

Chapter 3

Writing the Code

❖ Learning how to write Visual Basic 2010 Code

In the previous chapter, you have learned to design an interface, adding controls and setting control properties. You have also learned how to write some simple code without understanding the concepts behind. In this chapter, you will learn some basic concepts about VB2010 programming and the techniques in writing code. I will keep the theories short so that it would not be too taxing for beginners.

3.1 Understanding Event Driven Programming

VB2010 is an object oriented and event driven programming language. In fact, all windows applications are event driven. Event driven means the user decides what to do with the program, whether he or she wants to click the command button, enter text in a text box, or close the application and more. An event is related to an object, it is an incident that happens to the object due to the action of the user, such as a click or pressing a key on the keyboard. A class contains events as it creates instant of a class or an object. When we start a windows application in VB2010 in previous chapters, we will see a default form with the Form1 appears in the IDE. Form1 is the Form1 Class that inherits from the Form class `System.Windows.Forms.Form`, as shown in Figure 3.1

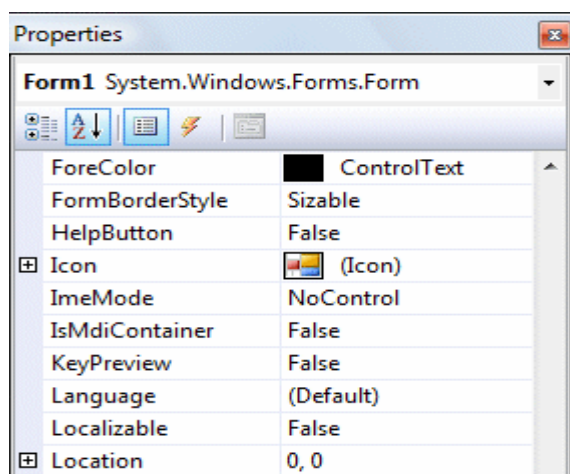


Figure 3.1: The Form1 Class

The other events associated with the Form1 class are click, DoubleClick, DragDrop, Enter and more, as shown in Figure 3.2 below (It appears when you click on the upper right pane of the code window)

