] [Partial]
Class name [(of typelist)] [Inherits classname] [Implements interface names]

[statements]

End Class

Program:

```
Public Class employee
    Public Sub New(ByVal sal As Integer, ByVal bonus As Integer)
        Dim t As Integer
        t = sal + bonus

        Console.WriteLine("Total salary =" & t)
        If t > 75000 Then
            Console.WriteLine("Excellent")
        ElseIf t <= 75000 And t >= 50000 Then
            Console.WriteLine("Good Salary")
        ElseIf t >= 50000 And t <= 20000 Then
            Console.WriteLine("Poor Salary")
        End If
    End Sub
End Class
```

```

        Else
            Console.WriteLine("Invalid Salary")
        End If
    End Sub
Protected Overrides Sub Finalize()
    Console.WriteLine("destructor is executed")
End Sub
End Class
Module Module1

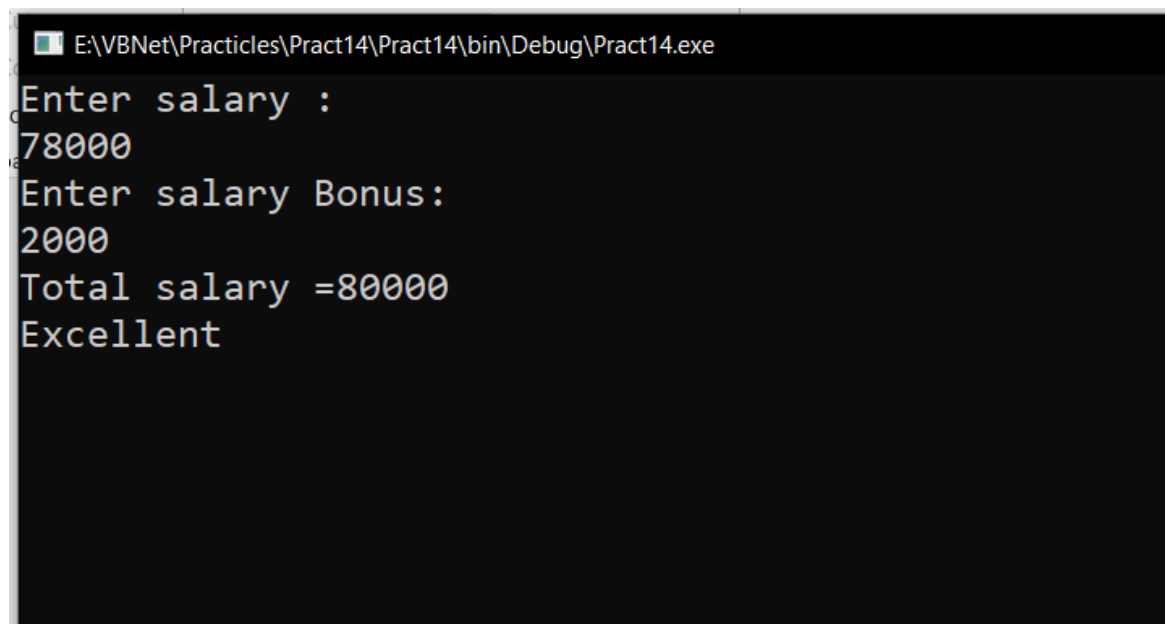
    Sub Main()
        Dim sal As Integer
        Dim bonus As Integer
        Console.WriteLine("Enter salary :")
        sal = Console.ReadLine()
        Console.WriteLine("Enter salary Bonus:")
        bonus = Console.ReadLine()

        Dim a As New employee(sal, bonus)
        Console.ReadLine()
    End Sub

End Module

```

Output:



```

E:\VBNet\Practicles\Pract14\Pract14\bin\Debug\Pract14.exe
Enter salary :
78000
Enter salary Bonus:
2000
Total salary =80000
Excellent

```

Practical no 15.

Aim: Develop a program to demonstrate Overloading and Overriding Methods.

Theory:

1.FUNCTION OVERLOADING:

When we have multiple functions with the same name but different number of parameters, then they are said to be overloaded. This technique is used to enhance the readability of the program.

There are two ways to overload a function, that is

- i. Having different number of arguments
- ii. Having different argument types

Function overloading is normally done when we have to perform one single operation with different number of types of arguments.

2.FUNCTION OVERRIDING:

When the base class and derived class have members functions with exactly the same name, same return-type, and same arguments list, then it is said to be function overriding.

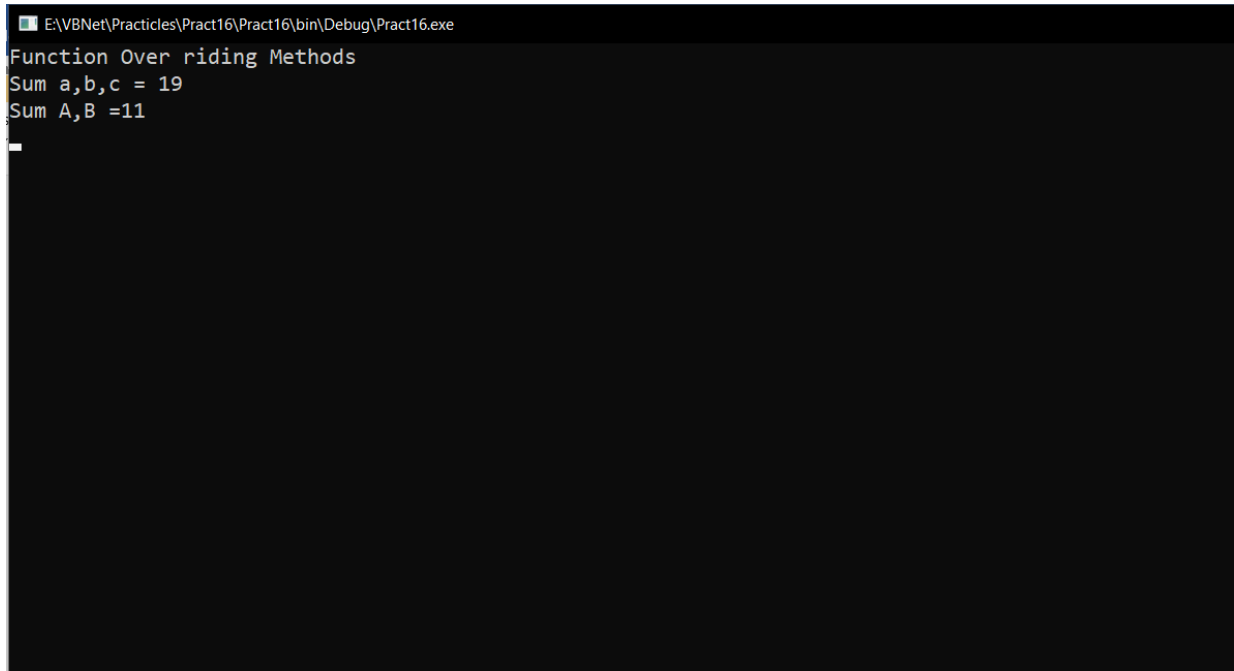
Program:

```
Module Module1
    Function sum(ByVal a As Integer, ByVal b As Integer)
        Dim c As Integer
        c = a + b
        Console.WriteLine("product = {0}", c)
        Console.ReadLine()

    End Function
    Function sum(ByVal a As Integer, ByVal b As Integer,
        ByVal c As Integer)
        Dim c As Integer
        c = a + b + c
        Console.WriteLine("product = {0}", c)
    End Function
    Sub Main(args As String())
```

```
        Dim x = sum(12, 2, 5)
        Dim y = sum(5, 6)
    End Sub
```

Output:



```
E:\VBNet\Practicles\Pract16\Pract16\bin\Debug\Pract16.exe
Function Over riding Methods
Sum a,b,c = 19
Sum A,B =11
```

Practical no 16.

