



Unit 5

Function

Introduction

MsgBox is used to create a pop-up message box and prompt the user to click on a command button before continues. InputBox() function display a message box where the user can enter a value or a message in the form of text. In this unit, we are going to learn two useful internal functions of Visual basic, i.e. the **MsgBox()** and **InputBox ()** functions.

Lesson 5.1 – 5.2

MsgBox() and InputBox () functions.

Upon completion of this unit you will be able to:

- Use dialogs to display messages.



Outcomes

MsgBox () Function

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

Msg=MsgBox(Prompt, Style Value, Title)

The first argument, Prompt, will display the message in the message box. The Style Value will determine what type of command buttons appear on the message box (Table 1) for types of command button displayed. The Title argument will display the title of the message board.

Table 1: Style Values

Style Value	Named Constant	Buttons Displayed
0	vbOkOnly	Ok button
1	vbOkCancel	Ok and Cancel buttons



2	vbAbortRetryIgnore	Abort, Retry and Ignore buttons.
3	vbYesNoCancel	Yes, No and Cancel buttons
4	vbYesNo	Yes and No buttons
5	vbRetryCancel	Retry and Cancel buttons

We can use named constants in place of integers for the second argument to make the programs more readable.

Example:

```
testmsg = MsgBox("Click OK to Proceed", 1, "Startup")  
and testmsg = MsgBox("Click OK to Proceed",  
vbOKCancel, "Startup") are the same.
```

testmsg is a variable that holds values that are returned by the MsgBox () function. The values are determined by the type of buttons being clicked by the users. Table .2 shows the values, the corresponding named constant and buttons.

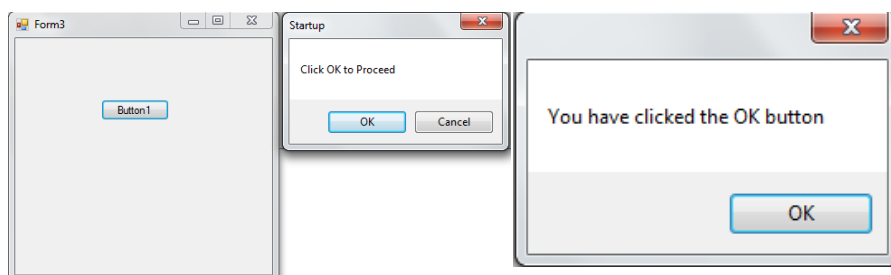
Table 2: Return Values and Command Buttons

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

Example

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

```
    Dim testmsg As Integer  
    testmsg = MsgBox("Click OK to Proceed", 1, "Startup")  
    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 looks more stylish, you can add an icon besides the message. There are four types of icons available in VB2008 as shown in Table 3



Tip

Text displayed in a dialog should be descriptive and as short as possible.

Table 3: Icons

Value	Named Constant	Icon	Description
16	vbCritical		Typically used to alert the user to errors or critical situations
32	vbQuestion		Typically used to ask the user a question.
48	vbExclamation		Typically used to caution the user against potential problems.
64	vbInformation		Typically used to display information about the state of the application.

The InputBox() Function

An InputBox() function will display a message box where the user can enter a value or a message in the form of text. You can use the following format:

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

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 users can use it as his intended input or he may change to the message he wish to enter. x-position and y-position - the position or the coordinates of the input box.

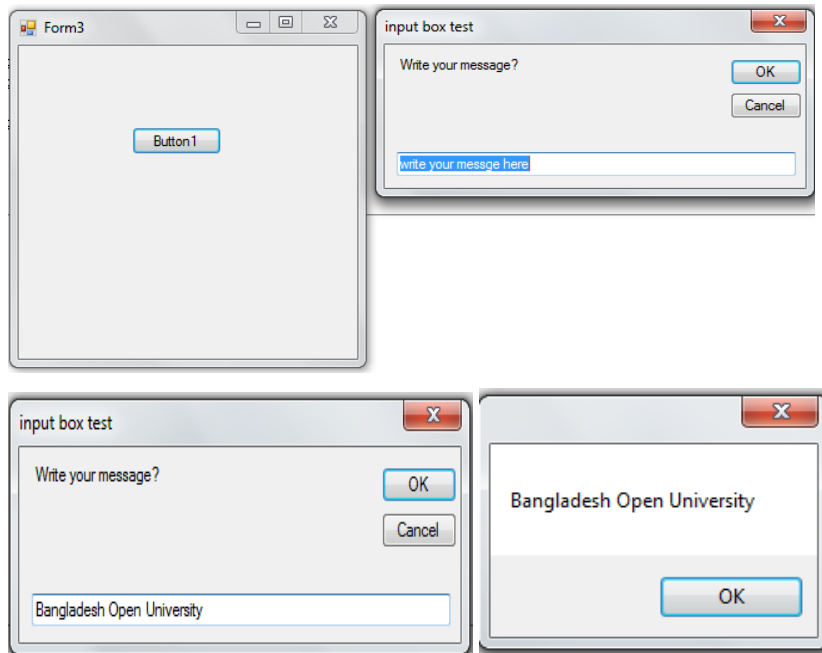


Example

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim testmsg As String
    testmsg = Microsoft.VisualBasic.InputBox("Write your message?", "input box test", "write your message here", 500, 700)

    If testmsg <> "" Then
        MessageBox.Show(testmsg)
    Else
        MessageBox.Show("No Message")
    End If
End Sub
```