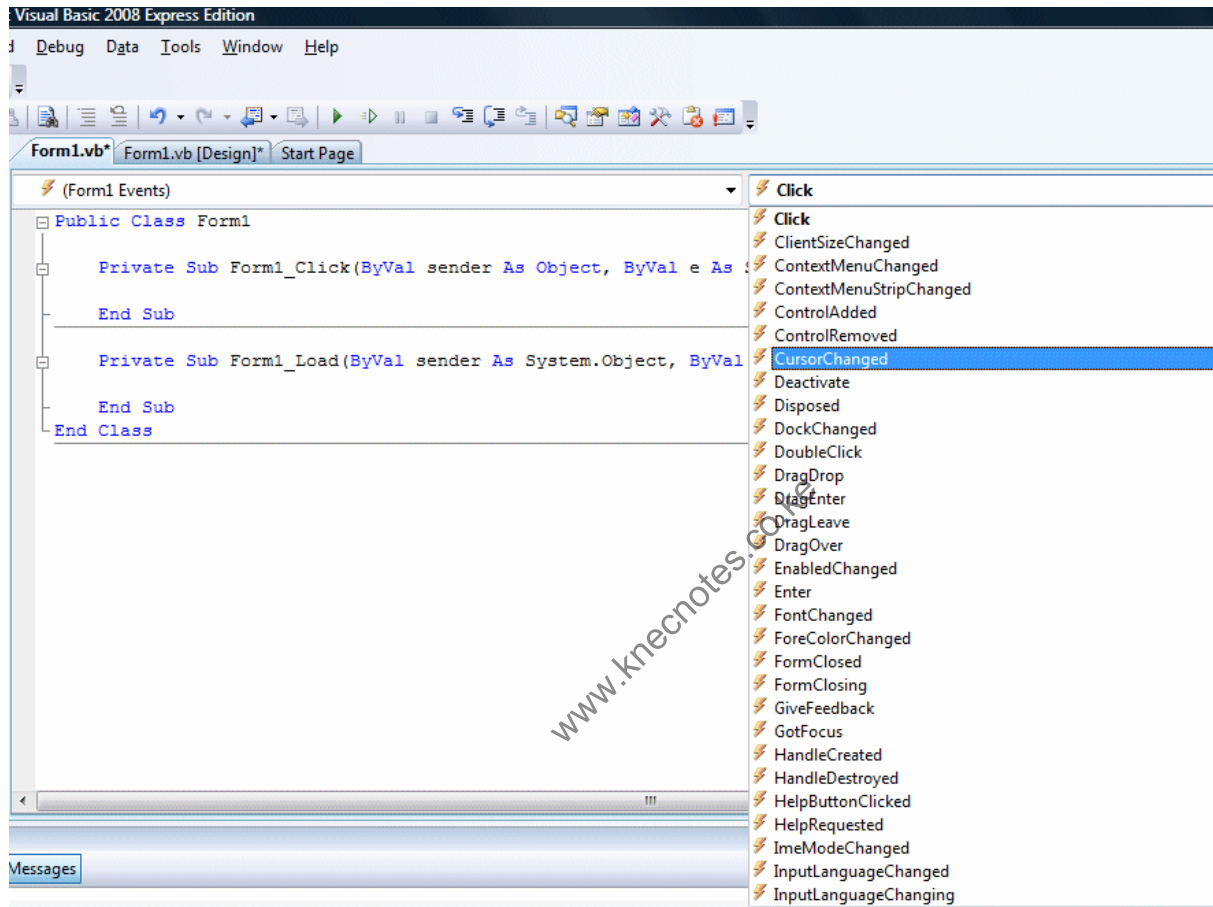


Figure 3.2

3.2 Understanding the Code Structure of an Event Procedure

Now you are ready to write the code for the event procedure so that it will do something more than loading a blank form. The structure of the code takes the following form:

Private Sub...

Statements

End Sub

You have to enter the code between Private Sub and End Sub.

Private Sub

Enter your code here

End Sub.

There are variations of the structure such as

i) **Public Sub**

Enter your code here

End Sub.

ii) **Sub**

Enter your code here

End Sub.

iii) **Function**

Enter your code here

End Function

Let us enter the following code:

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

```
    Me.Text="My First VB2010 Program"
```

```
    Me.ForeColor = Color.Yellow
```

```
    Me.BackColor = Color.Blue
```

```
End Sub
```

When you press F5 to run the program, the output is shown in Figure 3.3 below:

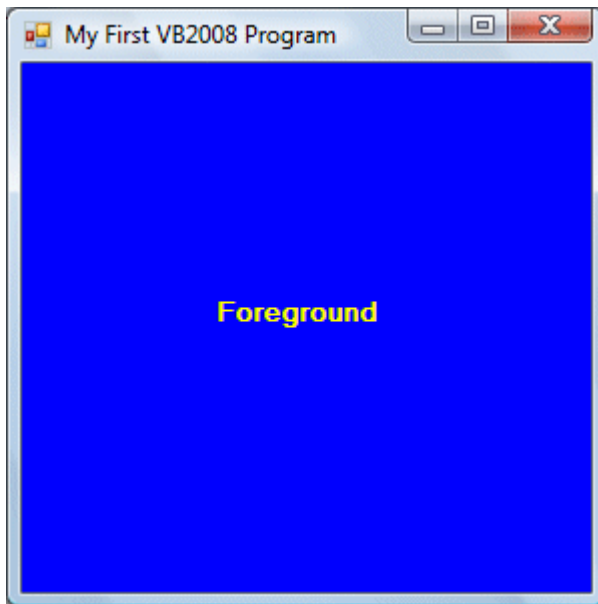


Figure 3.3: The Output Window

The first line of the code will change the title of the form to “My First VB2010 Program”, the second line will change the foreground object to yellow(in this case, it is a label that you insert into the form and change its name to Foreground) and the last line changes the background to blue color. The equal sign in the code is to assign something to the object, like assigning yellow color to the foreground of the Form1 object (or an instance of Form1). **Me** is the name given to the Form1 class. We can also call those lines as Statements. Therefore, the actions of the program will depend on the statements entered by the programmer. Here is another example.

```
Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
    Dim name1, name2, name3 As String
    name1 = "John"
    name2 = "Chan"
    name3 = "Ali"
    MsgBox(" The names are " & name1 & " , " & name2 & " and " & name3)
```

```
End Sub
```