Aim: Design a program to handle a run-time error using Exception handling.

THEORY:

An exception is a problem that arises during the execution of a program. An exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero.

Exceptions provide a way to transfer control from one part of a program to another. VB.Net exception handling is built upon four keywords - Try, Catch, Finally and Throw.

Try – A Try block identifies a block of code for which particular exceptions will be activated. It's followed by one or more Catch blocks.

Catch – A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The Catch keyword indicates the catching of an exception.

Finally – The Finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not.

Throw – A program throws an exception when a problem shows up. This is done using a Throw keyword.

Syntax

Assuming a block will raise an exception, a method catches an exception using a combination of the Try and Catch keywords. A Try/Catch block is placed around the code that might generate exception. Code within a Try/Catch block is referred to as protected code, and the syntax for using Try/Catch looks like the following –

Try

[tryStatements]

[Exit Try]

[Catch [exception [As type]] [When expression]

[catchStatements]

[Exit Try]]

[Catch ...]

[Finally

[finallyStatements]]

End Try

PROGRAM:

Module Program

```
Public Class studentsZeroException : Inherits Exception
```

```
    Public Sub New(ByVal stdetails As String)
```

```
        MyBase.New(stdetails)
```

```
    End Sub
```

```
End Class
```

```
Public Class studentManagement
```

```
    Dim stud As Integer = 0
```

```
    Sub showDetail()
```

```
        If (stud = 0) Then
```

```
            Throw (New studentsZeroException("student roll no 'Zero' dose not exit"))
```

```
        Else
```

```
            Console.WriteLine("student is {0}", stud)
```

```
        End If
```

```
    End Sub
```

```
End Class
```

```
Sub Main(args As String())
```

```
    Dim stdmg As studentManagement = New
```

```
studentManagement()
```

```
    Try
```

```
        stdmg.showDetail()
```

```
    Catch ex As Exception
```

```
        Console.WriteLine("studentsZeroException {0}", ex.Message)
```

```
    End Try
```

```
    Console.ReadLine()
```

```
End Sub
```

```
End Module
```

OUTPUT:

studentsZeroException student roll no 'Zero' dose not exit

Practical no 17.

Aim:

USE MULTIPLE TABLE DESIGN EXMPLE OF EMPLOYEE AND DEPARTMENT.

THEORY:

procedure to create multiple table:

Open Visual Studio .NET and select a new Visual Basic .NET Project.

From main menu in Visual Studio select PROJECT-->Add New Item . Then Add New Item dialogue will appear and select Crystal Reports from the dialogue box.

Select Report type from Crystal Reports gallery.

Accept the default settings and click OK.

Next step is to select the appropriate connection to your database. Here we are going to select OLEDB connection for SQL Server Select OLE DB (ADO) from Create New Connection.

Select Microsoft OLE DB Provider for SQL Server .

Next screen is the SQL Server authentication screen . Select your Sql

Server name , enter userid , password and select your Database Name . Click next , Then the screen shows OLE DB Property values , leave it as it is , and click finish.

Then you will get your Server name under OLEDB Connection from there select database name (Crystaldb) and click the tables , then you can see all your tables from your database.

Select all table from the table list to right side list box, because we are creating report from three tables (OrderMaster, OrderDetails, Product) .

The next step is to make relation between these selected tables.

Here we are connecting the related fields from each table. For that we arrange the tables in visible area in the list (this is not necessary) and select the field we are going to make relation and drag to the related field of the other table. After made the relation the screen is look like the following picture.

Next step is to select the fields from the tables . Here we are selecting only Customername , orderdate from ordermastertable , Productname from product table and quantity from order details.

Click the Finish button because now we are not using other functionalities of this wizard. After that you will get the Crystal Reports designer window . You can arrange the fields in the designer window according to your requirement to view the report . For rearranging you can drag the field object in the screen . For editing right click the

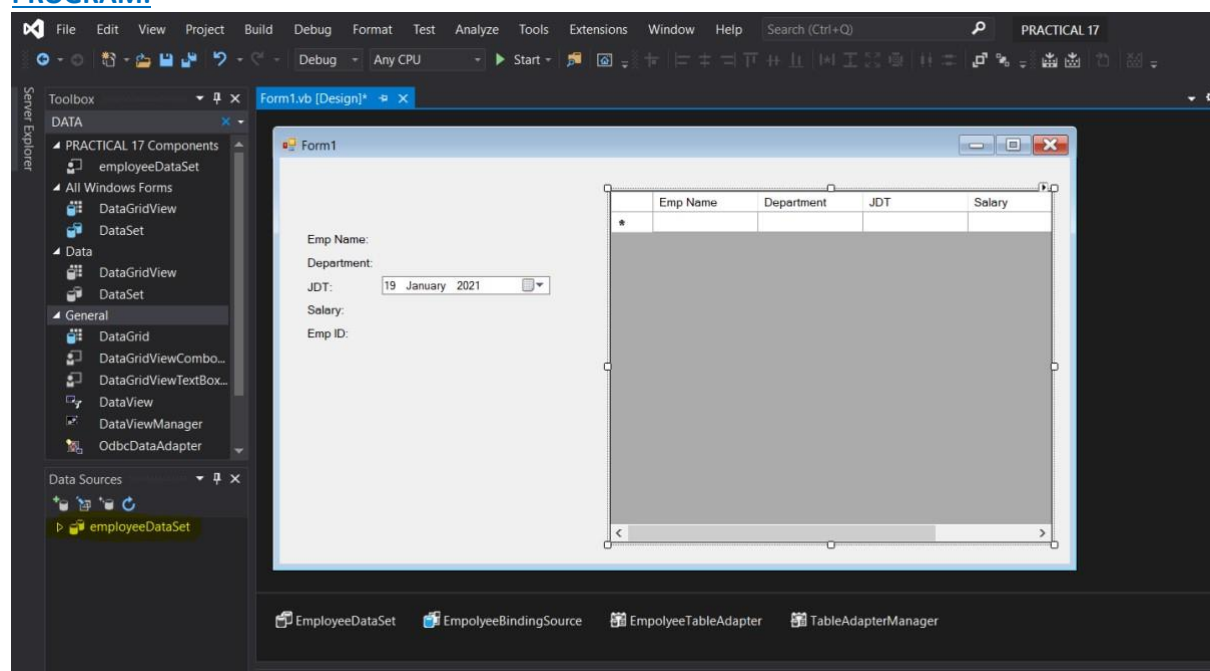
field object and select Edit Text Object. The following picture shows the sample of designer window after rearrange the field.

