

Tutorial 1 MT264

Tutor : Rifat Hamoudi

Staff No. : 00567451

Pager No. : 0781-2796265

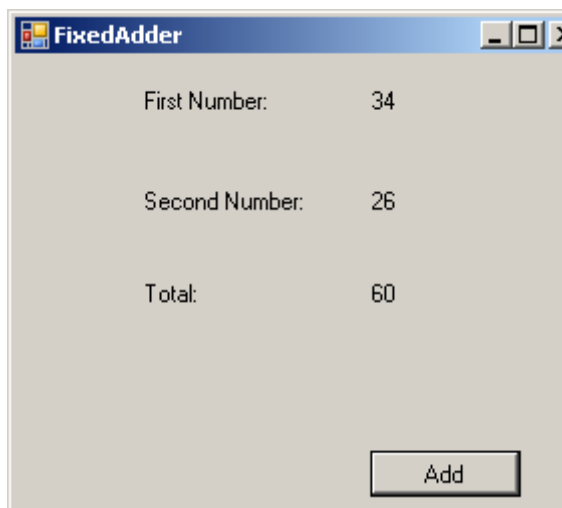
I have put this tutorial on the web. This tutorial can be viewed and downloaded from <http://www.rifathamoudi.co.uk> then selecting MT264 Tutorials then Tutorial 1.

1) Explain with examples what the followings are :

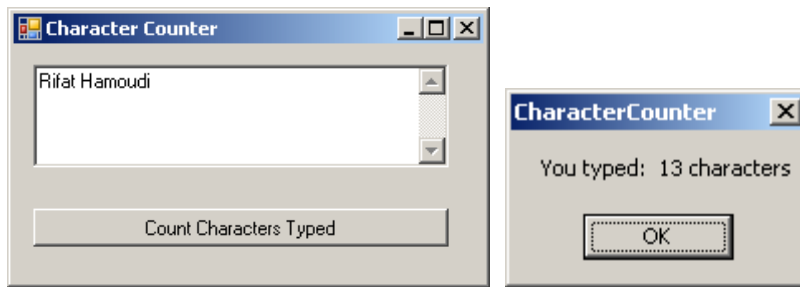
- a) Container
- b) Controls

2) Explain the activities during software development and suggest possible stages in the design of a simple project

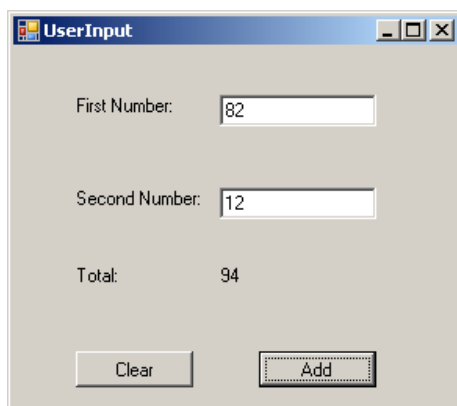
3) Design and implement a Visual Basic program that adds two numbers. An example output is as shown below :



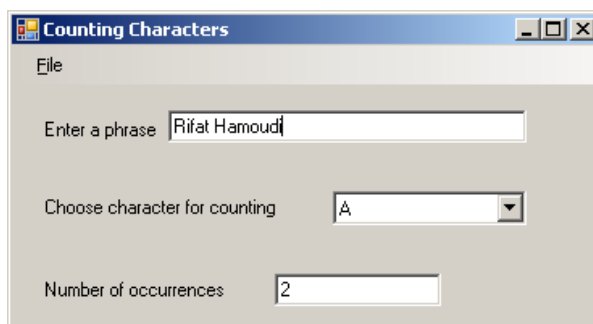
4) Design and implement a Visual Basic program that counts the number of letters in a string. An example output is as shown below :



5) Design and implement a Visual Basic program that adds two numbers entered by the user. An example output is as shown below :



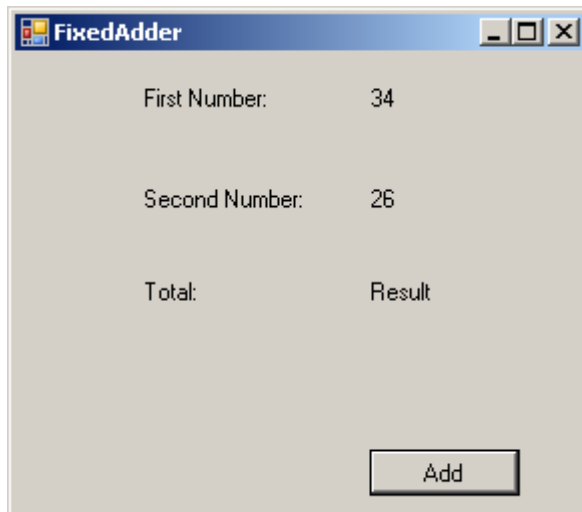
6) Design and implement a Visual Basic program that finds the occurrence of letter in a string entered by the user. An example output is as shown below :



