

Chapter 10

Introduction to Functions

❖ Getting to know all the functions in Visual Basic 2010

A function is similar to a normal procedure but the main purpose of the function is to accept a certain input and return a value, which is passed on to the main program to finish the execution. There are two types of functions, the built-in functions (or internal functions) and the functions created by the programmers.

The general format of a function is

FunctionName (arguments)

The arguments are values that are passed on to the function.

In this Chapter, we are going to learn two very basic but useful internal functions of Visual Basic, i.e. the `MsgBox()` and `InputBox()` functions.

10.1 MsgBox () Function

The objective of `MsgBox` is to produce a pop-up message box and prompt the user to click on a command button before he /she can continue. This format is as follows:

yourMsg=MsgBox(Prompt, Style Value, Title)

The first argument, `Prompt`, displays the message in the message box. The `Style Value` determines the type of command buttons appear on the message box, as shown in Table 10.1. The `Title` argument will display the title of the message board.

Style Value	Named Constant	Buttons Displayed
0	<code>vbOkOnly</code>	Ok button
1	<code>vbOkCancel</code>	Ok and Cancel buttons
2	<code>vbAbortRetryIgnore</code>	Abort, Retry and Ignore buttons.
3	<code>vbYesNoCancel</code>	Yes, No and Cancel buttons
4	<code>vbYesNo</code>	Yes and No buttons
5	<code>vbRetryCancel</code>	Retry and Cancel buttons

Table 10.1: Style Values

We can use named constant in place of integers for the second argument to make the programs more readable. In fact, VB6 will automatically shows up a list of names constant where you can select one of them.

For example,

```
yourMsg=MsgBox( "Click OK to Proceed", 1, "Startup Menu")
```

and

```
yourMsg=Msg("Click OK to Proceed". vbOkCancel,"Startup Menu")
```

are the same.

yourMsg is a variable that holds values that are returned by the MsgBox () function.

The type of buttons being clicked by the users determines the values. It has to be declared as Integer data type in the procedure or in the general declaration section.

Table 10.2 shows the values, the corresponding named constant and buttons.

Value	Named Constant	Button Clicked
1	vbOk	Ok button
2	vbCancel	Cancel button
3	vbAbort	Abort button
4	vbRetry	Retry button
5	vbIgnore	Ignore button
6	vbYes	Yes button
7	vbNo	No button

Table 10.2: Return Values and Command Buttons

A function is similar to a normal procedure but the main purpose of the function is to accept a certain input and return a value, which is passed on to the main program to finish the execution. There are two types of functions, the built-in functions (or internal functions) and the functions created by the programmers.

The general format of a function is

FunctionName (arguments)

The arguments are values that are passed on to the function.

Example 10.1

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim testmsg As Integer
testmsg = MsgBox("Click to test", 1, "Test message")
If testmsg = 1 Then
MessageBox.Show("You have clicked the OK button")
Else
MessageBox.Show("You have clicked the Cancel button")
End If
End Sub

```

To make the message box look more sophisticated, you can add an icon besides the message. There are four types of icons available in VB2010 as shown in Table 10.3





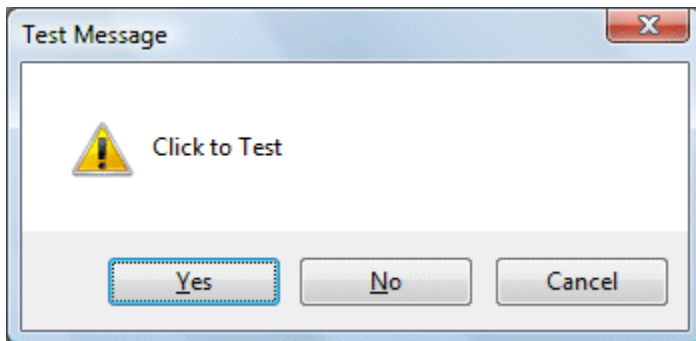
Value	Named Constant	Icon
16	vbCritical	
32	vbQuestion	
48	vbExclamation	
64	vbInformation	

Table 10.3: Named Constants and Icons

Example 10.2

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim testMsg As Integer
testMsg = MsgBox("Click to Test", vbYesNoCancel + vbExclamation, "Test
Message")

If testMsg = 6 Then
    MessageBox.Show("You have clicked the yes button")
ElseIf testMsg = 7 Then
    MessageBox.Show("You have clicked the NO button")
Else
    MessageBox.Show("You have clicked the Cancel button")
End If
End Sub
```

**Figure 10.1**

10.2 The InputBox() Function

An InputBox() function allows the user to enter a value or a message in a text box.

```
userMsg =Microsoft.VisualBasic.InputBox(Prompt, Title, default_text, x-position, y-position)
```

userMsg is a variant data type but typically it is declared as string, which accepts the message input by the user. The arguments are explained as follows:

- Prompt - The message displayed normally as a question asked.
- Title - The title of the Input Box.
- default-text - The default text that appears in the input field where the user may change the message according to his or her wish..
- x-position and y-position - the position or the coordinates of the input box.

Example 10.3

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim userMsg As String
userMsg = Microsoft.VisualBasic.InputBox("What is your message?", "Message
Entry Form", "Enter your messge here", 500, 700)
If userMsg <> "" Then
MessageBox.Show(userMsg)
Else
MessageBox.Show("No Message")
End If
End Sub
```

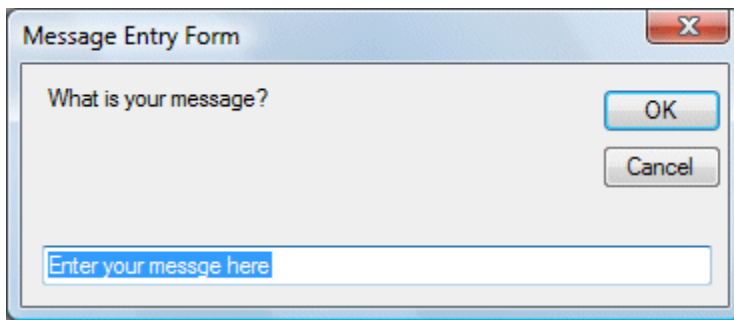


Figure 10.2

Summary

You learned that a function is a procedure that accepts an input and returns a value, which is passed on to the main program to finish the execution. You also learned that the format of a function is `FunctionName (arguments)`.

- In section 10.1, you learned to use the `Msg ()` function to produce a popup message box to prompt the user to click on a button to continue the execution.
- In section 10.2, you learned how to use the `InputBox` to get the input from the user.