Visual Basic 2008

Made Easy

Dr. Liew Voon Kiong

Disclaimer

Visual Basic 2008 ® Made Easy- A complete tutorial for beginners is an independent publication and is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.

Trademarks

Microsoft, Visual Basic, Excel and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Liability

The purpose of this book is to provide a basic guide for people interested in Visual Basic® programming. Although every effort and care has been taken to make the information as accurate as possible, the author shall not be liable for any error, harm or damage arising from using the instructions given in this book.

Acknowledgement

I would like to express my sincere gratitude to many people who have made their contributions in one way or another to the successful publication of this book.
My special thanks go to my children Xiang, Yi and Xun. My daughter Xiang edited this book while my sons Yi and Xun contributed their ideas and even wrote some of the sample programs for this book. I would also like to appreciate the support provided by my beloved wife Kim Huang and my youngest daughter Yuan. I would also like to thank the million of visitors to my **Visual Basic Tutorial** website at www.vbtutor.net, especially those who contributed their comments, for their support and encouragement.

About the Author

Dr. Liew Voon Kiong holds a bachelor's degree in Mathematics, a master's degree in Management and a doctorate in Business Administration. He has been involved in Visual Basic programming for more than 20 years. He created the popular online Visual Basic Tutorial at **www.vbtutor.net**, which has attracted millions of visitors since 1996. It has consistently been one of the highest ranked Visual Basic websites.

To provide more support for Visual Basic students, teachers, and hobbyists, Dr. Liew has written this book to complement the free Visual Basic 2008 tutorial with much more content. He is also the author of the Visual Basic Made Easy series, which includes **Visual Basic 6 Made Easy, Visual Basic 2008 Made Easy, Visual Basic 2010 Made Easy, Visual Basic 2013 Made Easy, Visual Basic 2015 , Visual Basic 2017 Made Easy** and **Excel VBA Made Easy.** Dr. Liew's books have been used in high school and university computer science courses all over the world.

TABLE OF CONTENTS

Chapter 1

Introduction to Visual Basic 2008

- A brief description of Visual Basic 2008
- Getting to know the Visual Basic 2008 Integrated Development Environment

1.1 A brief description of Visual Basic 2008

Visual Basic is a third-generation event-driven programming language first released by Microsoft in 1991. Visual Basic is a user-friendly programming language designed for beginners. Therefore, It enables anyone to develop GUI window applications easily. Visual Basic has gone through many phases of development since the days of BASIC that was built for DOS . BASIC stands for Beginners' All-purpose Symbolic Instruction Code. The program syntax in Visual Basic resembles the English language. Since the release of Visual Basic Version1 in 1991, the DOS versions of BASIC were slowly phased out and completely replaced by Visual Basic . The final version was Visual Basic 6.

In 2002, Microsoft released Visual Basic.NET(VB.NET) to replace Visual Basic 6. Thereafter, Microsoft declared VB6 a legacy programming language in 2008. However, Microsoft still provides some form of support for VB6. VB.NET is a fully object-oriented programming language implemented in the .NET Framework. It was created to cater for the development of the web as well as mobile applications. Visual Basic 2008 is the VB.NET version launched by Microsoft in the year 2008. It is almost similar to the earlier VB.NET version, Visual Basic 2005 and but it has added many new features.

Visual Basic 2008 is a full-fledged Object-Oriented Programming (OOP) Language, so it has caught up with other OOP languages such as C++, Java, C# and others.

However, you don't have to know OOP to learn Visual Basic 2008. In fact, if you are familiar with Visual Basic 6, you can learn Visual Basic 2008 effortlessly because the syntax and interface are almost similar. Visual Basic 2008 Express Edition is available free for download from the Microsoft site using the following link: https://go.microsoft.com/fwlink/?LinkId=104679

1.2 The Visual Basic 2008 Integrated Development Environment

When you launch Visual Basic 2008 Express, the Integrated Development Environment will be presented to you, as shown in Figure 1-1.

