

# Visual Basic 6.0

## Made Easy

**LIEW VOON KIONG**

## **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](http://www.vbtutor.net); especially those who contributed their comments, for their support and encouragement. Finally, I would like to thank my publisher BookSurge for the guidance and assistance in producing this book.

## **About the Author**

Dr. Liew Voon Kiong holds a bachelor degree in mathematics, a master degree in management and a doctoral degree in business administration. He has been involved in programming for more than 15 years. He created the popular online Visual Basic Tutorial at [www.vbtutor.net](http://www.vbtutor.net) in 1996 and since then the web site has attracted millions of visitors and it is one of the top searched Visual Basic Tutorial website in many search engines including Google. In order to provide more support for the Visual Basic hobbyists, he has written this book based on the Visual Basic tutorial.

|   |           |
|---|-----------|
| <b><u>ACKNOWLEDGEMENT</u></b> .....                                       | <b>I</b>  |
| <b><u>ABOUT THE AUTHOR</u></b> .....                                      | <b>I</b>  |
| <b><u>LESSON 1 INTRODUCTION TO VISUAL BASIC</u></b> .....                 | <b>1</b>  |
| 1.1 A Brief Description of Visual Basic .....                             | 1         |
| 1.2 The Visual Basic Environment .....                                    | 2         |
| <b><u>LESSON 2 BUILDING THE VISUAL BASIC APPLICATIONS</u></b> .....       | <b>5</b>  |
| 2.1 Creating Simple Visual Basic Applications .....                       | 5         |
| 2.2 Steps in Building a Visual Basic Application .....                    | 10        |
| <b><u>LESSON 3 WORKING WITH CONTROLS</u></b> .....                        | <b>13</b> |
| 3.1 The Properties of the Controls .....                                  | 13        |
| <b><u>LESSON 4 WRITING THE CODES</u></b> .....                            | <b>22</b> |
| <b><u>LESSON 5 VISUAL BASIC DATA</u></b> .....                            | <b>26</b> |
| 5.1 Types of Visual Basic Data .....                                      | 27        |
| <b><u>LESSON 6 MANAGING VISUAL BASIC DATA</u></b> .....                   | <b>31</b> |
| 6.1 Assigning Values to the Variables .....                               | 31        |
| 6.2 Arithmetic Operators in Visual Basic .....                            | 31        |
| <b><u>LESSON 7 CONTROLLING PROGRAM FLOW – PART I</u></b> .....            | <b>35</b> |
| 7.1 Conditional Operators .....   | 35        |
| 7.2 Logical Operators .....   | 35        |
| 7.3 Using If....Then....Elseif....Else Statements with Operators .....    | 36        |
| <b><u>LESSON 8 CONTROLLING PROGRAM FLOW – PART II</u></b> .....           | <b>41</b> |
| 8.1 The Select Case .... End Select Statement .....                       | 41        |
| <b><u>LESSON 9 LOOPING PART I</u></b> .....                               | <b>46</b> |
| 9.1 Do.... Loop .....   | 46        |
| 9.2: Exiting the Do Loop .....  | 49        |
| 9.3 The While....Wend Loop .....  | 50        |
| <b><u>LESSON 10 LOOPING PART II</u></b> .....                             | <b>52</b> |
| 10.1 The For....Next Loop .....   | 52        |
| 10.2 Exiting the For....Next Loop .....                                   | 54        |
| 10.3 The Nested Loop .....  | 55        |
| <b><u>LESSON 11 VISUAL BASIC FUNCTIONS- PART I</u></b> .....              | <b>58</b> |
| 11.1 Introduction to Visual Basic Functions .....                         | 58        |
| 11.2 The MsgBox ( ) Function .....  | 58        |
| 11.3 The InputBox( ) Function .....                                       | 62        |
| <b><u>LESSON 12 VISUAL BASIC FUNCTIONS- PART II</u></b> .....             | <b>64</b> |
| 12.1 The Rnd Function .....   | 64        |
| 12.2 The Numeric Functions .....  | 67        |
| 12.3 Trigonometric Functions .....  | 69        |
| <b><u>LESSON13 VISUAL BASIC FUNCTIONS PART III- FORMATTING OUTPUT</u></b> |           |
| <b>71</b>   |           |
| 13.1 The Tab Function .....   | 71        |

|   |    |
|---|----|
| <a href="#">13.2 The Space Function</a> .....       | 73 |
| <a href="#">13.3 The Format Function</a> .....      | 74 |
| <a href="#">13.4 Formatting Date and Time</a> ..... | 77 |

## **LESSON14 VISUAL BASIC FUNCTIONS PART IV- MANIPULATING STRINGS**

81

|  |    |
|--|----|
| <a href="#">14.1 The Len Function</a> .....                  | 81 |
| <a href="#">14.2 The Right Function</a> .....                | 81 |
| <a href="#">14.3 The Left Function</a> .....                 | 82 |
| <a href="#">14.4 The Ltrim Function</a> .....                | 82 |
| <a href="#">14.5 The Rtrim Function</a> .....                | 82 |
| <a href="#">14.6 The Trim function</a> .....                 | 82 |
| <a href="#">14.7 The Mid Function</a> .....                  | 82 |
| <a href="#">14.8 The InStr function</a> .....                | 83 |
| <a href="#">14.9 The Ucase and the Lcase functions</a> ..... | 83 |
| <a href="#">14.10 The Str and Val functions</a> .....        | 83 |
| <a href="#">14.11 The Chr and the Asc functions</a> .....    | 83 |

## **LESSON 15 VISUAL BASIC FUNCTIONS PART V- CREATING YOUR OWN FUNCTIONS**

81

### **LESSON 16 CREATING VB FUNCTIONS FOR MS EXCEL**..... 91

|   |    |
|---|----|
| <a href="#">16.1 The Needs to Create User-Defined Functions in MS-Excel</a> ..... | 91 |
| <a href="#">16.2 Using Microsoft Excel Visual Basic Editor</a> .....              | 92 |

### **LESSON 17 ARRAYS PART I**..... 95

|  |    |
|--|----|
| <a href="#">17.1 Introduction to Array</a> ..... | 95 |
| <a href="#">17.2 Declaring Arrays</a> .....      | 95 |
| <a href="#">17.3 The Control Array</a> .....     | 98 |