The inputbox will appear as shown in the figure below when you press the command button





Assessment



Assessment

MCQ

1. User can enter a value or a message in the form of text in
 - a) An InputBox () function
 - b) MsgBox () Function
 - c) Both
 - d) None
2. Which constant, when passed to method MessageBox.Show, indicates that a question is being asked?
 - a) MessageBox.Question
 - b) MessageBoxIcon.QuestionMark
 - c) MessageBox.QuestionMark
 - d) MessageBoxIcon.Question
3. The first argument passed to method MessageBox. Show is ____
 - a) the text displayed in the dialog's title bar
 - b) a constant representing the Buttons displayed in the dialog
 - c) the text displayed inside the dialog
 - d) a constant representing the icon that appears in the dialog
4. You can specify the Button(s) and icon to be displayed in a message dialog by using the MessageBoxButtons and ____ constants.
 - a) MessageBoxIcon
 - b) MessageBoxIconImages
 - c) MessageBoxPicture
 - d) MessageBoxIcon

Short Questions

1. What are Message Boxes? Discuss about the 'Msgbox' command with example.
2. Which of the followings are same?
 - i. testmsg = MsgBox("Click yes to Proceed", 4, "Startup")
 - ii. testmsg = MsgBox("Click yes to Proceed", vbOkOnly, "Startup")
 - iii. testmsg = MsgBox("Click yes to Proceed", vbYesNo, "Startup")



Lesson 5.3 – 5.4

String Manipulation

Upon completion of this unit you will be able to:

- Learn how to manipulate Strings.



Outcomes

String Manipulation Using + and & signs

String manipulation is an important part of programming because it helps to process data that come in the form of non-numeric types such as name, address, city, book title and etc. Strings can be manipulated using the & sign and the + sign, both perform the string concatenation which means combining two or more smaller strings into larger strings.

For example

"ABC" & "1234" 'Displays "ABC1234"

we can join "Bangladesh" and "Open University" into " Bangladesh Open University " using " Bangladesh "&" Open University " or " Bangladesh "+" Open University ", as shown in the example below