7) Design and implement functional code for a project entitled “Traffic Survey” with the following specification:

An application is required for collecting information about traffic from a particular location. More precisely, we wish to record the number of cars, bicycles and lorries passing (in either direction) a particular point at the roadside. The idea is that the user will record each car, bicycle and lorry as they pass. The application will maintain and display the total numbers of each type of vehicle.

Answer to Question 1



The Form "FixedAdder" is a **Container**

There are 7 **Controls** contained within it.

First Number, Second Number and Total are called **Label** controls

First Number **Label** control has Text property initialised to 34

Second Number **Label** control has Text property initialised to 26

Total **Label** control has Text property initialised to "Result"

Button control is initialised to "Add"

Answer to Question 2

Requirements analysis : This is the activity of gathering and analysing information about what the product should do and what the users' needs are.

Specification : This is the task of formally describing and agreeing what a product will do.

Design : This is the activity of creating a solution for the specification which does not yet give the full code in a chosen programming language. Usually notations are used that allow increasingly formal levels of detail to be given as the development of the solution proceeds.

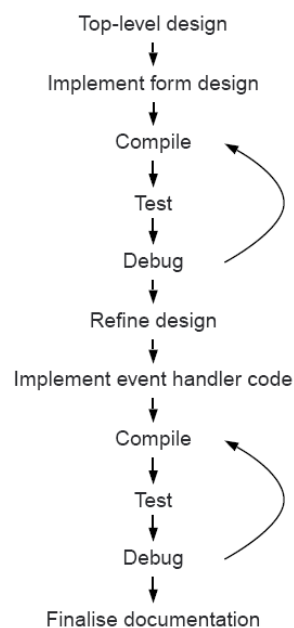
Implementation : This is the activity of coding a design and of putting together the different parts of the piece of software.

Testing : and debugging At this stage the software is run and checked. A selection of possible cases are tested to ensure that the software behaves according to the specification. Errors are corrected.

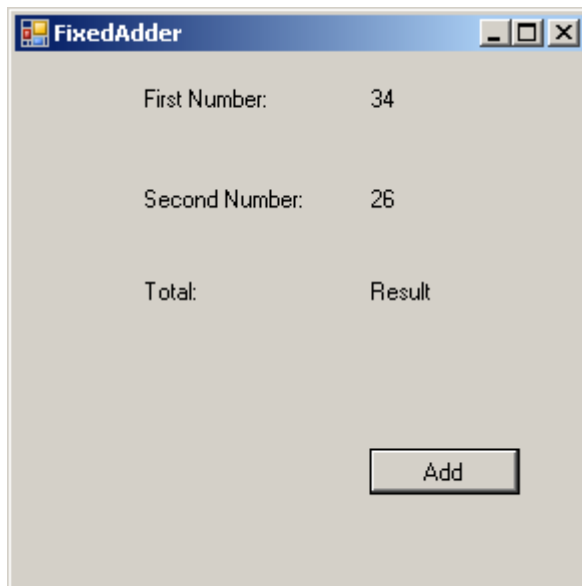
Documentation : In this activity the thoughts behind the design and implementation are written down. These explanations of the ideas behind the code are important for maintenance.

Maintenance : This is an activity that occurs after the release of the product: newly-discovered problems may need to be fixed or new/changed requirements may need to be implemented by extending or adjusting the design.

Possible stages in the design of a simple project can be summarized in the following figure :