Now the designing part is over and the next step is to call the created Crystal Reports in VB.NET through Crystal Reports Viewer control .

Select the default form (Form1.vb) you created in VB.NET and drag a button and CrystalReportViewer control to your form.

Select Form's source code view and put the code on top.

PROGRAM:



Data source:

employee : Database (Access 2007) - Microsoft Access

Table Tools: Home, Create, External Data, Database Tools, Datasheet

Security Warning: Certain content in the database has been disabled. Options...

All Tables: Employee

Emp Name	Department	JDT	Salary	Emp ID	Add New Field
ASHOKE	MXTG	23-01-2021	12000	2121	
NIKHIL	ACCTS	26-11-2020	25000	1212	
SOHAN	ACCTS	01-01-2021	20000	1111	
SURAJ	MKTG	10-12-2020	18000	2222	
*					

OUTPUT:

File Edit View Project Build Debug Format Test Analyze Tools Extensions Window Help Search (Ctrl+Q) PRACTICAL 17

Process: [16372] PRACTICAL 17.exe Lifecycle Events Thread: Stack Frame:

Form1.vb [Design]

Form1

Emp Name: Department: JDT: 19 January 2021 Salary: Emp ID:

Form1

Emp Name	Department	JDT
SOHAN	ACCTS	01-01-2021
ASHOKE	MXTG	23-01-2021
SURAJ	MKTG	10-12-2020
NIKHIL	ACCTS	26-11-2020

EmployeeDataSet EmployeeBindingSource EmployeeTableAdapter TableAd

Locals Search (Ctrl+E) Name Value Type Immediate Window

Practical no 16.

Aim: DESIGN REGISTRATION FROM OF COLLEGE USING AUTO POSTBACKPROPERTY

THEORY:

AutoPostBack property allows controls which cannot submit the Form (PostBack) on their own and hence ASP.Net has provided a feature using which controls like DropDownList, CheckBoxList, RadioButtonList, etc. can perform PostBack.

AutoPostBack is a feature that is available on a few controls, the main purpose of it is that if any change happened on a control by the client, it should postback to the server to handle that change at serverside, handling that change at serverside could be for many reasons i.e storing a value to DB, make changes to the page depending on it, etc

How does AutoPostBack work?

AutoPostBack property uses JavaScript and HiddenFields for performing PostBack. There are two HiddenFields __EVENTTARGET and __EVENTARGUMENT and a JavaScript function __doPostBack.

Program:

```
<%@ Page Title="Home Page" Language="VB"
MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.vb"
Inherits="_1._Default" %>
```

```
<asp:Content ID="BodyContent"
ContentPlaceHolderID="MainContent" runat="server">
```

```
    <section id ="main-ccontent">
        <section id ="wrapper">
            <div class ="row">
                <div class ="col-lg-12">
                    <section class ="panel">
                        <header class ="panel-heading">
                            <div class ="col-md-4 col-md-offset-4">
                                <h1>
```

```

        Student Regidtration
    From      </h1>

    </div>
</header>

<div class="panel-body">
    <div class="row">
        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:lable1 Text ="Student Name" runat
="server" />
                <asp:TextBox runat ="server" Enabled ="true" CssClass
="from-control input-sm" placeholder ="Student Name" />
            </div>
        </div>
        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:lable1 Text ="Father Name" runat
="server" />
                <asp:TextBox runat ="server" Enabled ="true" CssClass
="from-control input-sm" placeholder ="Father Name" />
            </div>
        </div>
    </div>
</div>
<div class="panel-body">
    <div class="row">
        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:lable1 Text ="DOB" runat ="server" />
                <asp:TextBox runat ="server" Enabled ="true"
TextMode="Date" CssClass ="from-control input-sm" placeholder
="Student
Name" />
            </div>
        </div>
    </div>

```

```

        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:label1 Text ="Program" runat ="server"
/>
                <asp:TextBox runat ="server" Enabled ="true" CssClass
="from-control input-sm" placeholder ="program" />
            </div>
        </div>
    </div>
</div>
<div class="panel-body">
    <div class="row">
        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:label1 Text ="region" runat ="server" />
                <div>
                    <asp:DropDownList runat ="server" CssClass
="fromcontrol input-sm">
                        <asp:ListItem Text ="Pakistan" />
                        <asp:ListItem Text ="Iran" />
                        <asp:ListItem Text ="Iraq" />
                        <asp:ListItem Text ="Turkey" />
                        <asp:ListItem Text ="India" />
                        <asp:ListItem Text ="China" />
                    </asp:DropDownList>
                </div>
            </div>
        </div>
        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:label1 Text ="Father Name" runat
="server" />
                <asp:TextBox runat ="server" Enabled ="true" CssClass
="from-control input-sm" placeholder ="Father Name" />
            </div>
        </div>
    </div>
</div>

```

</div>
</section>

COPY OF OUTPUT:

Application name Home About Contact Student Registration

Student Registration

Student Name	Father Name
<input type="text" value="Student Name"/>	<input type="text" value="Father Name"/>
DOB	Program
<input type="text" value="mm/dd/yyyy"/>	<input type="text" value="Program"/>
Region	Address
<input type="text" value="Pakistan"/>	<input type="text" value="Address"/>
Cell No	Program
<input type="text" value="Student Name"/>	<input type="radio"/> Male
	<input type="radio"/> Female
<div><div>Save</div><div>Update</div><div>Delete</div></div>	

Practical no 19.

Aim: Simple application for following function:

(1) Login (2) Surfing (3) Logout taking

Into considerations (Application, Session, Server object, global .asa file and their events, methods and collection) and demonstration enabling and disabling of session)

THEORY:

The ASP.NET login controls provide a robust login solution for ASP.NET Web applications without requiring programming. By default, login controls integrate with ASP.NET membership and forms authentication to help automate user authentication for a Web site. It provides you with a ready-to-use user interface that queries the user name and password from the user and offers a Log In button for login. It validates user credentials against the membership API and encapsulating the basic forms authentication functionality like redirecting back to the original requested page in a restricted area of your application after the successful login.