In this example, you insert one command button into the form and rename its caption as Show Hidden Names. The keyword Dim is to declare variables name1, name2 and name3 as string, which means they can only handle text. The function MsgBox is to

display the names in a message box that are joined together by the "&" signs. The output is shown in Figure 3.4 below:

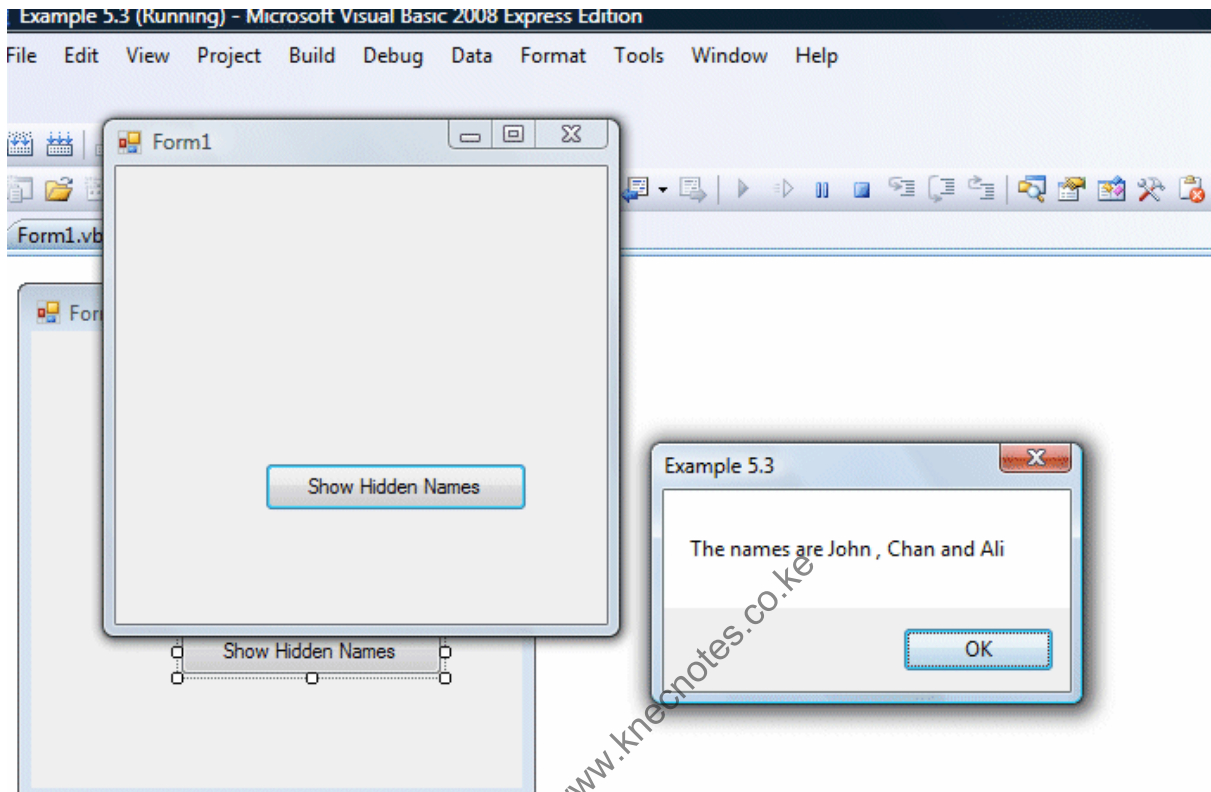


Figure 3.4: The Output Window for Displaying Names

3.3 Writing a Simple Multiplication Program

In this program, you insert two text boxes, three labels and one button. The text boxes are for the user to enter numbers, the label is to display the multiplication operator and the other label is to display the equal sign. The last label is to display the answer. The run time interface is shown in Figure 3.5