### **LESSON 18 ARRAYS PART II- TWO DIMENSIONAL ARRAYS**..... 99

### **LESSON 19 GRAPHICS**..... 103

|  |     |
|--|-----|
| <a href="#">19.1 The line and Shape controls</a> .....           | 103 |
| <a href="#">19.2 The Image Box and the Picture Box</a> .....     | 106 |
| <a href="#">19.3 PSet, Line and Circle Drawing Methods</a> ..... | 108 |

### **LESSON 20 CREATING MULTIMEDIA APPLICATIONS**..... 118

|   |     |
|---|-----|
| <a href="#">20.1 Creating a CD Player</a> .....         | 119 |
| <a href="#">20.2 Creating an Audio Player</a> .....     | 121 |
| <a href="#">20.3 Creating a Multimedia Player</a> ..... | 126 |

### **LESSON 21 ANIMATION - PART I**..... 131

### **LESSON 22 ANIMATION - PART II**..... 138

|  |     |
|--|-----|
| <a href="#">22.1 Animation using a drag and drop procedure</a> ..... | 138 |
| <a href="#">22.2 Animation with complete motion</a> .....            | 139 |

### **LESSON 23 ANIMATION - PART III**..... 143

|  |     |
|--|-----|
| <a href="#">23.1 Animation using timers</a> .....          | 143 |
| <a href="#">23.2 Animation using the Move Method</a> ..... | 146 |

### **LESSON 24 FUN AND GAMES PROGRAMMING- PART I**..... 147

|   |     |
|---|-----|
| <a href="#">24.1 Snake Chess</a> .....      | 147 |
| <a href="#">24.2 The Slot Machine</a> ..... | 159 |

### **LESSON 25 FUN AND GAMES PROGRAMMING- PART II**..... 169

|   |            |
|---|------------|
| <a href="#">25.1 Boggle</a> .....   | 169        |
| <a href="#">25.2 Reversi</a> .....  | 171        |
| <a href="#">25.3 Calculator</a> .....   | 196        |
| <b><a href="#">LESSON 27 WORKING WITH FILES</a></b> .....                               | <b>227</b> |
| <a href="#">27.1 Introduction</a> .....   | 227        |
| <a href="#">27.2 Creating a Text File</a> .....   | 227        |
| <a href="#">27.3 Reading a File</a> .....   | 228        |
| <b><a href="#">LESSON 28 CREATING BASIC DATABASE APPLICATIONS IN VB</a></b> .....       | <b>234</b> |
| <b><a href="#">LESSON 29 CREATING DATABASE APPLICATIONS USING ADO CONTROL</a></b> ..... | <b>238</b> |
| <b><a href="#">LESSON 30 INTERNET AND WEB APPLICATIONS</a></b> .....                    | <b>253</b> |
| <a href="#">30.1 The Web Browser</a> .....  | 252        |
| <a href="#">30.2 The FTP program</a> .....  | 257        |

# Lesson 1

## Introduction to Visual Basic

- 
- ❖ A brief description of Visual Basic
  - ❖ Getting to know the Visual Basic environment
- 

### 1.1 A brief description of Visual Basic

VISUAL BASIC is a high level programming language evolved from the earlier DOS version called BASIC. BASIC stands for Beginners' All-purpose Symbolic Instruction Code. The program codes in Visual Basic resemble the English language. Different software companies produce different versions of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, and IBM BASICA and so on.

Visual Basic is a fairly easy programming language to learn and it is for anybody who is interested in programming but lack professional training in software engineering. Learning VB will help young children to improve their logical thinking skills and develop their minds. You can choose to program in VB purely for fun and enjoyment or you can create more advanced applications such as educational courseware and commercial software.

VISUAL BASIC is a VISUAL and events driven Programming Language. These are the main divergences from the old BASIC. In BASIC, programming is done in a text-based environment and the program is executed sequentially. In VISUAL BASIC, programming is done in a graphical environment. In old BASIC, you have to write a text-based procedure to design the interface, but Visual Basic enables you to design the interface by dragging and resizing the objects as well as changing their colors, just like any windows-based programs.

Visual Basic is event-driven because users may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions (events). Examples of events are clicking a command button, entering text into a text box, selecting an item in a list box etc. Therefore, a VISUAL BASIC Program is made up of many subprograms; each with its own program code which can be

executed independently and at the same time can be linked together in one way or another.