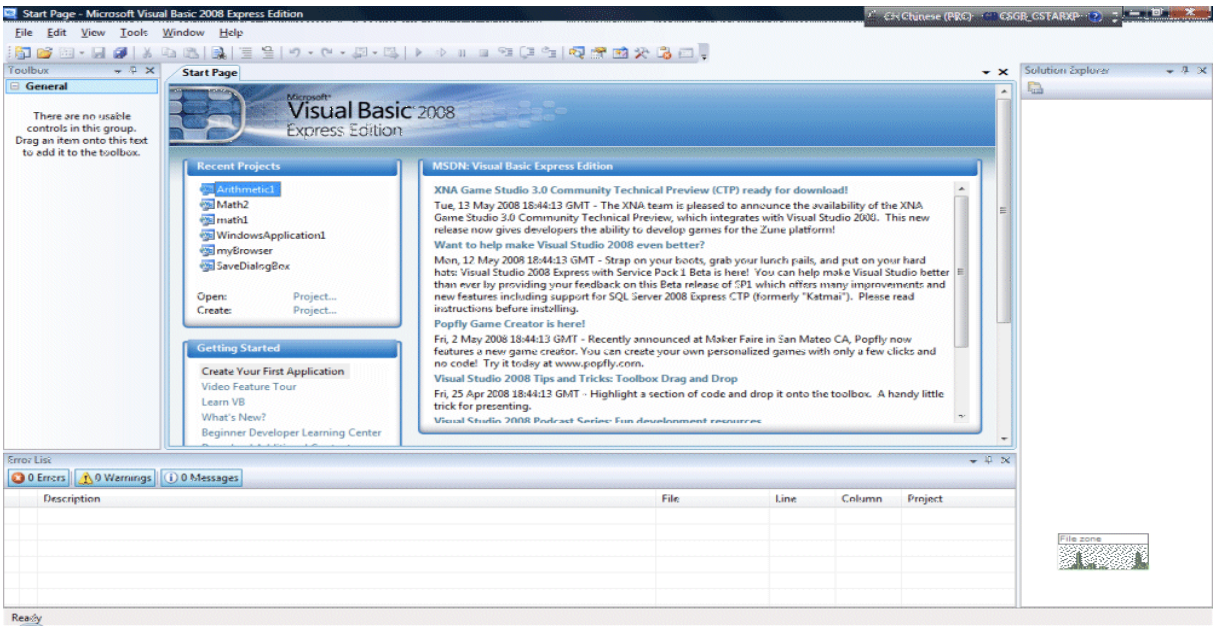

Figure 1-1: The Visual Basic 2008 IDE

The Visual Basic 2008 IDE consists of a few panes, namely:

- The Recent Projects Pane- it shows the list of projects that have been created by you recently.
- The Getting Started Pane- It provides some helpful tips to quickly develop your applications
- The VB Express Headlines pane- It provides latest online news about Visual Basic 2008 Express. It will announce new releases and updates.

To start creating your first application, you need to click on File on the menu bar and select New Project. The Visual Basic 2008 New Project dialog box will appear, as shown in Figure 1-2

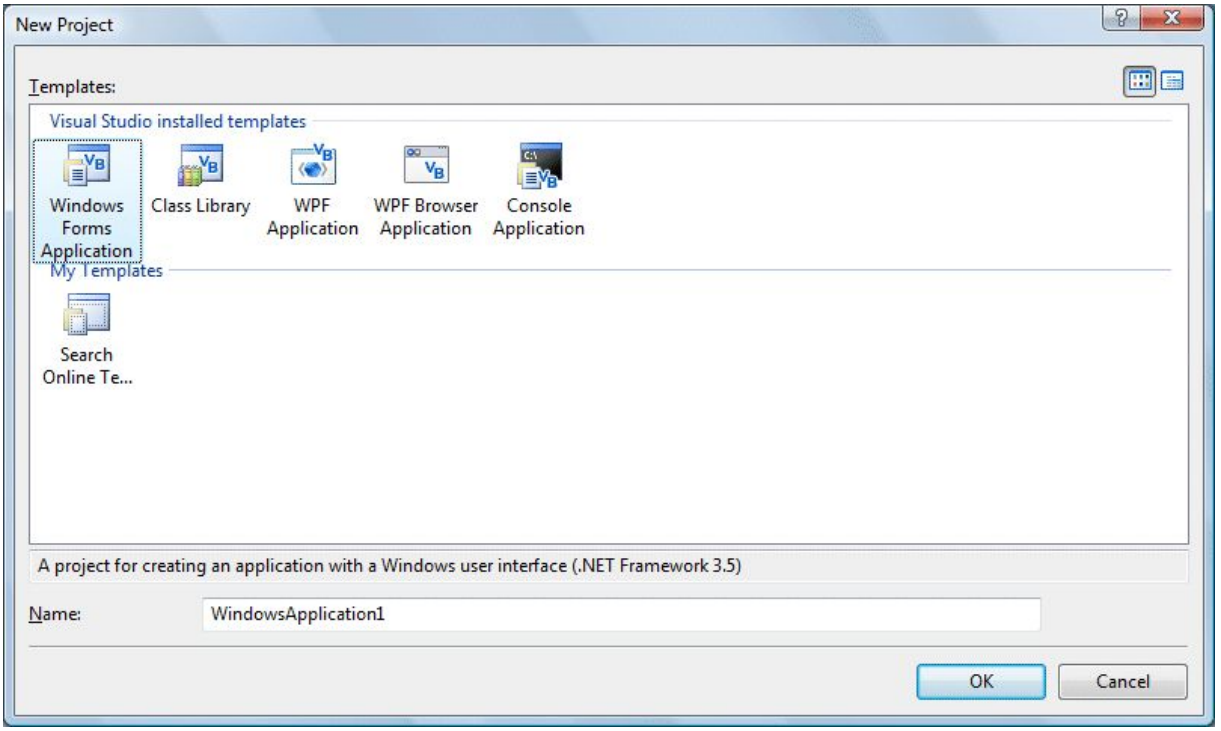

Figure 1-2: Visual Basic 2008 New Project Dialog Box

The dialog box offers you five types of projects that you can create. As we are going to create a Windows application, we will select Windows Forms Application. At the bottom of this dialog box, you can change the default project name **WindowsApplication1** to some other name you like, for example, **MyFirstProgram**. After having renamed the project, click OK to continue, the Visual Basic 2008 IDE with a new Form will appear, as shown in Figure 1-3. It consists of an empty form, the common controls toolbox, the solution explorer and the Properties Window.