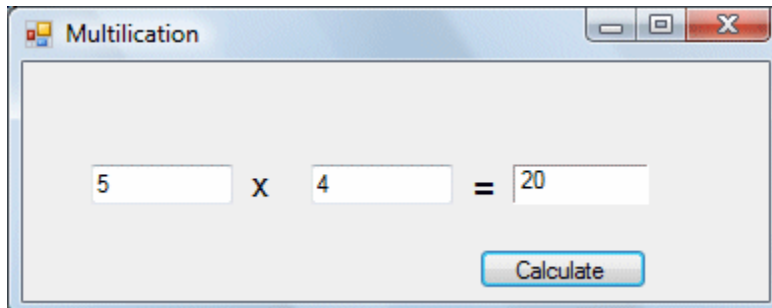


Figure 3.5: The Multiplication Program

The Code

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

```
Dim num1, num2, product As Single
num1 = TextBox1.Text
num2 = TextBox2.Text
product = num1 * num2
Label3.Text = product
```

```
End Sub
```

3.4 Writing a Program that Add Items to a List Box

This program will add one item at a time to a list box as the user enters an item into the text box and click the Add button. In this program, you insert a TextBox and a ListBox into the Form. The function of the TextBox is to let the user enter an item one at a time and add it to the Listbox. The method to add an item to the ListBox is Add. The output interface is shown in Figure 3.6.

The Code

```
Class Frm1
```

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

```
Dim item As String
item = TextBox1.Text

'To add items to a listbox
ListBox1.Items.Add(item)
```

```
End Sub
```

```
End Class
```

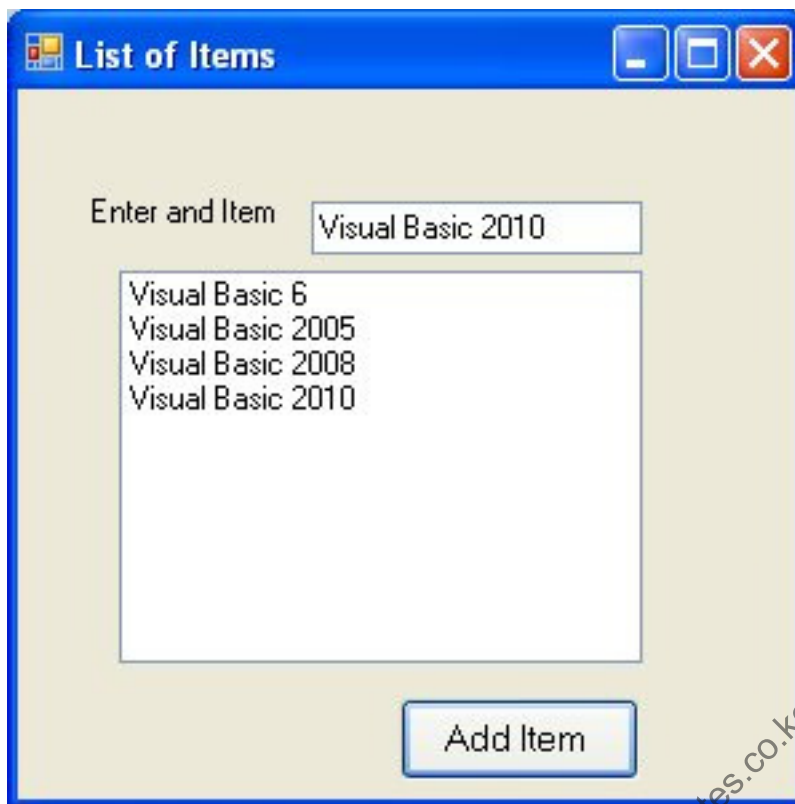


Figure 3.6: The Add Items Program

Summary

- In section 3.1, you learned the concept of event driven programming.
- In section 3.2, you learned how to write a simple code for an event procedure, including the usage of MsgBox().
- In section 3.3, you learned how to create a multiplication program.
- In section 3.4, you learned how to write a program to add some items to a list box.