Program:

CODE FOR LOGIN-DESIGN

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="loginPage.aspx.cs" Inherits="login_project.loginPage"
%>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

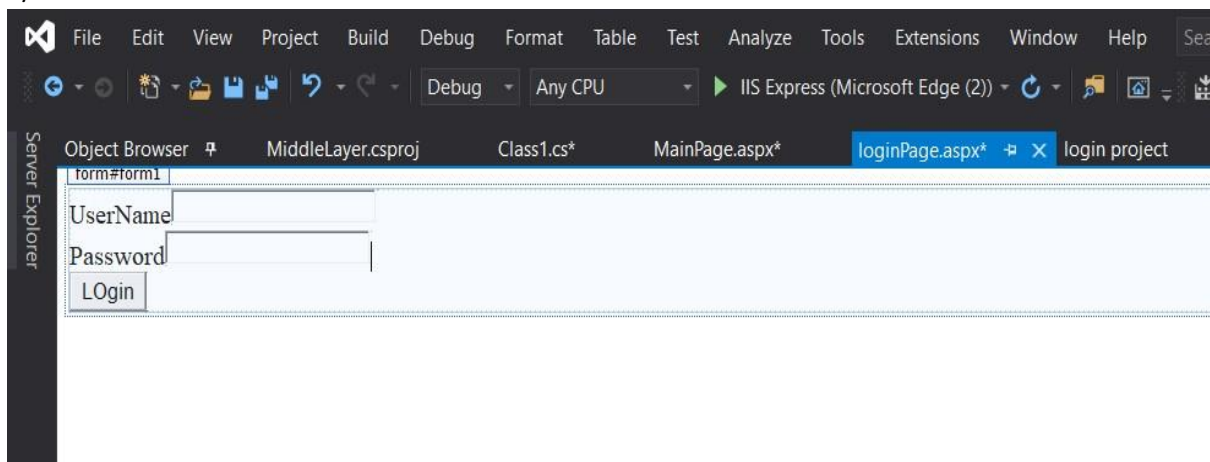
```
<title></title>
```

```
</head>
```

```

<body>
  <form id="form1" runat="server">
    <asp:Label ID="Label1" runat="server" Text
="UserName"></asp:Label>
    <asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>
    <br />
    <asp:Label ID="Label2" runat="server" Text
="Password"></asp:Label>
    <asp:TextBox ID="txtPassword" runat="server"></asp:TextBox>
    <br />
    <asp:Button ID="btnLogin" runat="server" Text="Login" />
  </form>
</body>
</html>

```



CODE FOR MAIN PAGE

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="MainPage.aspx.cs" Inherits="login_project.MainPage"
%>

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head runat="server">

```

```

  <title></title>

```

```

</head>

```

```

<body>

```

```

  <form id="form1" runat="server">

```

```

    <div>

```

```

      Welcome + User

```

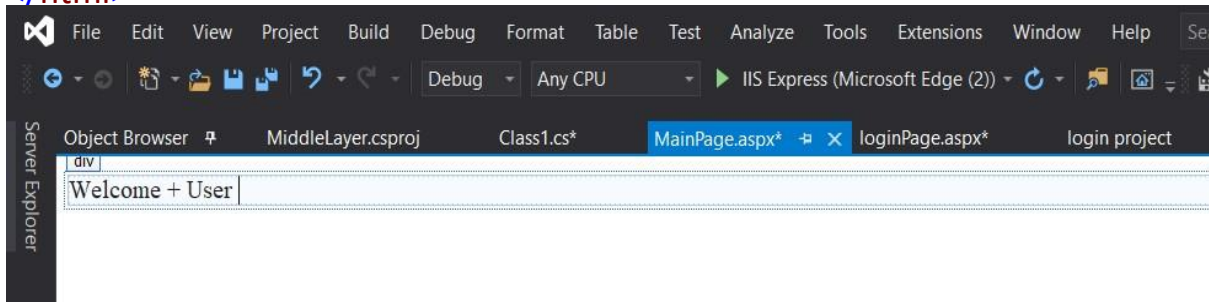
```

    </div>

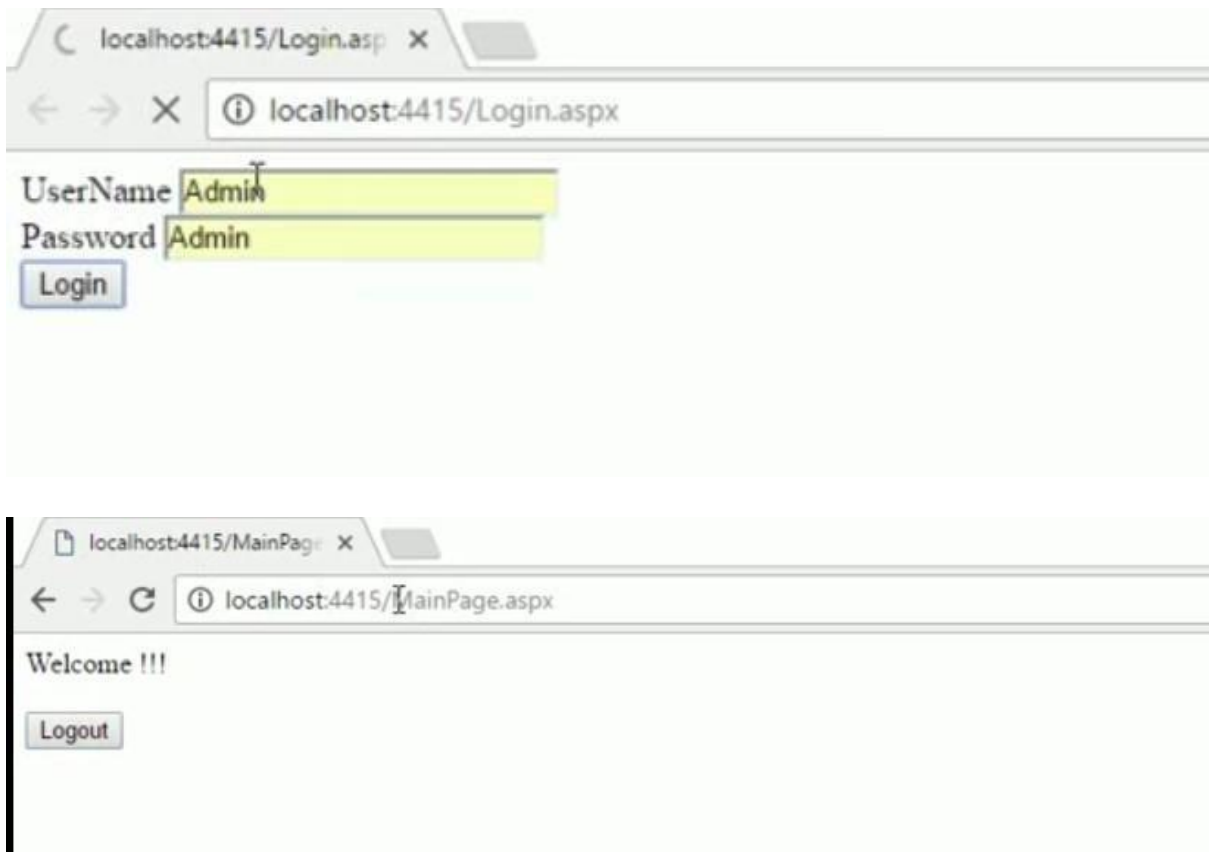
```



```
</form>
</body>
</html>
```



Output:



Practical no 20.

Aim: Creation of file, entry, reading data from a file.

THEORY:

Creation of file and reading content from file.

CODE:

```
using System;

using System.Collections.Generic;
using System.IO; using
System.Linq; using System.Text;
using System.Threading.Tasks;

namespace pract_20
{
    class Program
    {
        static void Main(string[] args)
        {
            string filepath = @"C:\Users\HP";
            //string[] lines = File.ReadAllLines(filepath);
            List<string> lines = new List<string>();      lines =
            File.ReadAllLines(filepath).ToList();      foreach(String
            line in lines )

            {

                Console.WriteLine(line);

            }

            Console.ReadLine();

        }
    }
}
```

COPY OF PROGRAM AND OUTPUT:

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace pract_20
9 {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             string filepath = @"C:\Users\HP\";
15             //string[] lines = File.ReadAllLines(filepath);
16             List<string> lines = new List<string>();
17             lines = File.ReadAllLines(filepath).ToList();
18             foreach(String line in lines )
19             {
20                 Console.WriteLine(line);
21             }
22             Console.ReadLine();
23         }
24     }
25 }
```

```
C:\Users\shad sluiter\source\repos\FileIODemo\FileIODemo\bin\Debug\FileIODemo.exe
Jeff, Bezos, www.yahoo.com
Bill, Gates, www.microsoft.com
Andy, Grove, www.intel.com
Sheryl, Sandberg, www.facebook.com
Steve, Jobs, www.apple.com
Shad, Sluiter, www.shadsluiter.com
```

Practical no 21.

Aim: Using components create: (1) Advertisement (using Ad rotator) (2) Book example (using Next function) (3) find capabilities of browser (Browser object capabilities)

THEORY:

%@PageLanguage="vb"AutoEventWireup="false"CodeBehind="WebForm1.aspx.vb"Inherits="WebApplication2.WebForm1"%

```
<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0  
Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1transiti  
onal.dtd">
```

```
<htmlxmlns="http://www.w3.org/1999/xhtml">
```

```
<headrunat="server">
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<formid="form1"runat="server">
```

```
<div>
```

```
<asp:AdRotatorID="AdRotator1"runat="server"
```

```
AdvertisementFile="~/XMLFile1.xml"/>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

XML FILE Name - XMLFile1.xml

```
<?xmlversion="1.0"encoding="utf-8" ?>
```

```
<Advertisements>
```

```
<Ad>
```

```
<ImageUrl>Image/Tulips.jpg</ImageUrl>
```

```
<NavigateUrl>http://www.aero.iitb.ac.in/</NavigateUrl>
```

```
<AlternateText>Aerospace Engineering Main Site</AlternateText>
```

```
<Impressions>50</Impressions>
```

```
<Keyword>education</Keyword>
```

```
</Ad>
```

```
<Ad>
```

```
<ImageUrl>Image/Desert.jpg</ImageUrl>
```

```
<NavigateUrl>http://www.jipmer.edu/</NavigateUrl>
```

```
<AlternateText>MBBS Pondicherry</AlternateText>
```

```
<Impressions>120</Impressions>
```

```
<Keyword>education</Keyword>
```

```
</Ad>
```

```
</Advertisements>
```

a) Find capabilities of browser (Browser object capabilities)

```
<%@PageLanguage="vb"AutoEventWireup="false"CodeBehind="WebForm1.aspx.vb" Inherits="Browsercap.WebForm1"%>
```

```
<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1transitional.dtd">
```

```
<htmlxmlns="http://www.w3.org/1999/xhtml">
```

```
<headrunat="server">
```

```
<title>Browser Capabilities</title>
```

```
</head>
```

```
<body>
```

```
<formid="form1"runat="server">
```

```
<div>
```

```
<%
```

```
'Dim obj As Object
```

```
'obj = Server.CreateObject("MSWC.BrowserType")
```

```
Response.Write("Browser="&Request.Browser.Browser)
```

```
Response.Write("<br/>")
```

```
Response.Write("active X  
Control="&Request.Browser.ActiveXControls)
```

```
Response.Write("<br/>")
```

```
Response.Write("AOL="&Request.Browser.AOL)
```

```
Response.Write("<br/>")
```

```
'Response.Write("CLR VERSION=" &Request.Browser.ClrVersion)
'Response.Write("<br/>")

Response.Write("Cookies="&Request.Browser.Cookies)

Response.Write("<br/>")

Response.Write("Crawler="&Request.Browser.Crawler)

Response.Write("<br/>")

Response.Write("JavaApplets="&Request.Browser.JavaApplets)

Response.Write("<br/>")

Response.Write("Major Version="&Request.Browser.MajorVersion)

Response.Write("<br/>")

Response.Write("Minor Version="&Request.Browser.MinorVersion)

Response.Write("<br/>")

Response.Write("Platform"&Request.Browser.Platform)

Response.Write("<br/>")

Response.Write("VBScript="&Request.Browser.VBScript)

Response.Write("<br/>")

Response.Write("Version="&Request.Browser.Version)

Response.Write("<br/>")

%>

</div>

</form>
```

</body></html>

COPY OF OUTPUT:



Browser=Mozilla
active X Control=False
AOL=False
Cookies=True
Crawler=False
JavaApplets=False
Major Version=0
Minor Version=0
PlatformWinNT
VBScript=False
Version=0.0



Practical no 22.

Aim: Create Online application (student, employee, product, shopping mall) using dataset, data reader

THEORY:

```
using System;
using
System.Collections.Gene
ric; using System.Linq;
using System.Web;
using System.Web.UI;
using
System.Web.UI.WebCon
trols;
```

```
namespace pract_22
{
    public partial class contact : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                btnDelete.Enabled = false;
            }
        }

        protected void bthClear_Click(object sender, EventArgs e)
        {
            Clear();
        }
        public void Clear()
        {
            hfcontactID.Value = " ";
            txtName.Text = txtMobile.Text = txtAddress.Text
            = "";
            lblErrorMessage.Text =
            lblErrorMessage.Text = "";
            btnSave.Text =
            "";
        }
    }
}
```

```

        btnDelete.Enabled = false;
    }
}
}

```

CODE FOR DESIGN

```
<!DOCTYPE html>
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
<asp:HiddenField ID="hfcontactID"
runat="server"></asp:HiddenField>
            <table>
                <tr>
                    <td>
                        <asp:Label ID="Lable1" runat="server"
Text="Name"></asp:Label>
                    </td>
                    <td colspan="2">
                        <asp:TextBox ID="txtName"
runat="server"></asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td>
                        <asp:Label ID="Labe2" runat="server"
Text="Mobile"></asp:Label>
                    </td>
                    <td colspan="2">
                        <asp:TextBox ID="txtMobile"
runat="server"></asp:TextBox>
                    </td>
                </tr>
            </table>
        </div>
    </form>

```