Answer to Question 3



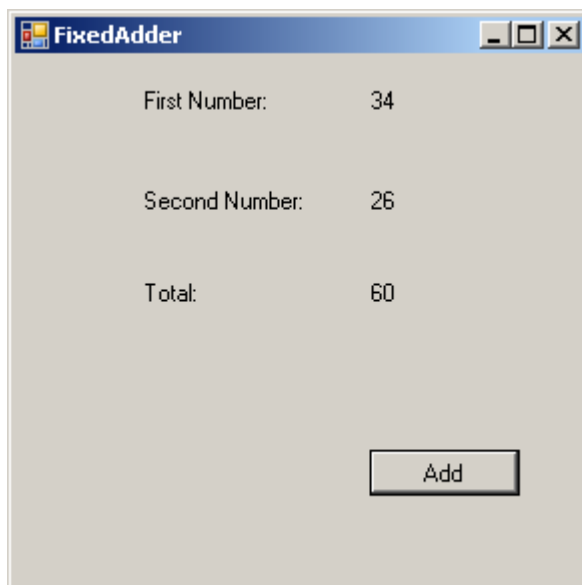
FixedAdder

First Number: 34

Second Number: 26

Total: Result

Add



FixedAdder

First Number: 34

Second Number: 26

Total: 60

Add

```
Public Class AddForm
```

```
    Private Sub AddButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AddButton.Click
```

```
        Dim FirstNum As Integer  
        Dim SecondNum As Integer
```

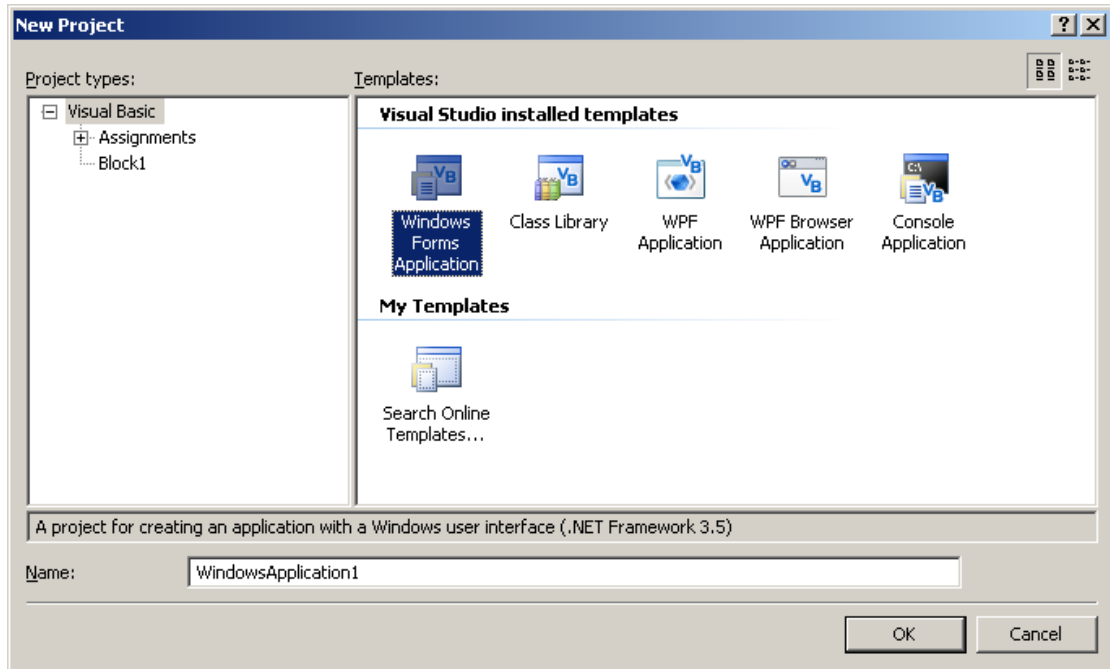
```
        FirstNum = Num1.Text  
        SecondNum = Num2.Text
```

```
        ' A statement to add user input values and assign the total.  
        Sum.Text = FirstNum + SecondNum
```

```
    End Sub  
End Class
```