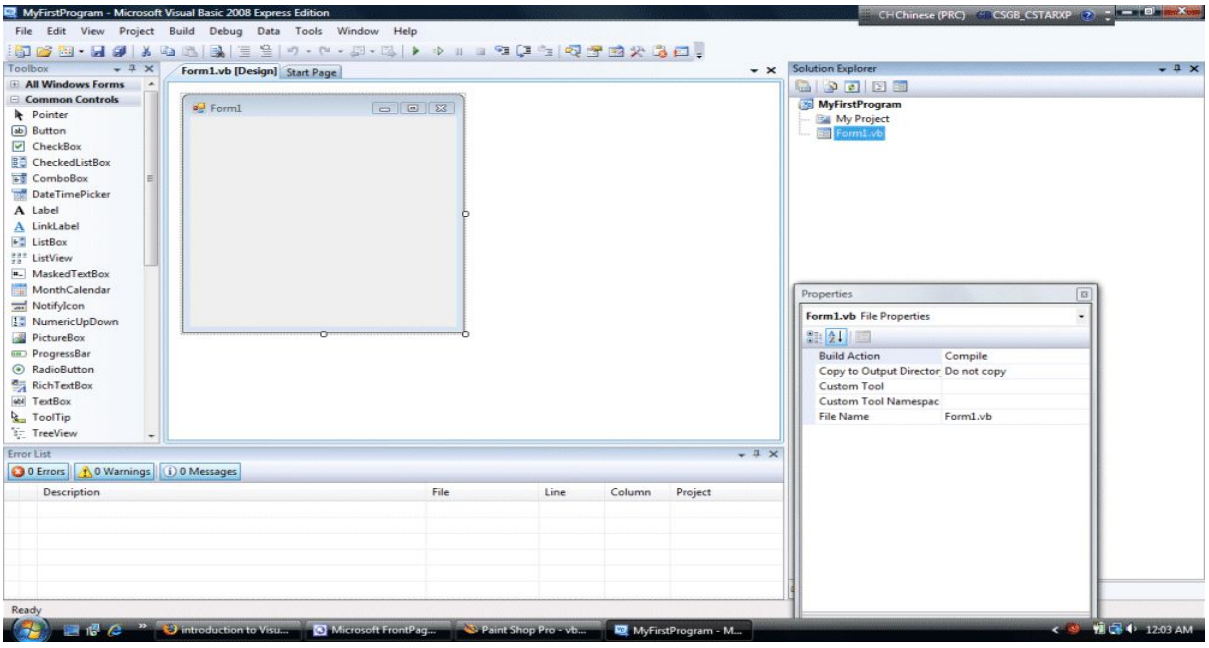

Figure 1-3: Visual Basic 2008 IDE with A New Form

Now let's create your first Visual Basic 2008 program. First of all, drag one common button into the form and change its default name to calculate, as shown in Figure 1-4.

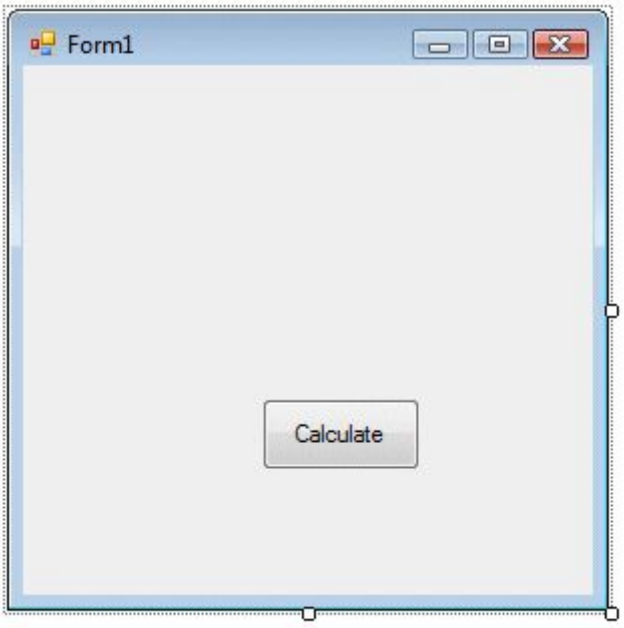

Figure 1-4 The design Interface

Next, click on the calculate button and enter the following code at the source code window as shown in Figure 1-5.