```

        <tr>
            <td>
                <asp:Label ID="Label3" runat="server"
Text="Address"></asp:Label>
            </td>
            <td colspan ="2">
                <asp:TextBox ID="txtAddress" runat="server" TextMode
="MultiLine" ></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>

            </td>
            <td colspan ="2">
                <asp:Button ID="btnSave" runat="server" Text="Save"
/>
                <asp:Button ID="btnDelete" runat="server"
Text="Delete" />
                <asp:Button ID="bthClear" runat="server" Text="Clear"
OnClick="bthClear_Click" />
            </td>
        </tr>
        <tr>
            <td>

            </td>
            <td colspan ="2">
                <asp:Label ID="lbiSuccessMessage" runat="server"
Text=""
ForeColor ="Green" ></asp:Label>
            </td>
        </tr>
        <tr>
            <td>

            </td>
            <td colspan ="2">
                <asp:Label ID="lbiErrorMessage" runat="server"

```

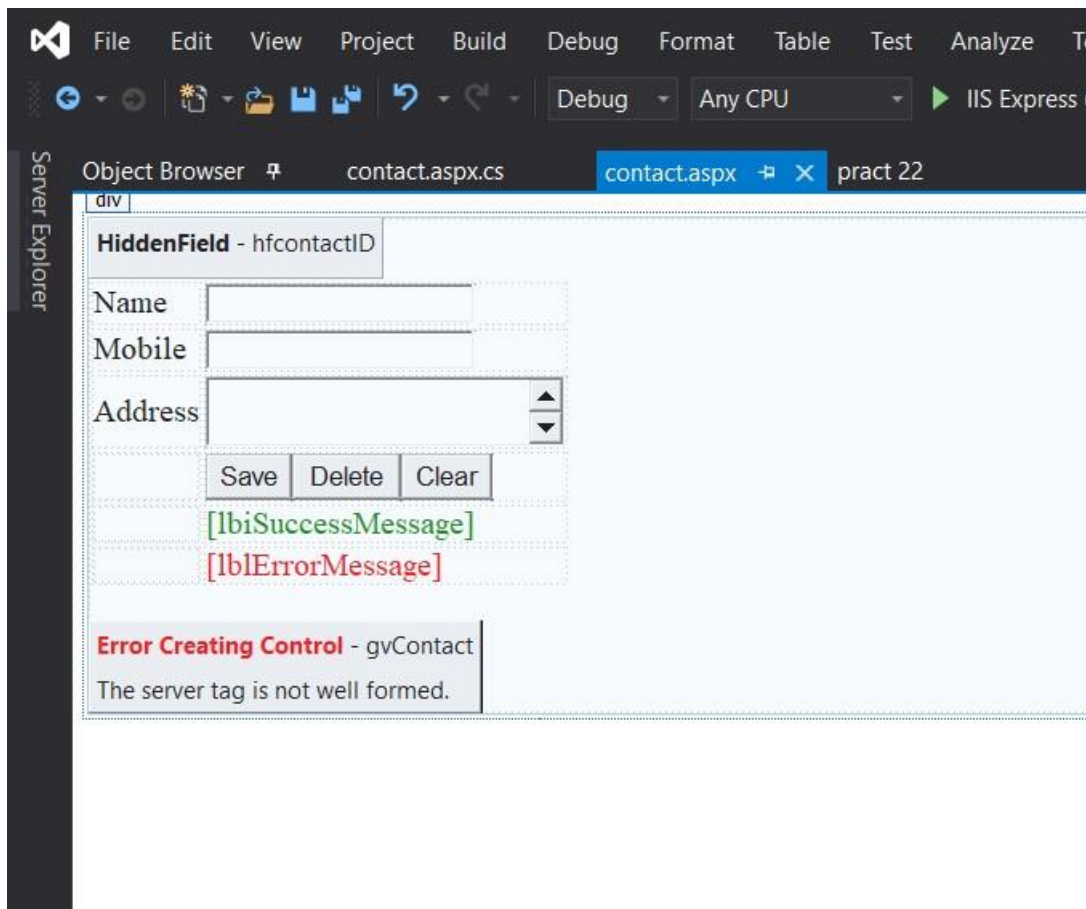
```

Forecolor="Red" Height="22px"></asp:Label>

    </tr>
</table>
<br />
<asp:GridView ID="gvContact" runat="server" AutoGenerateColumns
="false" >
    <Columns>
        <asp:BoundField DataField ="Name" HeaderText ="Name" />
        <asp:BoundField DataField ="Mobile" HeaderText
="Mobile" />
        <asp:BoundField DataField ="Address" HeaderText
="Address" />
        <asp:TemplateField >
            <ItemTemplate>
                <asp:LinkButton ID="lnkView" runat="server"
CommandArgument ="<%#Eval("contact") %>">View</asp:LinkButton>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
</div>
</form>
</body>
</html>

```

DESIGN:



Name

Mobile

Address

Name	Mobile	Address	
Smith	5421587432	address	View

Name

Mobile

Address

Saved Successfully

100 %

Results		Messages		
ContactID	Name	Mobile	Address	
1	1	+	Smith	5421587432 address

Name
 Mobile
 Address

Name	Mobile	Address	
Smith	5421587432	address	View

COPY OF PROGRAM:

```

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)
Debug Any CPU IIS Express (Microsoft Edge (2))
Server Explorer Object Browser contact.aspx.cs contact.aspx pract 22
pract 22 pract_22.contact
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7
8 namespace pract_22
9 {
10     2 references
11     public partial class contact : System.Web.UI.Page
12     {
13         0 references
14         protected void Page_Load(object sender, EventArgs e)
15         {
16             if (!IsPostBack)
17             {
18                 btnDelete.Enabled = false;
19             }
20         }
21         0 references
22         protected void btnClear_Click(object sender, EventArgs e)
23         {
24             Clear();
25         }
26         1 reference
27         public void Clear()
    
```

Practical no 23.

Aim: Create Online application (student, employee, product, shopping mall) using data grid

THEORY:

Public Class Form1

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'Database1DataSet.Table1' table. You can move, or remove it, as needed.

Me.Table1TableAdapter.Fill(Me.Database1DataSet.Table1)

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

Table1BindingSource.MovePrevious()

End Sub

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

Table1BindingSource.AddNew()

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click

Table1BindingSource.MoveNext()

End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click

Table1BindingSource.RemoveCurrent()

End Sub


```

Private Sub Button6_Click(sender As Object, e As EventArgs)
Handles
Button6.Click
    Me.Close()
End Sub

```

```

Private Sub Button4_Click(sender As Object, e As EventArgs)
Handles
Button4.Click
    On Error GoTo SaveError
    Table1BindingSource.EndEdit()
    Table1TableAdapter.Update(Database1DataSet.Table1)
    MsgBox("Data has been saved")

```

```

SaveError:
    Exit Sub

```

```

End Sub

```

```

Private Sub BtnSearch_Click(sender As Object, e As EventArgs)
Handles
BtnSearch.Click
    If TxtSearch.Text = "" Then
        Call notFound()
        Exit Sub