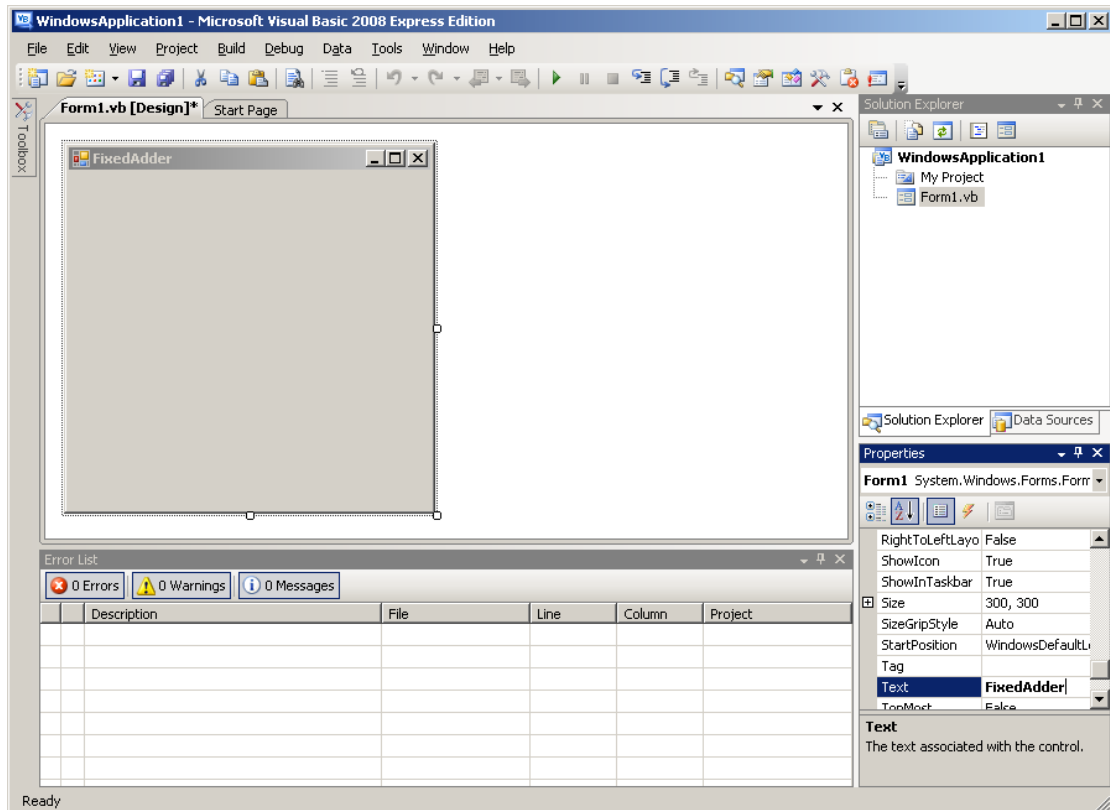
Implementation of the answer to question 3

1) File then New Project

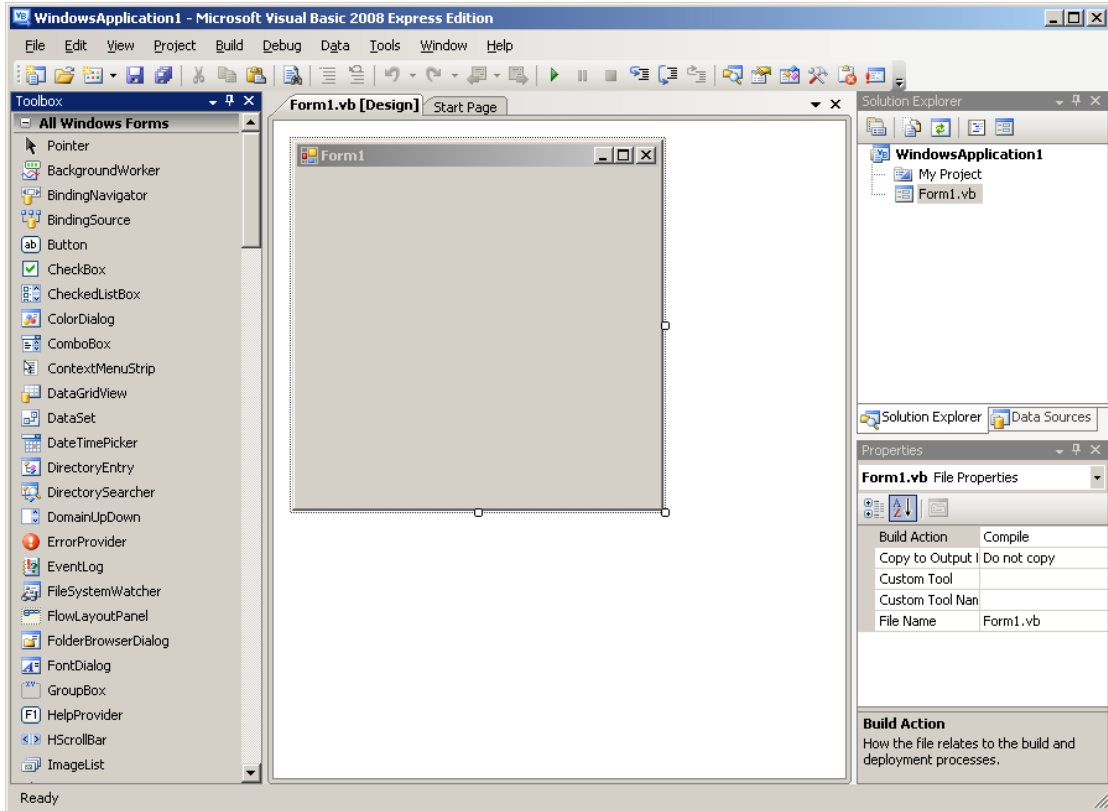


Choose “Windows Forms Application”