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
Dim num1, num2, sum As Single
num1 = 100
num2 = 200
sum = num1 + num2
MsgBox (" The Sum of" & num1 & " and "& num2 & "is " & sum)
End Sub
```

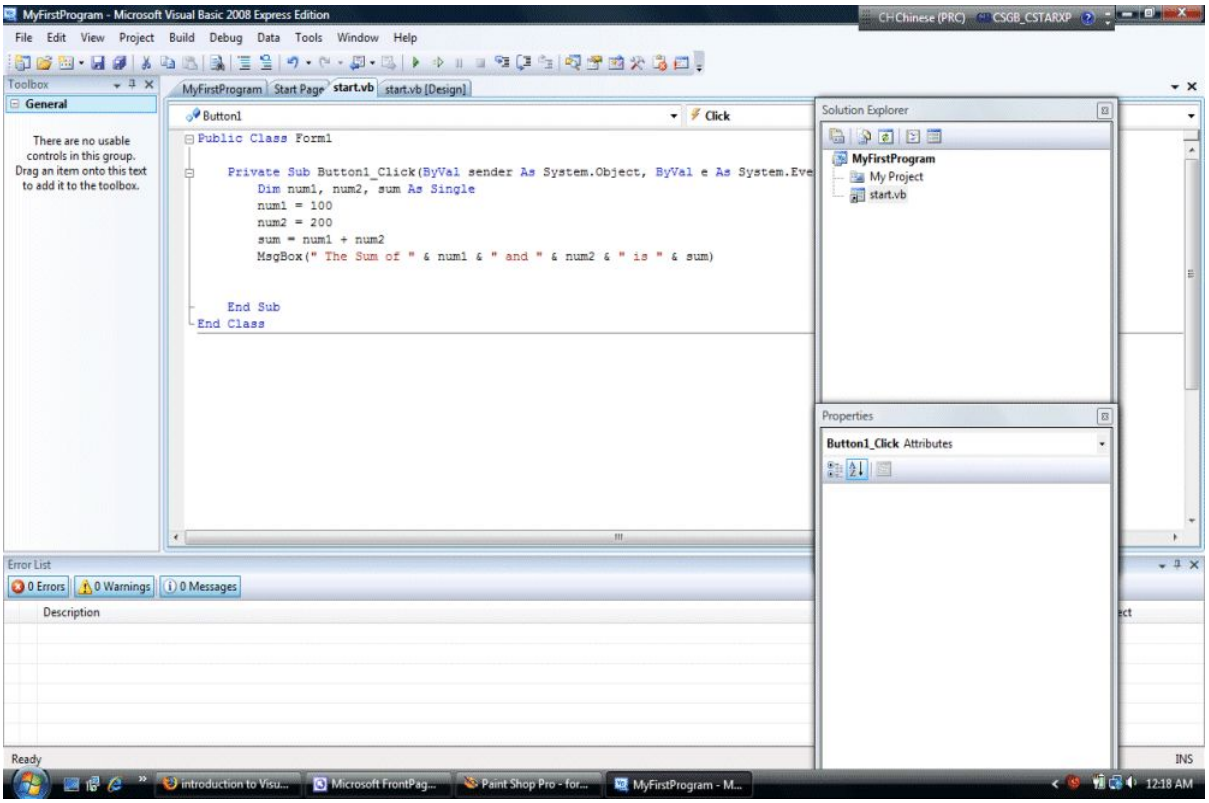

Figure 1-5: The Code window

Now run your first application! And you can see the follow message box showing the sum of two numbers, as illustrated in Figure 1-6.

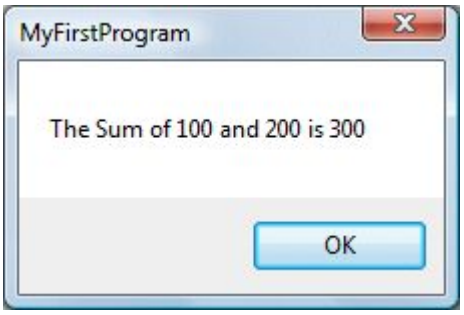

Figure 1-6: The Output Window

Chapter 2 Working with Controls

- ❖ Getting to know Visual Basic 2008 controls
- ❖ Getting to know the Control Properties.

The Visual Basic 2008 Common Control Toolbox consists of all the controls essential for developing a VISUAL BASIC 2008 application. The Controls in Visual Basic 2008 are useful tools that can be placed in the form to perform various tasks. They are used to create many kinds of Windows applications. They are categorized into Common Controls, Containers, Menus, Toolbars, Data, Components, Printings and Dialogs. At the moment, we will focus on the common controls. Some of the most used common controls are Button, Label, ComboBox, ListBox, PictureBox and TextBox, as shown in Figure 2-1

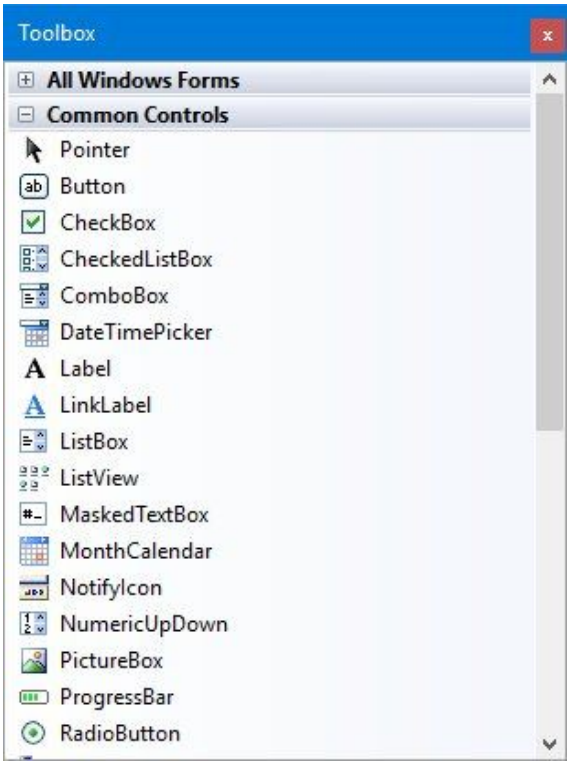

Figure 2-1 The Toolbox

To insert a control into your form, you drag the control and drop it into the form. You can reposition and resize it as you like. Let's examine a few programs that made use of Button, Label, TextBox, ListBox and PictureBox. You don't have to worry so much about the code because we will explain the program syntax as you progress to later Chapters.

2.1 Using Text Box

In this program, you insert two textboxes, three labels and one button. The two textboxes are for the users to enter two numbers, one 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 2-2

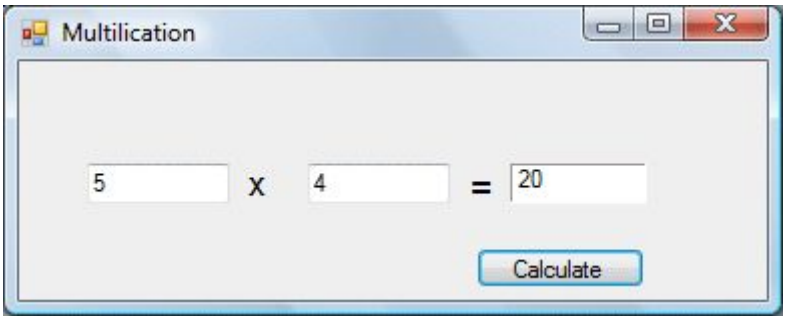

Figure 2-2 The Multiplication Program

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
```

2.2 Using List Box

This program will add one item at a time as the user enters an item into the TextBox 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 as shown in Figure 2-3.