    Else
        Table1BindingSource.Filter = "(Convert (ID,'System.String') LIKE
"" & TxtSearch.Text & "" )" &
            "OR(STUDENT_ID LIKE "" & TxtSearch.Text & "" )"
OR(FIRST_NAME LIKE "" & TxtSearch.Text & "" )" &
            "OR(LAST_NAME LIKE "" & TxtSearch.Text & "" )" OR(EMAIL
LIKE "" &
TxtSearch.Text & "" )" &
            "OR(MOBILE_NO LIKE "" & TxtSearch.Text & "" )"

        If Table1BindingSource.Count <> 0
Then
            With DataGridView1
                .DataSource = Table1BindingSource

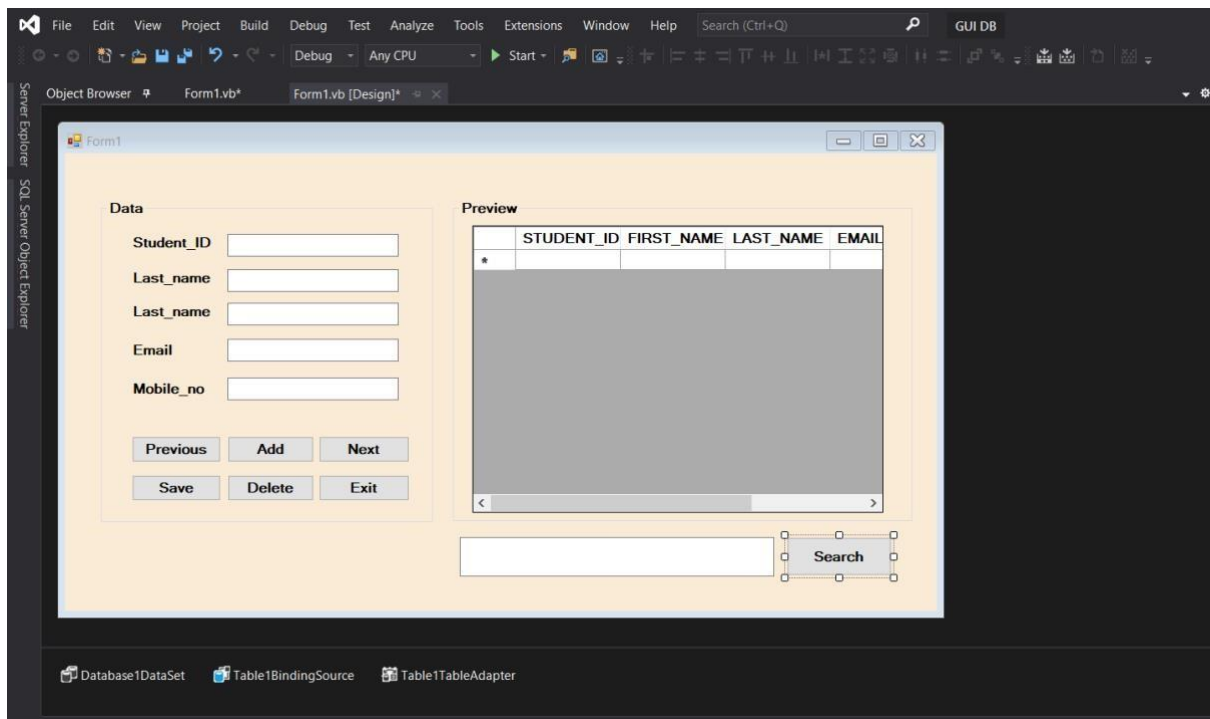
```

```

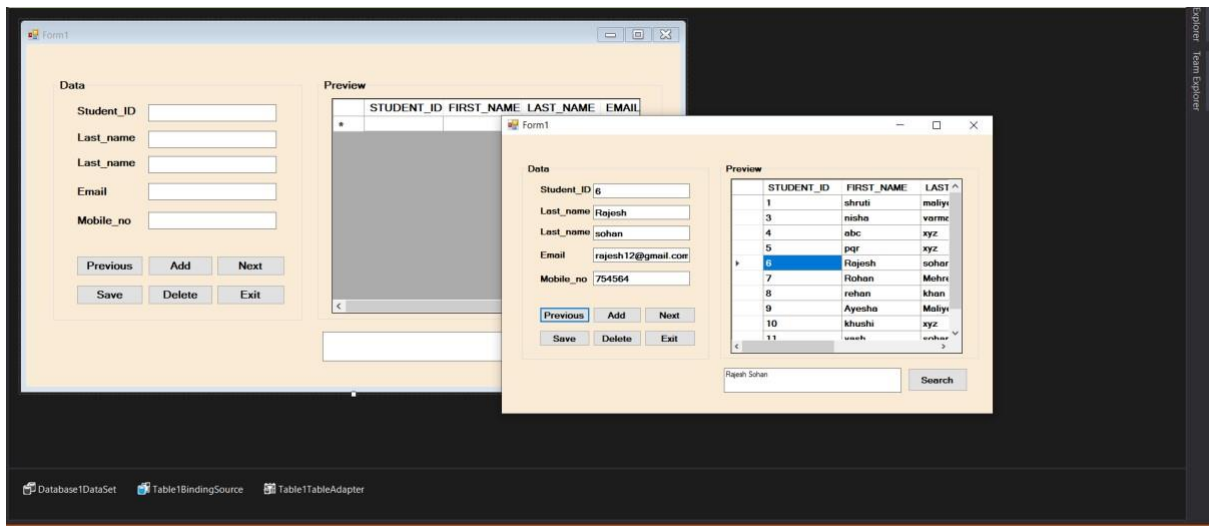
        End With
    Else
        MsgBox("The search item was not found")
        Table1BindingSource.Filter = Nothing
    End If
End If
End Sub
Private Sub notFound()
End Sub
End Class

```

DESIGNE:



COPY OF OUTPUT:



Practical no 24.

Aim: Design a program to handle a run-time error using Exception handling.

THEORY:

BIND DATA TO DATA GRID USING PROPERTIES

STEPS TO CREATE DATA WITH DATA GRID VIEW

Step 1- Create table in the database (SQL Server 2012) Create the database and name it as Login.

Add table (here table name: tbllogin)

Set primary key to Id column

Step 2- Insert values into tbllogin table. **Step 3-**

- Create new project in Visual Studio 2015.
Go to File-> New-> Website -> Visual C#-> ASP.NET empty Website-> Entry Application Name-> OK.
- Add Web form to the Website.
Project name-> Add-> Add New Item-> Web Form-> write name
- >Add
- HomePage.aspx (Web form) page is created
Click Design Button-> Add Grid View From Toolbox. ToolBox-> Data-> Grid View.
- Right Click on Grid View-> select View Code.
Add the namespaces, mentioned below, in the code back-end page. After the completion of adding the namespaces, you need to write the code, as shown below

CODE:

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq; using
```