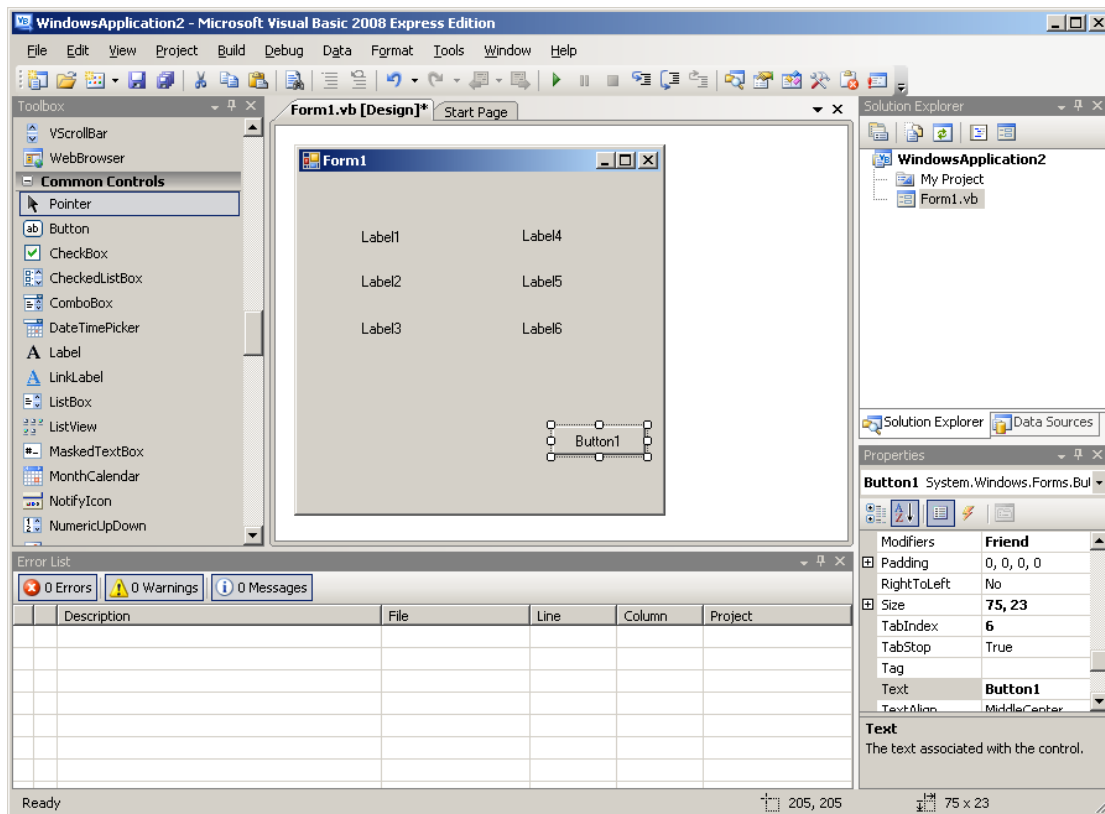
2)