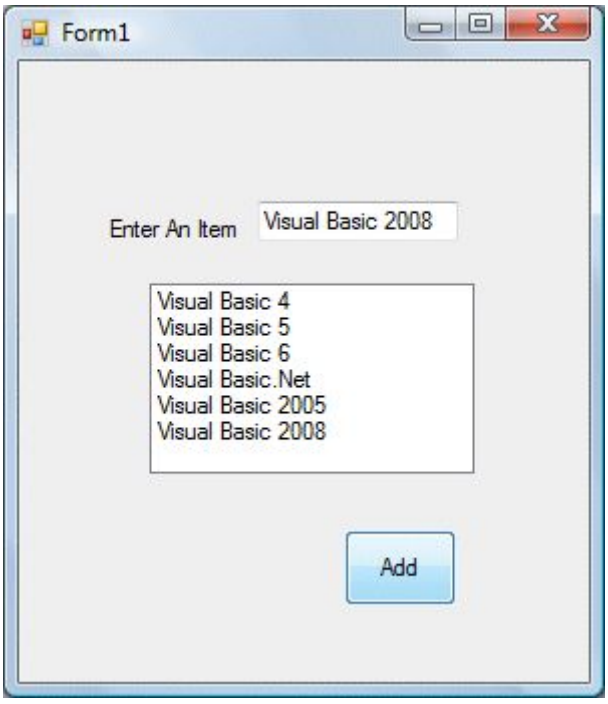

Figure 2-3: The Add Items Program

The Code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
Dim item As String
item = TextBox1.Text
ListBox1.Items.Add(item)
End Sub
```

Chapter 3

Working with Control Properties

- ❖ Setting the properties of the controls

3.1 Setting Control Properties in the Properties Window

Before writing an event procedure for the control to response to a user's input, you have to set certain properties for the control to customize its appearance and the way it will work with the event procedure. You can set the properties of the controls in the properties window at design time or you can set the properties using code

The typical properties window for a form as shown in Figure 3.1. The title of the form is defined by the Text property and its default name is Form 1. To change the form's title to any name that you like, simple click in the box on the right of the Text property and type in the new name, in this example, the title is Multiplication. Notice that this title will appear on top of the windows. In the properties window, the item appears at the top part is the object currently selected (in Figure 3.1, the object selected is Form1). At the bottom part, the items listed in the left column represent the names of various properties associated with the selected object while the items listed in the right column represent the states of the properties. Properties can be set by highlighting the items in the right column then change them by typing or selecting the options available. You may also alter other properties of the form such as font, location, size, foreground color, background color, MaximizeBox, MinimizeBox and more.

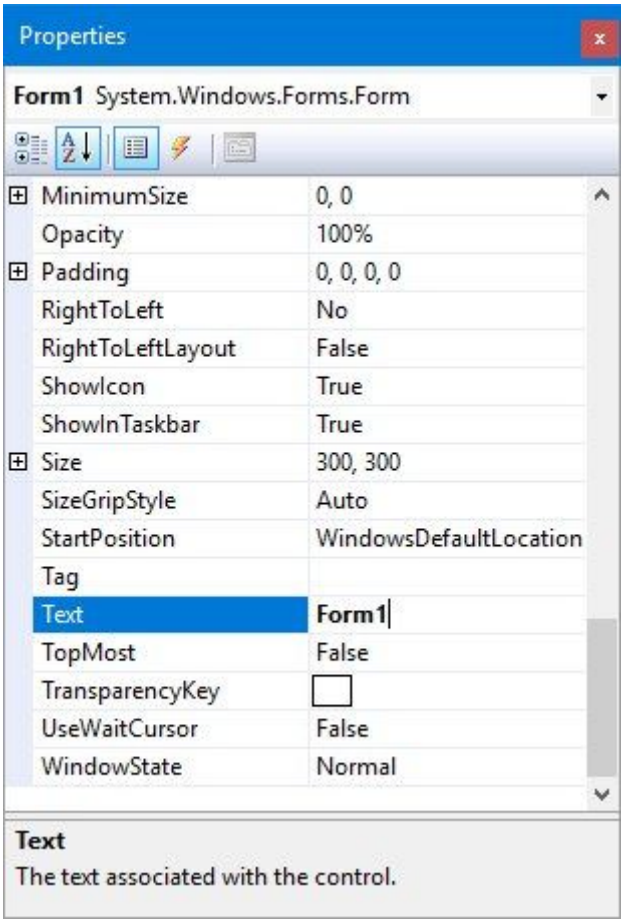

Figure 3.1 The Properties Windows

For example, when you select background color, the dialog box for color selection will appear, as shown in Figure 3-2. You can then select any color by clicking on one of the colors.

Figure 3-2: Color Selection

3.2 Setting Control Properties using Code

You can also change the properties of the object at run time to give special effects such as change of color, change of shape, animation effect and more. For example the following code will change the form color to yellow every time the form is loaded. VB2008 uses RGB(Red, Green, Blue) to determine the colors. The RGB code for yellow is 255,255,0. Me in the code refer to the current form and Backcolor is the property of the form's background color. The formula to assign the RGB color to the form is Color.FormArbg(RGB code).

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.BackColor = Color.FromArgb(255, 255, 0)
End Sub
```