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

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

```
namespace BG
```

```
{
```

```
    public partial class INDEX : System.Web.UI.Page
```

```

{

    string connectionString = @"Data Source = (local)\sqle2017; Initial catalog =
phoneBookBD; Integrated security = true; ";    protected void Page_Load(object
sender, EventArgs e)
    {

        using(sqlConnection SqlCon = new sqlConnectionString)

        {

            SqlCon.open();

            SqlDataAdapter SqlDa = new SqlDataAdapter("SELECT
*FROM phoneBook", SqlCon);
            DataTable dtbl = new DataTable();

            SqlDa.Fill(dtbl);

            gvphoneBook.DataSource = dtbl;

            gvphoneBook.DataBind();

        }

    }
}

```

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="BG.WebForm1" %>

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="gvphoneBook" runat="server"
AutoGenerateColumns="false" >
                <Columns >

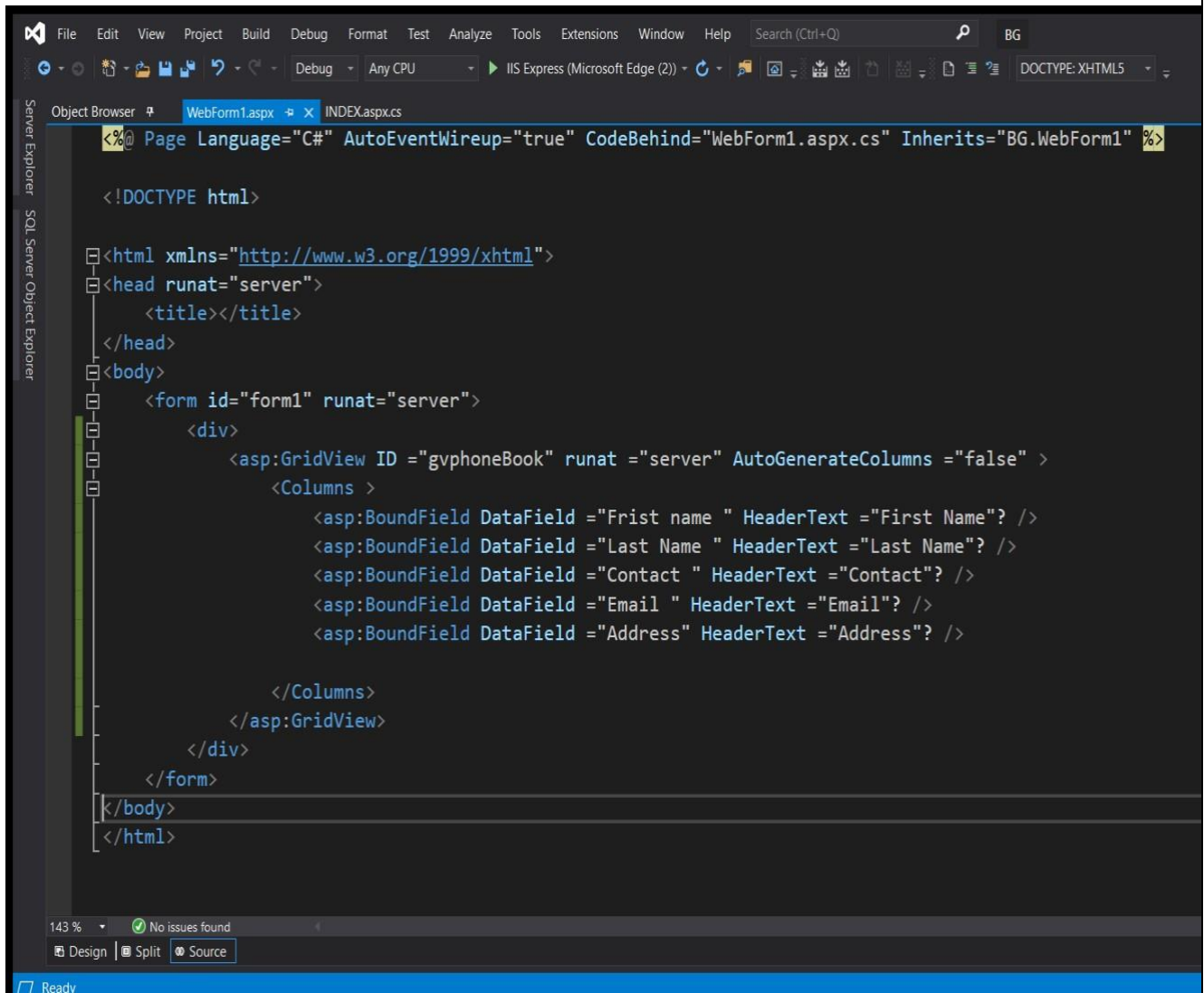
```

```
        <asp:BoundField DataField ="Frist name " HeaderText
="First Name"? />
        <asp:BoundField DataField ="Last Name " HeaderText
="Last
Name"? />
        <asp:BoundField DataField ="Contact " HeaderText
="Contact"? />        <asp:BoundField DataField ="Email "
HeaderText ="Email"? />
        <asp:BoundField DataField ="Address" HeaderText
="Address"? />

    </Columns>
</asp:GridView>
</div>
</form>
</body>
</html>
```

COPY OF PROGRAM AND

OUTPUT:



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="BG.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="gvphoneBook" runat="server" AutoGenerateColumns="false">
                <Columns>
                    <asp:BoundField DataField="Frist name " HeaderText="First Name"? />
                    <asp:BoundField DataField="Last Name " HeaderText="Last Name"? />
                    <asp:BoundField DataField="Contact " HeaderText="Contact"? />
                    <asp:BoundField DataField="Email " HeaderText="Email"? />
                    <asp:BoundField DataField="Address" HeaderText="Address"? />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.UI;
6  using System.Web.UI.WebControls;
7
8  namespace BG
9  {
10     public partial class INDEX : System.Web.UI.Page
11     {
12
13         string connectionString = @"Data Source = (local)\sqle2017; Initial catalog = phoneBookBD; Integrated security = true; ";
14         protected void Page_Load(object sender, EventArgs e)
15         {
16             using(sqlConnection SqlCon = new SqlConnection(connectionString))
17             {
18                 SqlCon.open();
19                 SqlDataAdapter SqlDa = new SqlDataAdapter("SELECT *FROM phoneBook", SqlCon);
20                 DataTable dtbl = new DataTable();
21                 SqlDa.Fill(dtbl);
22                 gvphoneBook.DataSource = dtbl;
23                 gvphoneBook.DataBind();
24             }
25         }
26     }
27 }

```

COPY OF OUTPUT:

First Name	Last Name	Contact	Email	Address
James	Smith	51465145	James789@gmail.com	address
Michael	Smith	54851714	email@gmail.com	address
Robert	Smith	54787712	robert@gmail.com	address
Maria	Garcia	54475778	maria132@hotmail.com	address
David	Smith	54851032	davidsmith01@yahoo.com	address