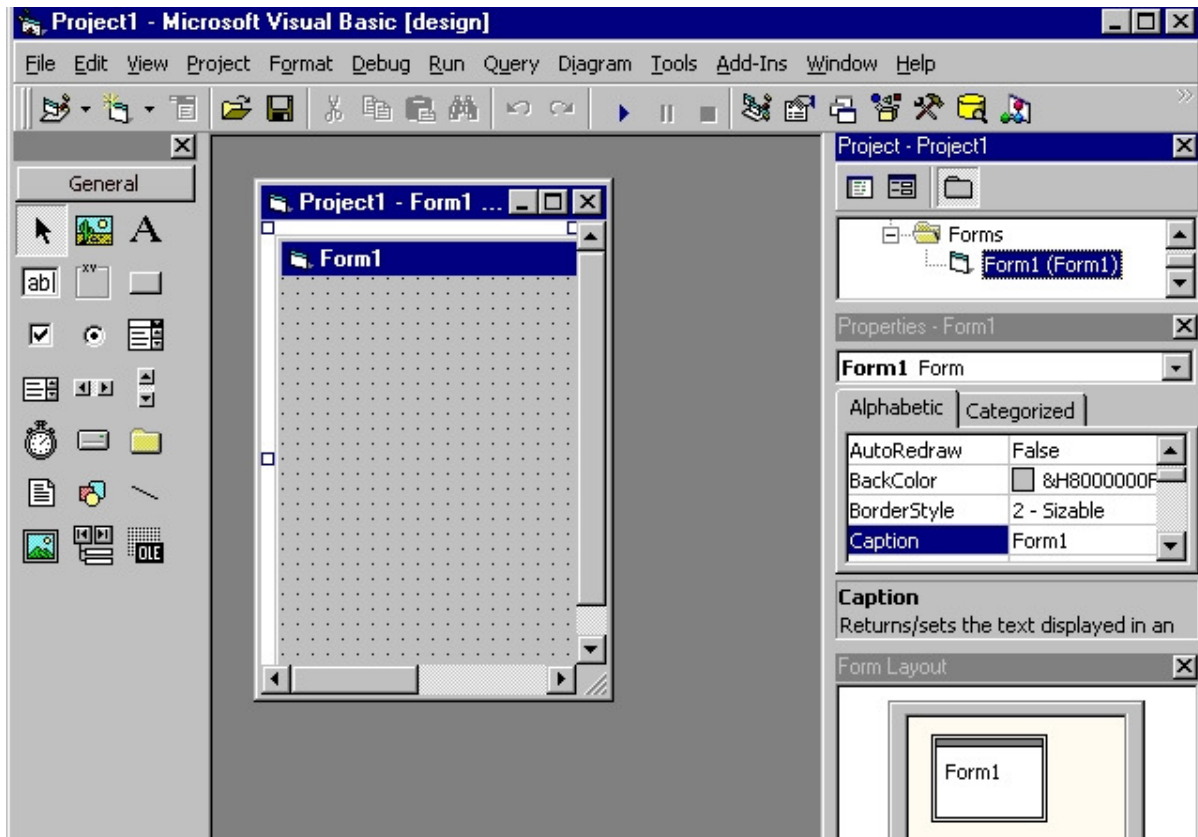
## 1.2 The Visual Basic Environment

Upon start up, Visual Basic 6.0 will display the following dialog box as shown in Figure 1.1. You can choose to start a new project, open an existing project or select a list of recently opened programs. A project is a collection of files that make up your application. There are various types of applications that can be created; however, we shall concentrate on creating Standard EXE programs (EXE means executable program). Now, click on the Standard EXE icon to go into the VB programming environment.

**Figure 1.1 The Visual Basic Start-up Dialog Box**



**Figure 1.2: The Visual Basic Environment**



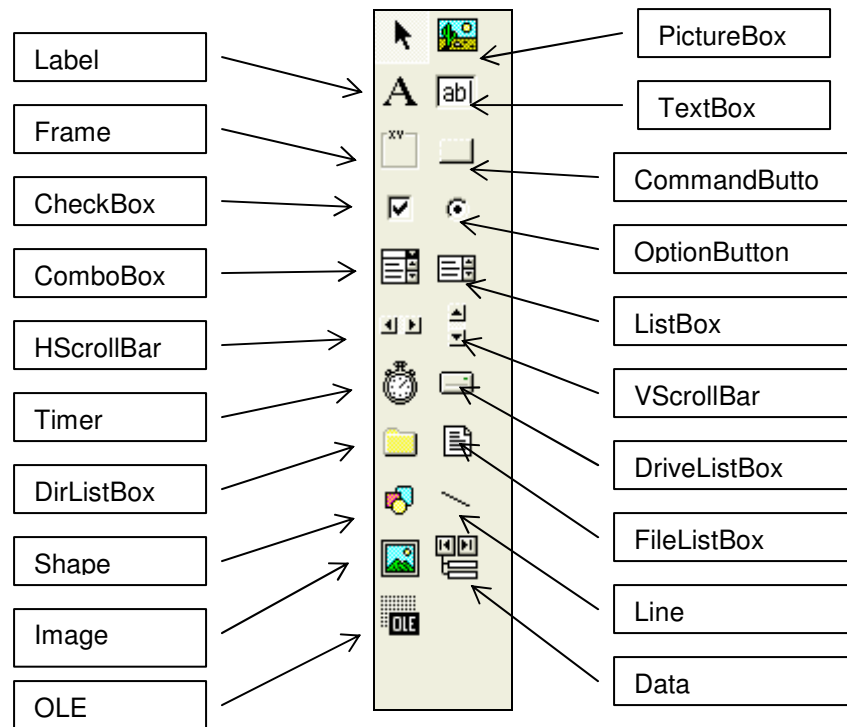
In Figure 1.2, the Visual Basic Environment consists of

- A blank form for you to design your application's interface.
- The project window which displays the files that are created in your application.
- The properties window which displays the properties of various controls and objects that are created in your application.

It also has a Toolbox that consists of all the controls essential for developing a VB Application. The controls include text boxes, command buttons, labels, combo boxes, picture boxes, image boxes, timers and other objects that can be dragged to and drawn on the form to perform certain tasks according to the events assigned to them. You may also add additional objects. First, click on the project item on the menu, then on the components item on the drop-down list, and lastly select the controls you want to use in your program. The controls for Standard.EXE window are shown in Figure 1.3.



**Figure 1.3: The controls available for a Standard.exe project**



### Exercise 1

1. Write down the meaning of BASIC.
2. Briefly explain the difference between VISUAL BASIC and the old BASIC.
3. List out all the controls in a Standard.exe Visual Basic project.

## Lesson 2

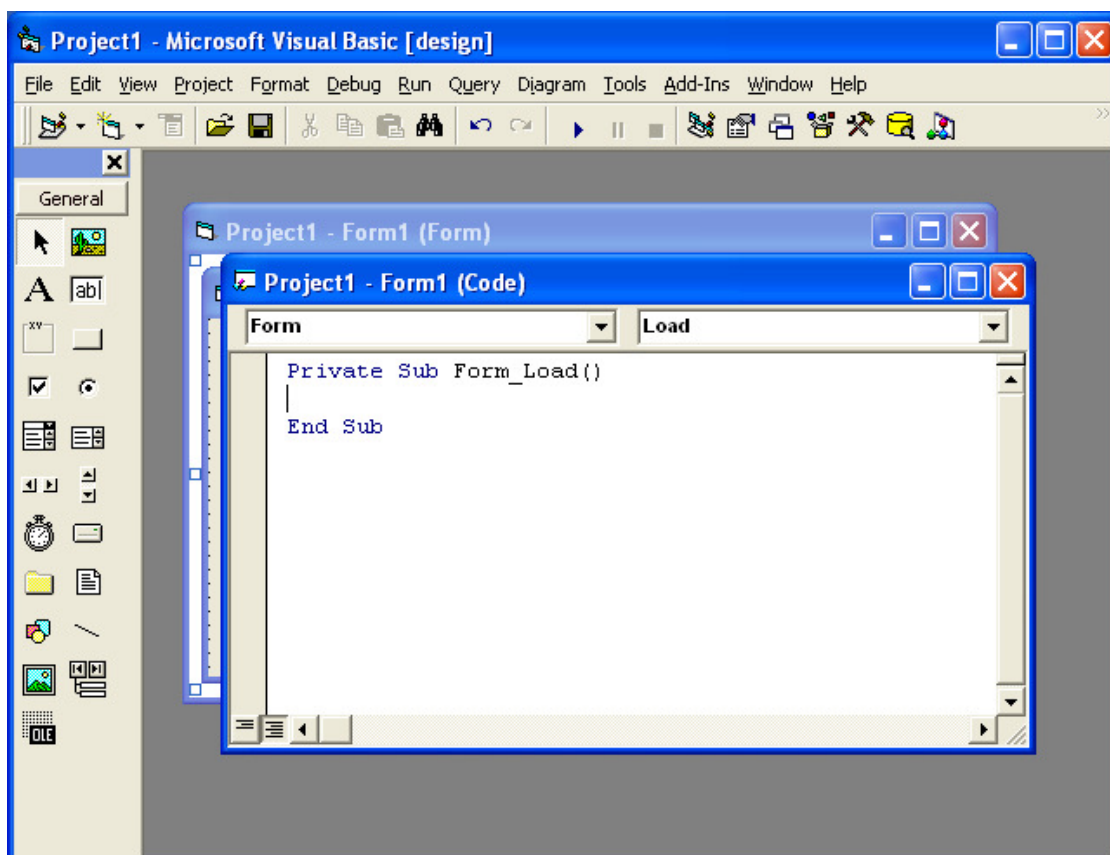
# Building the Visual Basic Applications