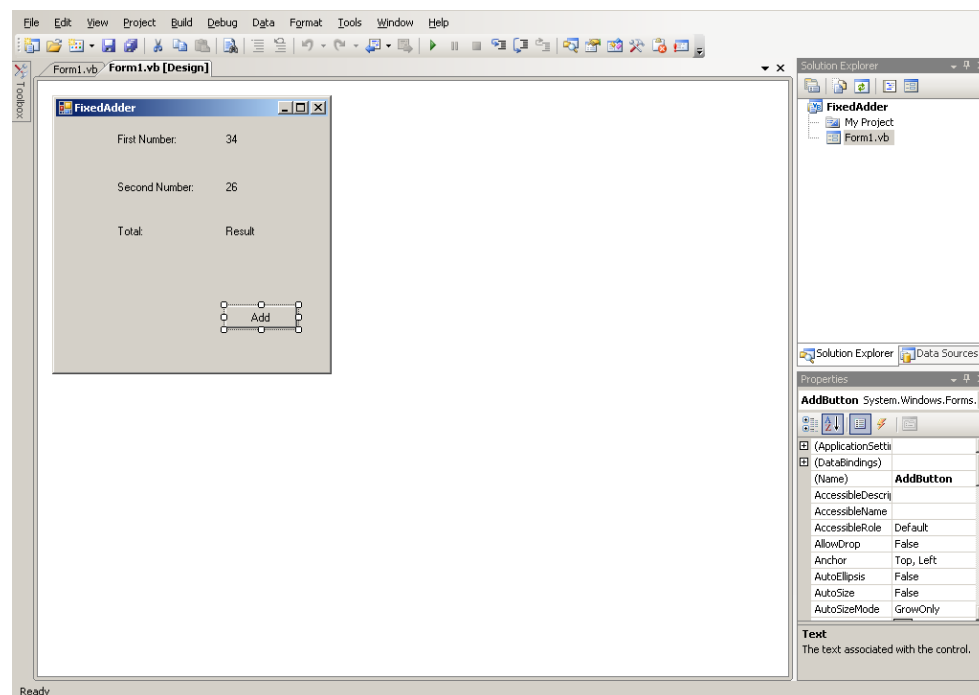
Click on Toolbar and fix it



3) In this example choose 6 Label controls and 1 Button control and drag that onto the Form1 as follows :



4) For the first 3 controls put the names by changing them from the Text property in the Label object. In Label4 change Text to a number for example 34. In the Button1 object change the Text to “Add” and the Name to “AddButton”.



5) Double click on the Add button and you get the event handler and add the following statement :

```
Dim FirstNum As Integer
Dim SecondNum As Integer

FirstNum = Num1.Text
SecondNum = Num2.Text

' A statement to add user input values and assign the total.
Sum.Text = FirstNum + SecondNum
```

So the final code should look like this :

```
Public Class AddForm

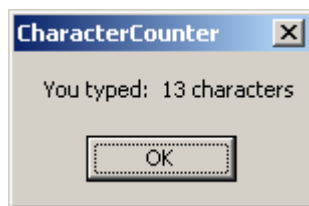
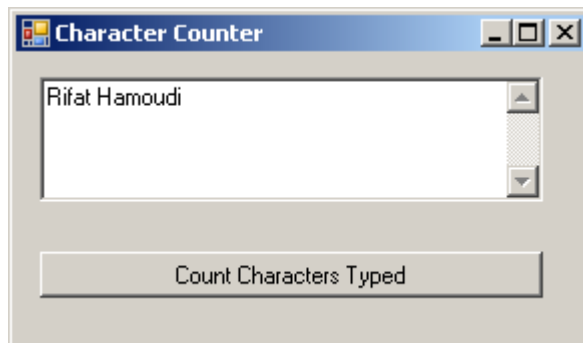
    Private Sub AddButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AddButton.Click

        Dim FirstNum As Integer
        Dim SecondNum As Integer

        FirstNum = Num1.Text
        SecondNum = Num2.Text
        Sum.Text = FirstNum + SecondNum
    End Sub
End Class
```

6) Press F5 or the green button to load the software. Click on the “Add” button and you should see the addition of the 2 numbers

Answer to Question 4



```
Public Class Form1
```

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

```
        ' Display the character count of the textbox.
        MsgBox("You typed: " & Str(Len(TextBox1.Text)) & " characters")
```

```
    End Sub
End Class
```

The above code is the efficient type hence it is shorter but less readable. A less efficient (but more readable) code is as follows :

```
Dim Lengthofstring As Integer
Dim StringLength As String
Lengthofstring = TextBox1.Text.Length
StringLength = Str(Lengthofstring)
MsgBox("You typed: " & StringLength & " characters")
```

But the most efficient code is as follows:

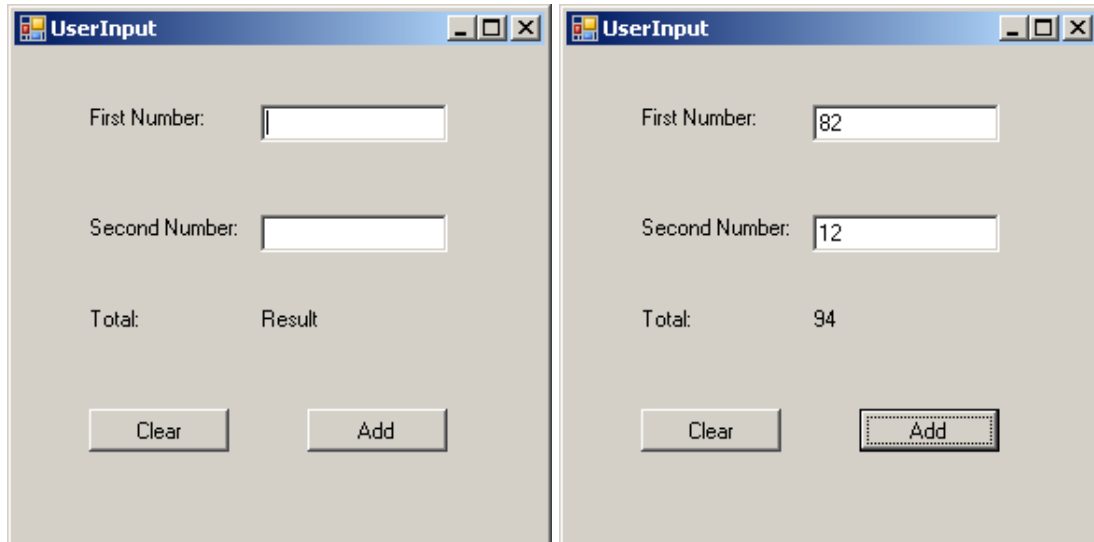
```
Public Class Form1
```

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

```
        ' Display the character count of the textbox.
        MsgBox("You typed: " & TextBox1.Text.Length.ToString & "
characters")
```

```
    End Sub
End Class
```

Answer to Question 5



```
Public Class Form1
```

```
    Private Sub AddBtn_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles AddBtn.Click
```

```
        Dim FirstNum As Integer  
        Dim SecondNum As Integer
```

```
        FirstNum = Num1.Text  
        SecondNum = Num2.Text
```

```
        ' A statement to add user input values and assign the total.  
        Sum.Text = FirstNum + SecondNum
```

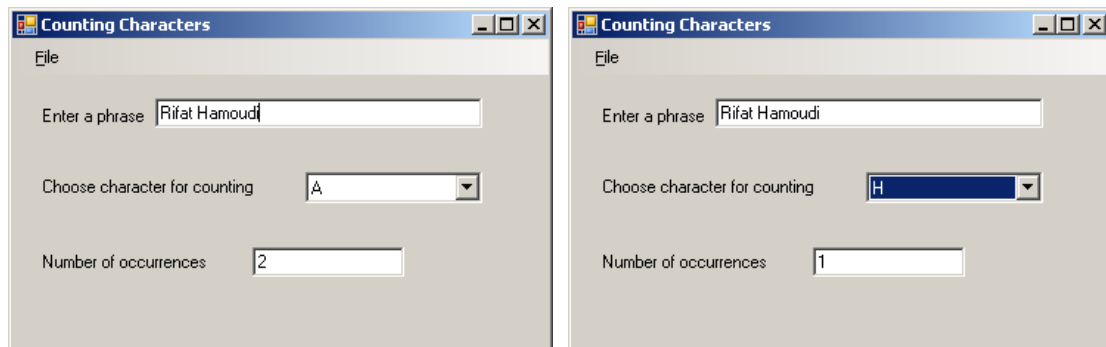
```
    End Sub
```

```
    Private Sub ClearBtn_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles ClearBtn.Click
```

```
        ' Statements to resume the initial state.  
        Sum.Text = "Result" : Num1.Text = "" : Num2.Text = ""
```

```
    End Sub  
End Class
```

Answer to Question 6



```
Imports MT264
Public Class MainForm
    Private fCharCounter As CharCounter

    Public Sub New()
        fCharCounter = New CharCounter
        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent() call.
        charComboBox.SelectedIndex = 0
    End Sub

    Private Sub phraseTextBox_TextChanged(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles phraseTextBox.TextChanged
        'Update the current string and update the view.
        fCharCounter.CurrentString = phraseTextBox.Text
        updateView()
    End Sub

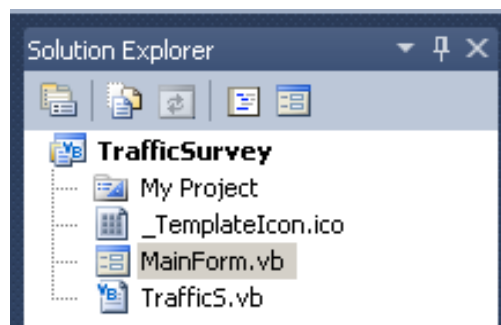
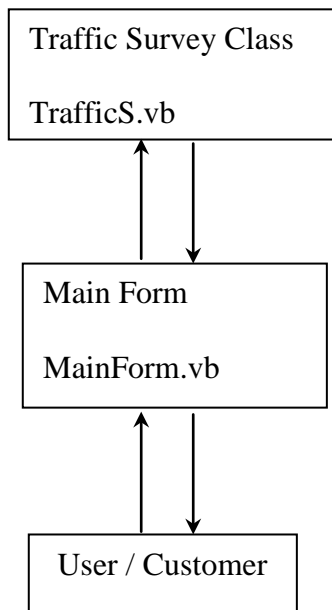
    Private Sub charComboBox_SelectedIndexChanged(ByVal sender As
        System.Object, ByVal e As System.EventArgs) Handles
        charComboBox.SelectedIndexChanged
        'Display count by updating the view.
        updateView()
    End Sub

    Private Sub updateView()
        'If the phrase box is not empty then the output box is to display
        'the number of times the selected character occurs in the
        'entered text. Otherwise the output box is made blank.
        If Not (fCharCounter.CurrentString = "") Then
            outputTextBox.Text =
fCharCounter.countChar(Convert.ToChar(charComboBox.SelectedItem))
        Else
            outputTextBox.Text = ""
        End If
    End Sub

    Private Sub exitMenuItem_Click(ByVal sender As System.Object, ByVal e
        As System.EventArgs) Handles exitMenuItem.Click
        Close()
    End Sub
End Class
```