Example 8.1

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
    Dim text1, text2, text3 As String
    text1 = "Bangladesh"
    text2 = " Open University"
    text3 = text1 + text2
    TextBox3.Text = text3
End Sub
```

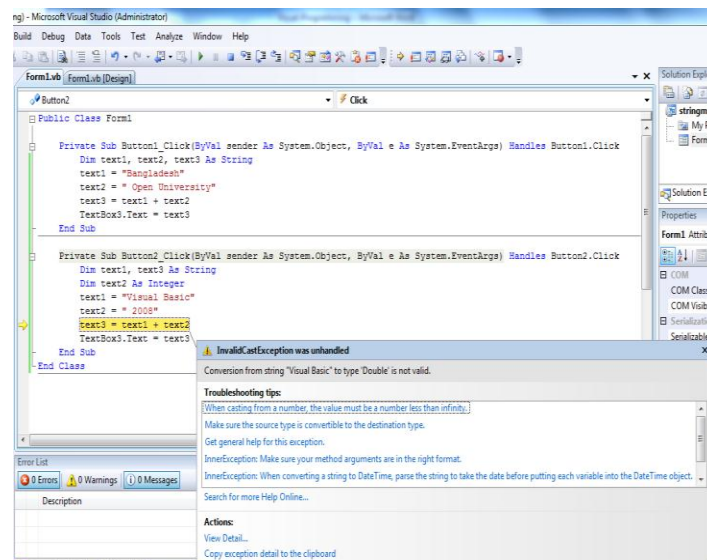
The line `text3=text1+ text2` can be replaced by `text3=text1 & text2` and produced the same output. However, if one of the variables is declared as numeric data type, you cannot use the + sign, you can only use the & sign. An error occurs if you do not following rule, as shown in the following example.



String Manipulation

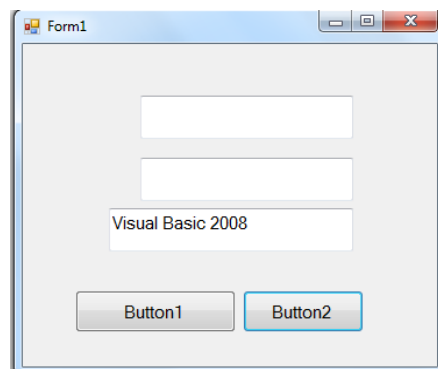
```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim text1, text3 As String
    Dim text2 As Integer
    text1 = "Visual Basic"
    text2 = " 2008"
    text3 = text1 + text2
    TextBox3.Text = text3
End Sub
```

This code will produce an error because of data mismatch.



However, using & instead of + will be all right.

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim text1, text3 As String
    Dim text2 As Integer
    text1 = "Visual Basic "
    text2 = " 2008"
    text3 = text1 & text2
    TextBox3.Text = text3
End Sub
```





String Manipulation Using VB2008 Built-in Functions

There are many string manipulation functions that are built into VB2008 some of that are-

Len Function

The length function returns an integer value which is the length of a phrase or a sentence, including the empty spaces. The format is
Len ("Phrase")

For example,

Len (Visual Basic) = 12

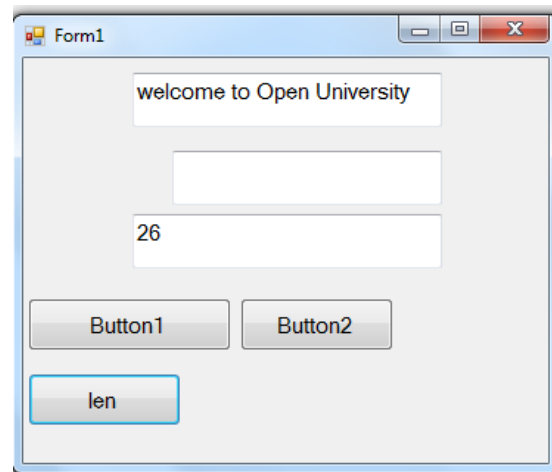
Len (Welcome to Open University) = 26

Example

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    TextBox3.Text = Len(TextBox1.Text)
End Sub
```

Or

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    TextBox3.Text = Len("Welcome to Open University")
End Sub
```



Right Function

The Right function extracts the right portion of a phrase. The format is
Right ("Phrase",n)

Where n is the starting position from the right of the phrase where the portion of the phrase is going to be extracted.

For example,

Microsoft.VisualBasic.Right ("Visual Basic", 5) = Basic

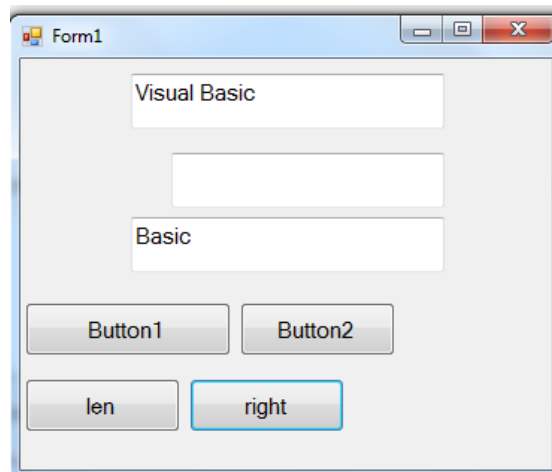
Let us look at another example.

```
Private Sub Button4_Click(ByVal sender As
```




```
System.Object, ByVal e As System.EventArgs) Handles  
Button4.Click  
    Dim text1 As String  
    text1 = TextBox1.Text  
    TextBox3.Text =  
Microsoft.VisualBasic.Right(text1, 5)  
End Sub
```