- 
- ❖ Creating simple Visual Basic Applications.
  - ❖ Getting to know the steps in building a Visual Basic Application.
- 

### 2.1 Creating Simple Visual Basic Applications

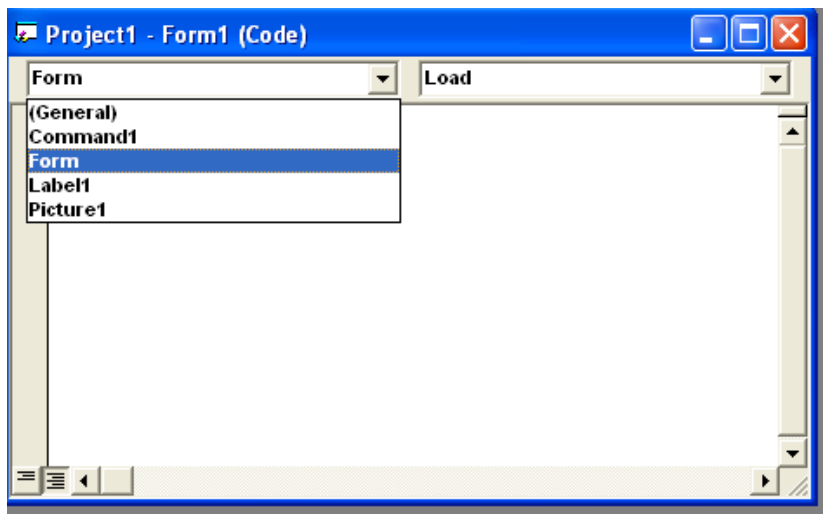
In this section, we are not going into the technical aspects of VB programming; the purpose of this section is to let you get a feel for it. Now, you can try out a few examples in this lesson. Example 2.1 is a simple program. First of all, you have to launch Microsoft Visual Basic. Normally, a default form Form1 will be available for you to start your new project. Double click on Form1, and the source code window for it as shown in Figure 2.1 will appear. The top of the source code window consists of a list of objects (on the left) and their associated events or procedures (on the right). In Figure 2.1, the object displayed is Form and the associated procedure is Load.

**Figure 2.1 The Source Code Window**

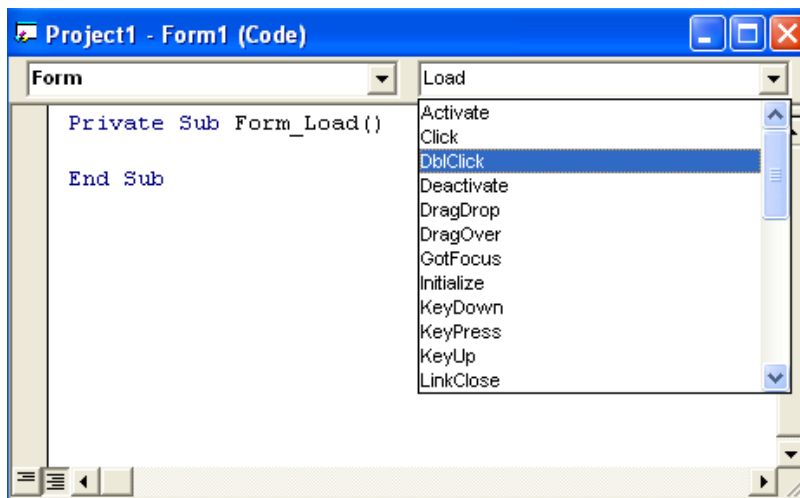


When you click on the object box, the drop-down list will display a list of objects you have inserted into your form as shown in Figure 2.2. Here, you can see the form, the command button with the name Command1, the Label with the name Label1 and the PictureBox with the name Picture1. Similarly, when you click on the procedure box, a list of procedures associated with the object will be displayed as shown in Figure 2.3. Some of the procedures associated with the object Form are Activate, Click, DbClick (Double-Click) , DragDrop, keyPress etc. Each object has its own set of procedures. You can select an object and write codes for any of its procedures in order to perform certain tasks.

**Figure 2.2: List of Objects**



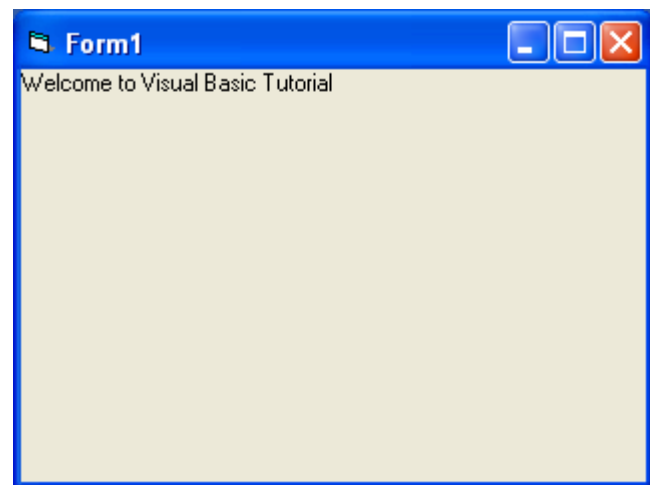
**Figure 2.3: List of Procedures**



You do not have to worry about the beginning and the end statements (i.e. Private Sub Form\_Load.....End Sub.); Just key in the codes between the above two statements exactly as shown here. When you run the program, don't be surprised that nothing shows up. In order to display the output of the program, you have to add the Form1.show statement like in Example 2.1 or you can just use the Form\_Activate ( ) event procedure as shown in example 2.2. The command Print does not mean printing using a printer. Instead, it means displaying the output on the computer screen. Now, press F5 or click on the run button to run the program and you will get the output as shown in Figure 2.4.