Answer to Question 7

Software design



Initial state

The screenshot shows the "TrafficSurveyForm" application window. It has a menu bar with "File". The main area contains three input fields with corresponding buttons:

- "Number of Cars" with a text box containing "0" and a "Cars" button.
- "Number of bicycles" with a text box containing "0" and a "Bicycles" button.
- "Number of lorries" with a text box containing "0" and a "Lorries" button.

At the bottom, there is a "Total number of vehicles" label and an empty text box, with an "Add Vehicles" button to its right.

User input state

The screenshot shows the "TrafficSurveyForm" application window after user input. The input fields now contain the following values:

- "Number of Cars" with a text box containing "1" and a "Cars" button.
- "Number of bicycles" with a text box containing "2" and a "Bicycles" button.
- "Number of lorries" with a text box containing "3" and a "Lorries" button.

The "Total number of vehicles" text box now contains the value "6", and the "Add Vehicles" button is highlighted.

```

Public Class TrafficS
    ' Declare the fields here.

    Private fCars As Integer
    Private fBicycles As Integer
    Private fLorries As Integer

    Public Sub New()
        ' An instance of Traffic is created with all vehicle counts set to zero.

        fCars = 0
        fBicycles = 0
        fLorries = 0

    End Sub

    ' Setters

    Public Sub addCar()
        fCars = fCars + 1
    End Sub

    Public Sub addBicycle()
        fBicycles = fBicycles + 1
    End Sub

    Public Sub addLorry()
        fLorries = fLorries + 1
    End Sub

    'Getters

    Public ReadOnly Property getCar() As Integer
        Get
            Return fCars
        End Get
    End Property

    Public ReadOnly Property getBicycle() As Integer
        Get
            Return fBicycles
        End Get
    End Property

    Public ReadOnly Property getLorry() As Integer
        Get
            Return fLorries
        End Get
    End Property

End Class

```

```

Public Class MainForm
    Private fTrafficSurvey As TrafficS

    Public Sub New()

        fTrafficSurvey = New TrafficS

        ' This call is required by the designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent() call.
        updateView()

    End Sub

    Private Sub CarsCount_Click(sender As System.Object, e As
System.EventArgs) Handles CarsCount.Click

        fTrafficSurvey.addCar()
        Car.Text = fTrafficSurvey.getCar()
    End Sub

    Private Sub BicyclesCount_Click(sender As System.Object, e As
System.EventArgs) Handles BicyclesCount.Click

        fTrafficSurvey.addBicycle()
        Bicycle.Text = fTrafficSurvey.getBicycle()
    End Sub

    Private Sub LorriesCount_Click(sender As System.Object, e As
System.EventArgs) Handles LorriesCount.Click

        fTrafficSurvey.addLorry()
        Lorry.Text = fTrafficSurvey.getLorry()
    End Sub

    Private Sub updateView()

        Car.Text = fTrafficSurvey.getCar()
        Bicycle.Text = fTrafficSurvey.getBicycle()
        Lorry.Text = fTrafficSurvey.getLorry()
    End Sub

    Private Sub AddVehicles_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AddVehicles.Click

        Total.Text = fTrafficSurvey.getCar() + fTrafficSurvey.getBicycle()
+ fTrafficSurvey.getLorry()

    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
        'The application is closed.
        Me.Close()
    End Sub
End Class

```