You may also use the follow procedure to assign the color at run time.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
      Me.BackColor = Color.Yellow
End Sub
```

Both procedures above will load the form with a yellow background, as shown in Figure 3-3

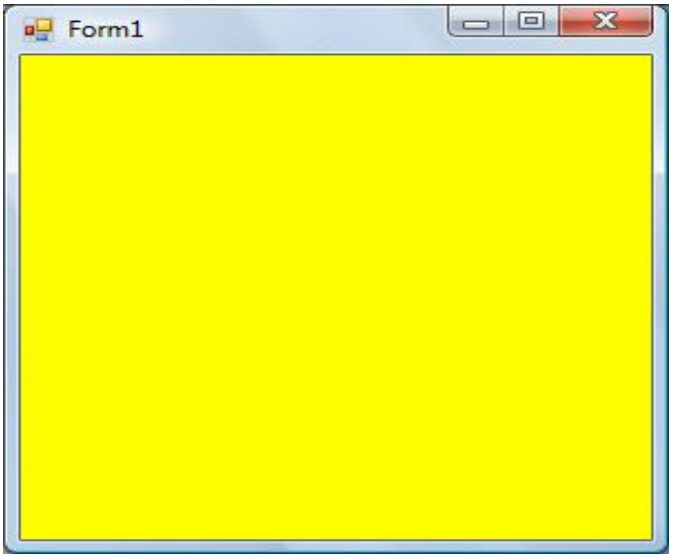

Figure 3-3: The form with yellow background

Here are some of the common colors and the corresponding RGB codes. You can always experiment with other combinations, but remember the maximum number for each color is 255 and the minimum number is 0. The table below shows some of the common colors with their corresponding codes.

Table 3-1: Some common colors and their corresponding RGB codes

Color	RGB code	Color	RGB code	Color	RGB Code
	255,0,0		255, 255, 0		255, 165, 0
	0,255,0		0, 255, 255		0, 0, 0
	0, 0, 255		255, 0, 255		255, 255, 255

The following is a program that allows the user to enter the RGB codes into three different Textboxes and when he/she clicks the display color button, the background color of the form changes according to the RGB code. So, this program allows the user to change the color properties of the form at run time.

The code

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
Dim rgb1, rgb2, rgb3 As Integer
rgb1 = TextBox1.Text
rgb2 = TextBox2.Text
rgb3 = TextBox3.Text
Me.BackColor = Color.FromArgb(rgb1, rgb2, rgb3)
End Sub
```

The output interface is shown in Figure 3-4

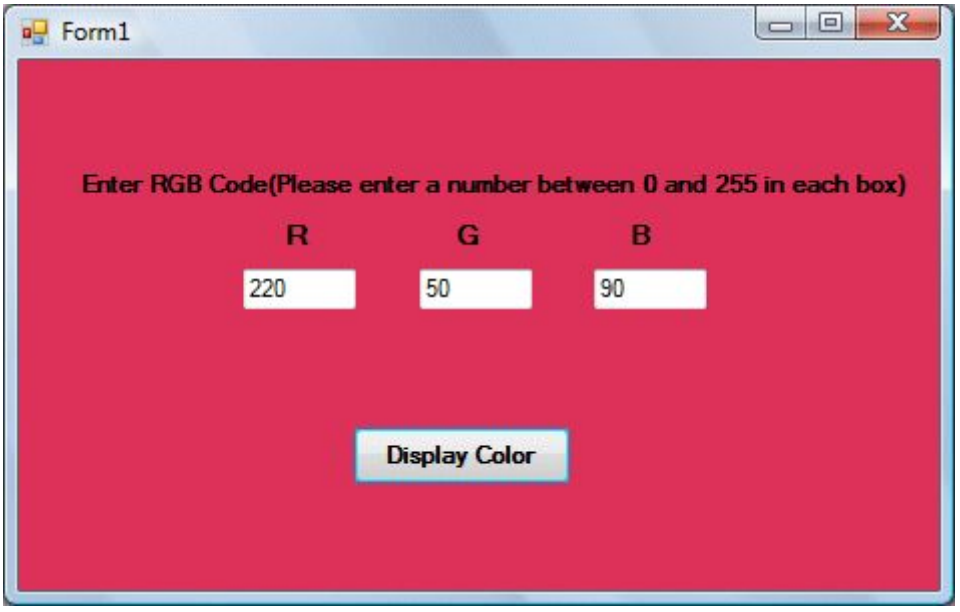

Figure 3-4: The RGB Program

Chapter 4

Object Oriented Programming

- Learning about the Concept of Object Oriented Programming

In first three Chapters, you have learned how to enter the program code and run the sample Visual Basic 2008 programs but without understanding much about the logics of Visual Basic 2008 programming. Now, let's get down to learning a few basic rules about writing the Visual Basic 2008 program code.

First, let me say that though Visual Basic 2008 is very much similar to VB6 in terms of Interface and program structure, their underlying concepts are quite different. The main different is that VB2008 is a full Object Oriented Programming Language while VB6 may have OOP capabilities, it is not fully object oriented. In order to qualify as a fully object oriented programming language, it must have three core technologies namely **encapsulation, inheritance** and **polymorphism**. These three terms are explained below:

4.1 Encapsulation

Encapsulation refers to the creation of self-contained modules that bind processing functions to the data. These user-defined data types are called classes. Each class contains data as well as a set of methods which manipulate the data. The data components of a class are called instance variables and one instance of a class is

an object. For example, in a library system, a class could be member, and John and Sharon could be two instances (two objects) of the library class.

4.2 Inheritance

Classes are created according to hierarchies, and inheritance allows the structure and methods in one class to be passed down the hierarchy. That means less programming is required when adding functions to complex systems. If a step is added at the bottom of a hierarchy, then only the processing and data associated with that unique step needs to be added. Everything else about that step is inherited. The ability to reuse existing objects is considered a major advantage of object technology.

4.3 Polymorphism

Object-oriented programming allows procedures about objects to be created whose exact type is not known until runtime. For example, a screen cursor may change its shape from an arrow to a line depending on the program mode. The routine to move the cursor on screen in response to mouse movement would be written for "cursor," and polymorphism allows that cursor to take on whatever shape is required at runtime. It also allows new shapes to be easily integrated.