### Example 2.1

```
Private Sub Form_Load ( )  
    Form1.show  
    Print "Welcome to Visual Basic tutorial"  
End Sub
```



**Figure 2.4 : The output of example 2.1.**

You can also perform simple arithmetic calculations as shown in example 2.2. VB uses \* to denote the multiplication operator and / to denote the division operator. The output is shown in Figure 2.5, where the results are arranged vertically.

### Example 2.2

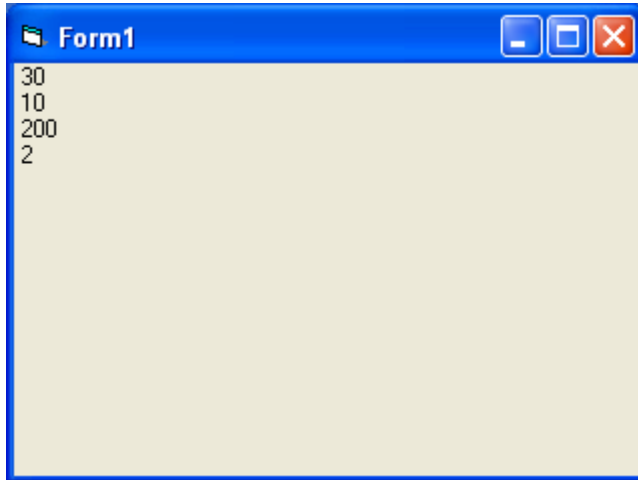
```
Private Sub Form_Activate ( )  
    Print 20 + 10  
    Print 20 - 10
```

```
Print 20 * 10
```

```
Print 20 / 10
```

```
End Sub
```

**Figure 2.5: The output of example 2.2**



Example 2.2 can also be written as

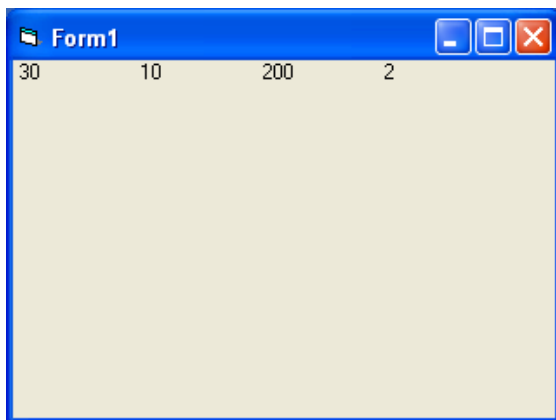
```
Private Sub Form_Activate ( )
```

```
    Print 20 + 10, 20 - 10, 20 * 10, 20 / 10
```

```
End Sub
```

The numbers will be arranged in a horizontal line separated by spaces as shown in Figure 2.6

**Figure 2.6: Output in a horizontal line**



Example 2.3 is an improved version of example 2.2 as it employs two variables  $x$  and  $y$  and assigns initial values of 20 and 10 to them respectively. When you need to change the values of  $x$  and  $y$ , just change the initial values rather than changing every individual value which is more time consuming.

**Example 2.3**

```

Private Sub Form_Activate ( )
    x = 20
    y = 10
    Print x + y
    Print x - y
    Print x * y
    Print x / y
End Sub

```

Besides, you can also use the **+** or the **&** operator to join two or more texts (string) together like in example 2. 4 (a) and (b)

**Example 2.4(a)**

```

Private Sub Form_Activate ( )
    A = "Tom "
    "likes "
    "to "
    "eat "
    "burgers."
    A + B + C + D + E
End Sub

```

**Example 2. 4(b)**

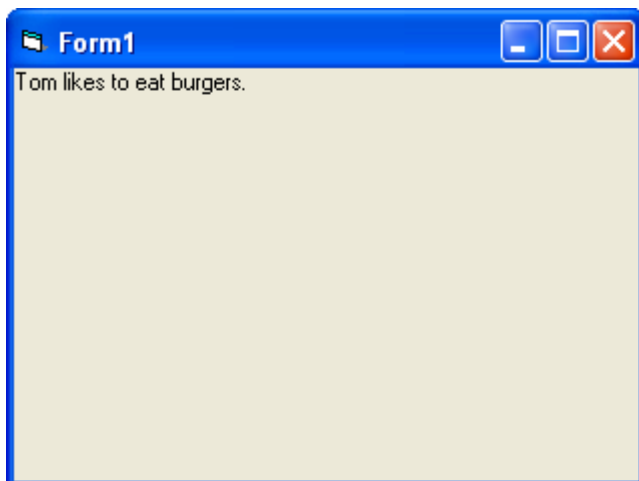
```

Private Sub Form_Activate ( )
    A = "Tom "           B =
    B = "likes "        C =
    C = "to "           D =
    D = "eat "          E =
    E = "burgers."     Print
    Print A & B & C & D & E
End Sub

```

The output is as shown in Figure 2.7

**Figure 2.7: The Output of Example 2.4(a) &(b)**



## 2.2 Steps in Building a Visual Basic Application

Generally, there are three basic steps in building a VB application. The steps are as follows:

Step 1: Design the interface

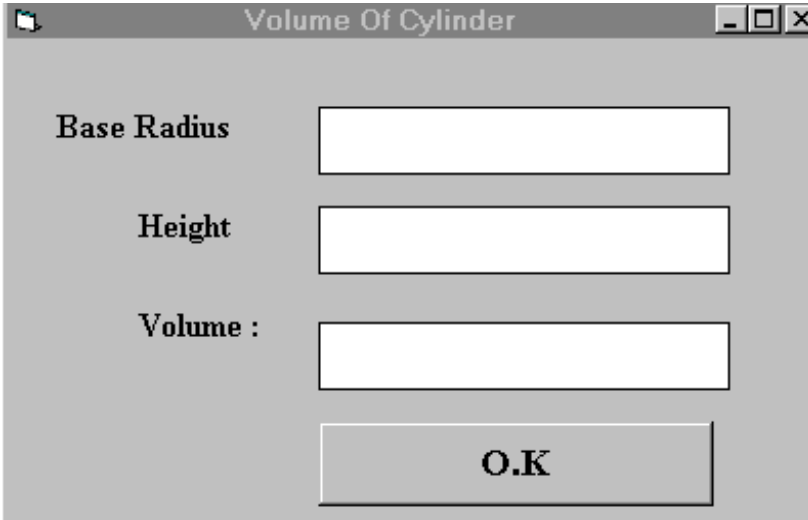
Step 2: Set the properties of the controls (Objects)

Step 3: Write the events' procedures

### Example 2.5

This program is a simple program that calculates the volume of a cylinder.

**Figure 2.8 A Program to Calculate the Volume of a Cylinder**



The image shows a screenshot of a Visual Basic application window titled "Volume Of Cylinder". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main area of the window is light gray and contains three labels: "Base Radius", "Height", and "Volume :". Each label is positioned to the left of an empty white text box. Below these three text boxes is a single button with the text "O.K" centered on it.

First of all, go to the properties window and change the form caption to Volume of Cylinder, then drag and insert three labels into the form and change their captions to Base Radius, Height and Volume respectively. After that, insert three Text Boxes and clear its text contents so that you get three empty boxes. Name the text boxes as radius, hght (we cannot use height as it is one of the built-in control names) and volume respectively. Lastly, insert a command button and change its caption to O.K and its name to OK. Now save the project as cylinder.vbp and the form as cylinder.frm. We shall leave out the codes at the moment which you shall learn in the following lessons.

### Example 2.6

Designing an attractive and user friendly interface is the first step in constructing a VB program. To illustrate this point, let's look at the calculator program.

**Fig 2.9 A Calculator**



Now, follows the steps below to design the calculator interface.

- Resize the form until you get the size you are satisfied with.
- Go to the properties window and change the default caption of the form to the caption you like, such as 32 Calculator.
- Change the other properties of the form, such as background color, foreground color, and border style. For this particular program, I recommend you set the following properties for Form1:

- |             |              |
|-------------|--------------|
| BorderStyle | Fixed Single |
| MaxButton   | False        |
| minButton   | True         |

These properties will ensure that the users cannot resize or maximize your calculator window, but able to minimize the window.



- Draw the Display Panel by clicking on the Label button and place your mouse on the form. Start drawing by pressing down your mouse button and dragging it.
- Click on the panel and the corresponding properties window will appear. Clear the default label so that the caption is blank. It is better to set the background color to a brighter color while the foreground color should be of a darker color like black (for easy viewing). Change the name to display as I am going to use it later to write codes for the calculator.
- Now draw the command buttons that are necessary to operate a calculator. I suggest you follow exactly what is shown in the image above.

Now run the project by pressing F5. If you are satisfied with the appearance, go ahead and save the project.

## Exercise 2

1. Write down the list of procedures which are associated with the Form object.
2. Write a program to display the sentence "I like Visual Basic".
3. Write a program to compute the value of  $1500+1000-450*10+300\div 5$
4. Using a value of 9 to X and a value of 13 to y, write a program to compute the values of  $2x+y$ ,  $(x+y)\div 2$  and  $x^2y$

## Lesson 3

# Working with Controls

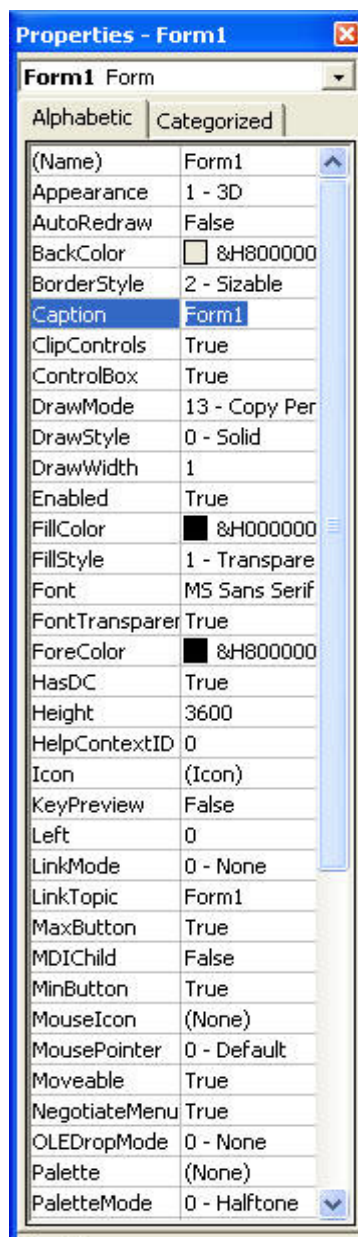
- 
- ❖ Setting the properties of the controls
  - ❖ Learning how to work with the controls
- 

### 3.1 The Properties of the Controls

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 determine its appearance and how it will work with the event procedure. You can set the properties of the controls in the properties window or at runtime.

Figure 3.1 on the right is a typical properties window for a form. You can rename the form caption to any name that you like best. In the properties window, the item which 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 and then changing them by typing or selecting the options available. For example, in order to change the caption, just highlight Form1 under the name Caption and change it to other names. You may also alter the appearance of the form by setting it to 3D or flat. You can do other things like changing the foreground and background color, changing the font type and font size, enabling or disabling the minimize and maximize buttons and etc. You can also change the properties at runtime to give special effects such as changing of colors, shape, animation effects and so on. For example the following code will change the form color to red every time the form is loaded. VB uses hexadecimal system

**Figure 3.1**



to represent colors. You can check the color codes in the properties windows which are shown under ForeColor and BackColor .

```
Private Sub Form_Load()  
    Form1.Show  
    Form1.BackColor = &H000000FF&  
End Sub
```

Another example is to change the control Shape to a particular shape at runtime by writing the following code. This code will change the shape to a circle at runtime. Later you will learn how to change the shapes randomly by using the RND function.

```
Private Sub Form_Load()  
    Shape1.Shape = 3  
End Sub
```

I would like to stress that Learning how and when to set the objects' properties is very important as it can help you to write a good program. So, I advice you to spend a lot of time playing with the objects' properties.

I am not going into the details on how to set the properties. However, I would like to stress a few important points about setting up the properties.

- You should set the Caption Property of a control clearly so that a user knows what to do with that command. For example, in the calculator program, as all the captions of the command buttons such as +, - , MC, MR are commonly found in an ordinary calculator, a user should have no problem in manipulating the buttons.
- A lot of programmers like to use a meaningful name for the Name Property because it is easier for them to write and read the event procedure and easier to debug or modify the programs later. However, it is not a must to do that as long as you label your objects clearly and use comments in the program whenever you feel necessary.
- One more important property is whether the control is being enabled or not.
- Finally, you must also consider making the control visible or invisible at runtime, or when should it become visible or invisible.

### 3.2 Handling some of the common controls

#### a) The Text Box

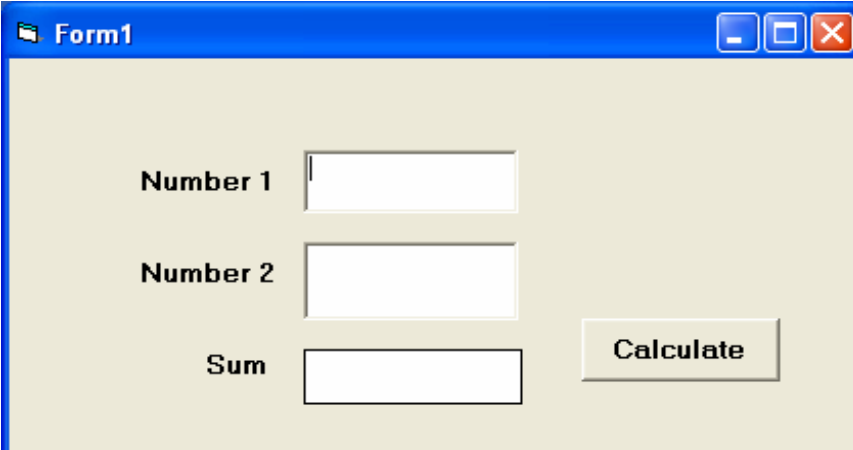
The text box is the standard control that is used to receive input from the user as well as to display the output. It can handle string (text) and numeric data but not images or pictures. Strings in a text box can be converted to numeric data by using the function `Val(text)`. The following example illustrates a simple program that processes the inputs from the user.

#### Example 3.1

In this program, two text boxes are inserted into the form together with a few labels. The two text boxes are used to accept inputs from the user and one of the labels will be used to display the sum of two numbers that are entered into the two text boxes. A command button is also programmed to calculate the sum of the two numbers using the plus operator. The program creates a variable, 'sum', to accept the summation of values from Textbox 1 and text box 2. The procedure to calculate and display the output on the label is shown below. The output is shown in Figure 3.1.

```
Private Sub Command1_Click()  
    'To add the values in Textbox 1 and text box 2  
    Sum = Val(Text1.Text) + Val(Text2.Text)  
    'To display the answer on label 1  
    Label1.Caption = Sum  
End Sub
```

Figure 3.1



The screenshot shows a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main area of the form is light beige and contains three text boxes and one button. The first text box is labeled "Number 1" and is empty. The second text box is labeled "Number 2" and is also empty. The third text box is labeled "Sum" and is empty. To the right of the "Sum" text box is a button labeled "Calculate".

### b) The Label

The label is a very useful control for Visual Basic, as it is not only used to provide instructions and guides to the users, it can also be used to display outputs. One of its most important properties is **Caption**. Using the syntax **label.Caption**, it can display text and numeric data. You can change its caption in the properties window and also at runtime. Please refer to Example 3.1 and Figure 3.1 for the usage of labels.

### c) The Command Button

The command button is a very important control as it is used to execute commands. It displays an illusion that the button is pressed when the user clicks on it. The most common event associated with the command button is the Click event, and the syntax for the procedure is

```
Private Sub Command1_Click ()
    Statements
End Sub
```

### d) The Picture Box

The Picture Box is one of the controls that used to handle graphics. You can load a picture during the designing phase by clicking on the picture item in the properties window and select the picture from the selected folder. You can also load the picture at runtime using the **LoadPicture** method. For example, this statement will load the picture grape.gif into the picture box.

```
Picture1.Picture=LoadPicture ("C:\VB program\Images\grape.gif")
```

You will learn more about the picture box in future lessons. The image in the picture box is not resizable.

### e) The Image Box

The Image Box is another control that handles images and pictures. It functions almost identically to the picture box. However, there is one major difference, the image in an Image Box is stretchable, which means it can be resized. This feature is not available in the Picture Box. Similar to the Picture Box, the LoadPicture method can also be used. For example, this statement loads the picture grape.gif into the image box.

```
Image1.Picture=LoadPicture ("C:\VB program\Images\grape.gif")
```

### f) The List Box

The function of the List Box is to present a list of items where the user can click and select items from the list. In order to add items to the list, we can use the **AddItem method**. For example, if you wish to add a number of items to List box 1, you can key in the following statements

#### Example 3.2

```
Private Sub Form_Load ( )
    List1.AddItem "Lesson1"
    List1.AddItem "Lesson2"
    List1.AddItem "Lesson3"
    List1.AddItem "Lesson4"
End Sub
```

The items in the list box can be identified by the **ListIndex** property, the value of the ListIndex for the first item is 0, the second item has a ListIndex 1, and the second item has a ListIndex 2 and so on

### g) The Combo Box

The function of the Combo Box is also to present a list of items where the user can click and select the items from the list. However, the user needs to click on the small arrowhead on the right of the combo box to see the items which are presented in a drop-down list. In order to add items to the list, you can also use the **AddItem method**. For example, if you wish to add a number of items to Combo box 1, you can key in the following statements

#### Example 3.3

```
Private Sub Form_Load ( )
    Combo1.AddItem "Item1"
    Combo1.AddItem "Item2"
    Combo1.AddItem "Item3"
    Combo1.AddItem "Item4"
End Sub
```

### h) The Check Box

The Check Box control lets the user select or unselect an option. When the Check Box is checked, its value is set to 1 and when it is unchecked, the value is set to 0. You can include the statements `Check1.Value=1` to mark the Check Box and `Check1.Value=0` to unmark the Check Box, and use them to initiate certain actions. For example, the program will change the background color of the form to red when the check box is unchecked and it will change to blue when the check box is checked. You will learn about the conditional statement `If...Then....Elseif` in later lessons. `VbRed` and `vbBlue` are color constants and `BackColor` is the background color property of the form.