VB6 is not a full OOP in the sense that it does not have inheritance capabilities although it can make use of some benefits of inheritance. However, VB2008 is a fully functional Object Oriented Programming Language, just like other OOP such as C++ and Java. It is different from the earlier versions of VB because it focuses more on the data itself while the previous versions focus more on the actions. Previous versions of VB are known as **procedural** or **functional** programming language. Some other procedural programming languages are C, Pascal and Fortran.

Visual Basic 2008 allows users to write programs that break down into modules. These modules will represent the real-world objects and are known as classes or types. An object can be created out of a class and it is known as an instance of the

class. A class can also comprise subclass. For example, apple tree is a subclass of the plant class and the apple in your backyard is an instance of the apple tree class. Another example is student class is a subclass of the human class while your son John is an instance of the student class. A class consists of data members as well as methods. In Visual Basic 2008, the program structure to define a Human class can be written as follows:

```
Public Class Human
'Data Members
Private Name As String
Private Birthdate As String
Private Gender As String
Private Age As Integer
'Methods
Overridable Sub ShowInfo( )
MessageBox.Show(Name)
MessageBox.Show(Birthdate)
MessageBox.Show(Gender)
MessageBox.Show(Age)
End Sub
End Class
```

After you have created the human class, you can create a subclass that inherits the attributes or data from the human class. For example, you can create a students class that is a subclass of the human class. Under the student class, you don't have to define any data fields that are already defined under the human class; you only have to define the data fields that are different from an instance of the human class. For example, you may want to include StudentID and Address in the student class. The program code for the StudentClass is as follows:

```
Public Class Students
Inherits Human
Public StudentID As String
Public Address As String
Overrides  Sub ShowInfo( )
MessageBox.Show(Name)
MessageBox.Show(StudentID)
```

```
MessageBox.Show(Birthdate)
MessageBox.Show(Gender)
MessageBox.Show(Age)
MessageBox.Show(Address)
End Sub
```

Chapter 5 Writing the Code

- ❖ Learning how to write Visual Basic 2008 Code

In previous chapter, you have learned that Visual Basic 2008 is an object oriented programming language. You have understood the meanings of class, object, encapsulation inheritance as well as polymorphism. You have also learned how to write some simple programs without much understanding of the underlying foundations and theories. In this chapter, you will learn some basic theories about Visual basic B2008 programming but we will focus more on learning by doing, i.e. learning by writing code.

5.1 The event Procedure

Visual Basic 2008 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, enters text in a text box, or closes the application and more. An event is related to an object, it is something that happens to the object due to the action of the user, such as clicking the mouse button or pressing a key on the keyboard. A class is consists of events as it creates instant of a class or an object. When we start a windows application in Visual Basic 2008 in previous chapters, we will see a default form with the default name Form1 appears in the IDE. Form 1 is the Form1 Class that inherits from the Form class System.Windows.Forms.Form, as shown in Figure 5-1

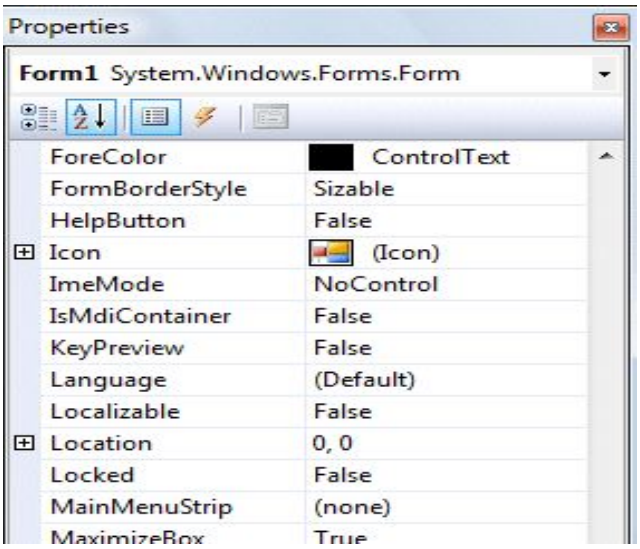

Figure 5-1: The Form1 Class

The other events associated with the Form1 class are click, DoubleClick, DragDrop, Enter as more, as shown in Figure 5-2 below