#### Example 3.4

```
Private Sub Check1_Click ()
    If Check1.Value = 0 Then
        Form1.BackColor = vbRed
    Elseif Check1.Value = 1 Then
        Form1.BackColor = vbBlue
    End If
End Sub
```

### i) The Option Box

The Option Box control also lets the user selects one of the choices. However, two or more Option Boxes must work together because as one of the Option Boxes is selected, the other Option Boxes will be unselected. In fact, only one Option Box can be selected at one time. When an option box is selected, its value is set to "True" and when it is unselected; its value is set to "False". In the following example, the shape control is placed in the form together with six Option Boxes. When the user clicks on different option boxes, different shapes will appear. The values of the shape control are 0, 1, and 2,3,4,5 which will make it appear as a rectangle, a square, an oval shape, a rounded rectangle and a rounded square respectively.

**Example 3.5**

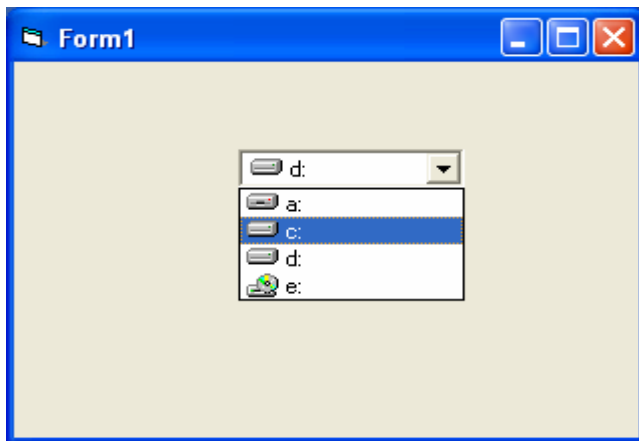
```
Private Sub Option1_Click ( )  
    Shape1.Shape = 0  
End Sub  
Private Sub Option2_Click()  
    Shape1.Shape = 1  
End Sub  
Private Sub Option3_Click()  
    Shape1.Shape = 2  
End Sub  
Private Sub Option4_Click()  
    Shape1.Shape = 3  
End Sub  
Private Sub Option5_Click()  
    Shape1.Shape = 4  
End Sub  
Private Sub Option6_Click()  
    Shape1.Shape = 5  
End Sub
```

**j) The Drive List Box **

The Drive ListBox is used to display a list of drives available in your computer. When you place this control into the form and run the program, you will be able to select different drives from your computer as shown in Figure 3.2

**Figure 3.2 The Drive List Box**

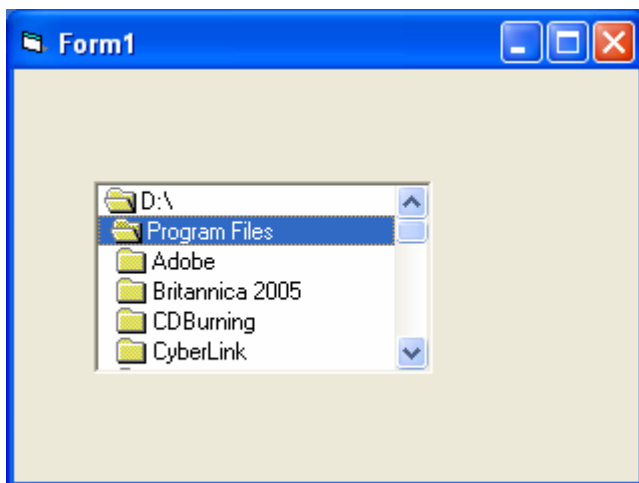




### k) The Directory List Box

The Directory List Box is used to display the list of directories or folders in a selected drive. When you place this control into the form and run the program, you will be able to select different directories from a selected drive in your computer as shown in Figure 3.3

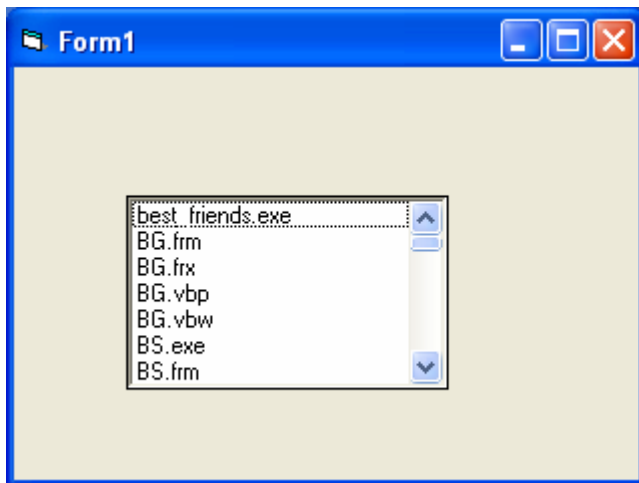
**Figure 3.3 The Directory List Box**



### l) The File List Box

The File List Box is used to display the list of files in a selected directory or folder. When you place this control into the form and run the program, you will be able to a list of files in a selected directory as shown in Figure 3.4

**Figure 3.4**



You can coordinate the Drive List Box, the Directory List Box and the File List Box to search for the files you want. The procedure will be discussed in later lessons.

### Exercise 3

1. Start a Visual Basic project and set the following properties of the form
  - a. Appearance=Flat
  - b. Caption=My First Program
  - c. Font=Times New Roman, Font Size=10
  - d. Forecolor=White
  - e. Backcolor=Blue
2. Insert six shape controls into the form and set their shape properties to rectangle, square, oval, circle, rounded rectangle, rounded square.
3. Insert two text boxes, one label and one command button, then write the program to compute the product of two numbers that are entered into the text boxes and display the result in the label.
4. Insert a picture control in the form and load a picture from your computer using the LoadPicture Method.
5. Insert a List Box control into the form and add the 5 items into the List Box using the AddItem Method.