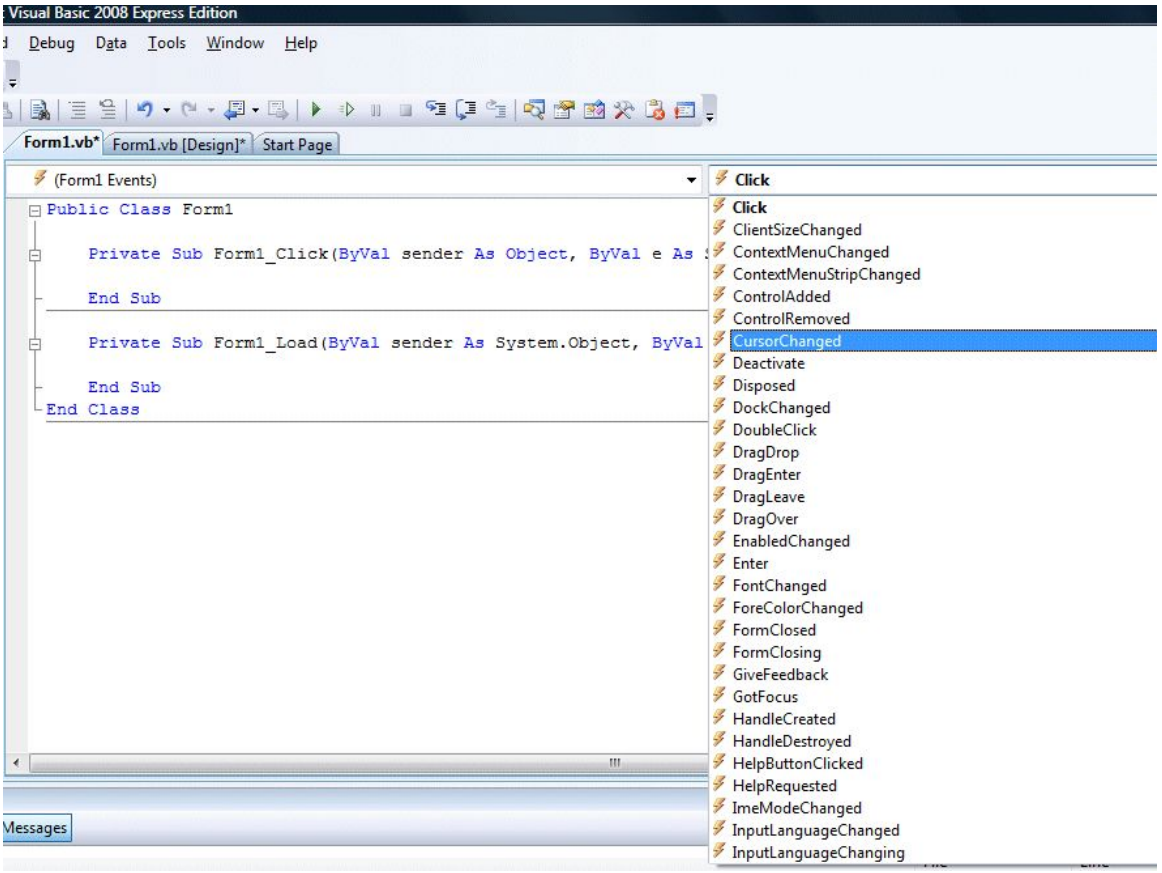

Figure 5-2: Events associated with the Form1 class in the drop-down list

5.2 Writing the code

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 code must be entered between Private Sub.....End Sub.

Example 5.1

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Me.Text="My First VB2008 Program"
Me.ForeColor = Color.Yellow
Me.BackColor = Color.Blue
End Sub
```

The output is shown in Figure 5-3 below:

Figure 5-3: The Output Window

The first line of the code changes the title of the form to 'My First VB2008 Program', the second line changes 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 symbol = in the code is used to assign values or properties 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 Statements. So, the actions of the program will depend on the statements entered by the programmer. Here is another example.

Example 5.2

```
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 Example 5.2, 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 5-4 below:

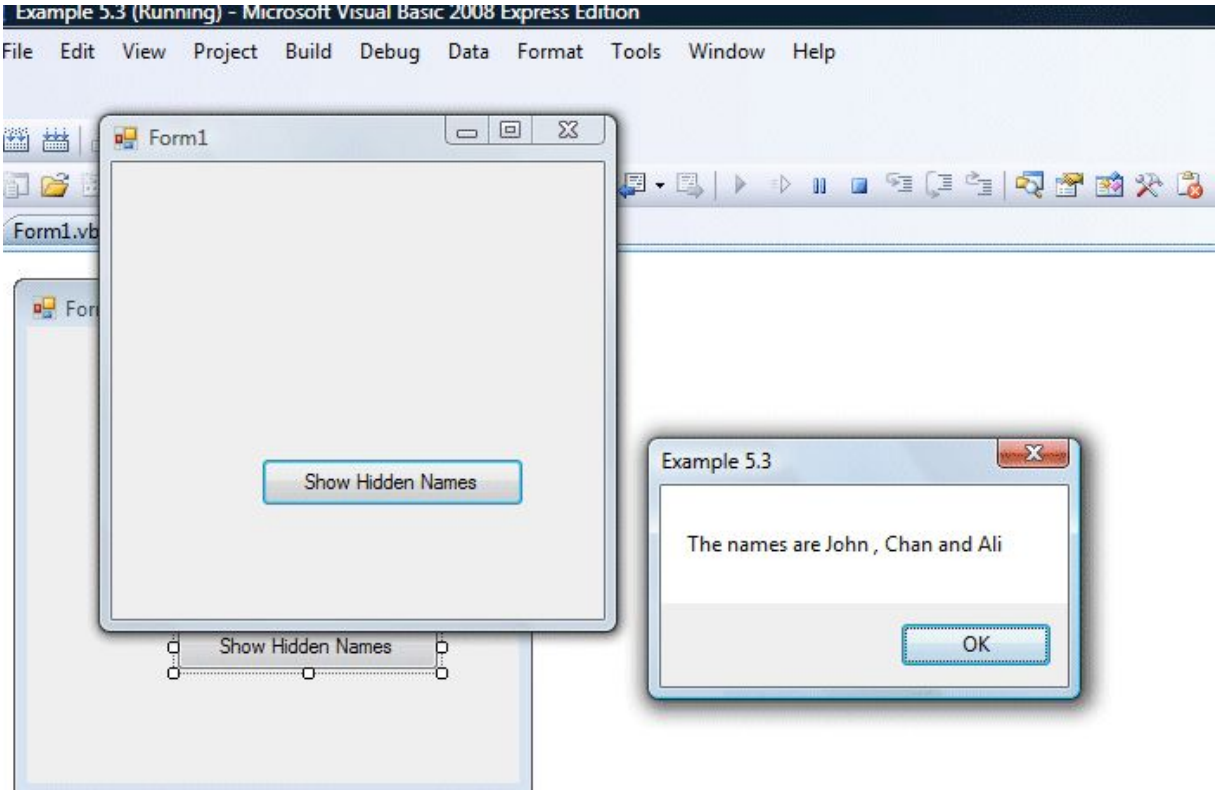

Figure 5-4: The Output Window for Displaying Names