The above program will return four right most characters of the phrase entered into the textbox, as shown in following figure



Left Function

The Left function extract the left portion of a phrase. The format is `Microsoft.VisualBasic.Left("Phrase",n)` Where n is the starting position from the left of the phase where the portion of the phrase is going to be extracted. For example, `Microsoft.VisualBasic.Left ("Visual Basic", 6) = Visual.`

Mid Function

The Mid function is used to retrieve a part of text form a given phrase. The syntax of the Mid Function is

Mid(phrase, position,n)

Where

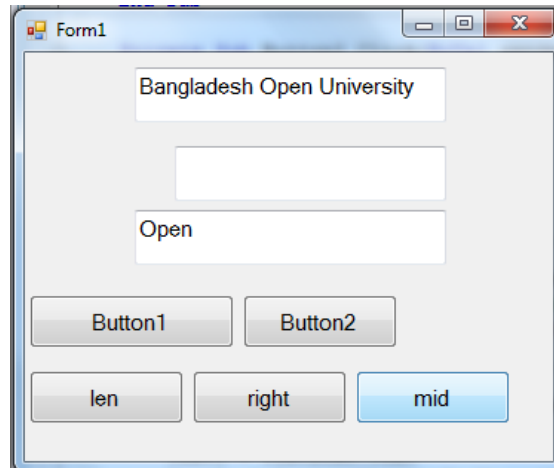
- phrase is the string from which a part of text is to be retrieved.
- position is the starting position of the phrase from which the retrieving process begins.
- n is the number of characters to retrieve.

Example

```
Private Sub Button5_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button5.Click  
    Dim mtext As String  
    mtext = TextBox1.Text
```



```
        TextBox3.Text = Mid(mtext, 12, 4)  
End Sub
```



The Trim Function

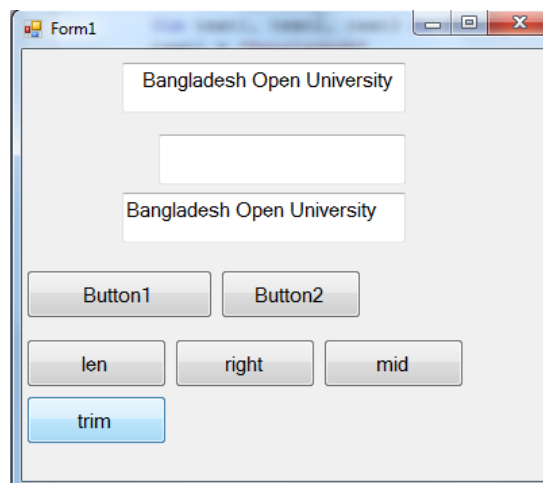
The Trim function trims the empty spaces on both side of the phrase. The syntax is

Trim(“Phrase”)

For example, Trim(“ Visual Basic ”)= Visual basic

Example

```
Private Sub Button6_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button6.Click  
    Dim mtext As String  
    mtext = TextBox1.Text  
    TextBox3.Text = Trim(mtext)  
End Sub
```





The Ltrim Function

The Ltrim function trims the empty spaces of the left portion of the phrase. The syntax is

Ltrim(“Phrase”)

For example,

Ltrim (“ Visual Basic”) = Visual basic

The Rtrim Function

The Rtrim function trims the empty spaces of the right portion of the phrase. The syntax is

Rtrim(“Phrase”)

For example,

Rtrim (“Visual Basic ”) = Visual Basic

Ucase and the Lcase Functions

The **Ucase** function converts all the characters of a string to capital letters. On the other hand, the **Lcase** function converts all the characters of a string to small letters.

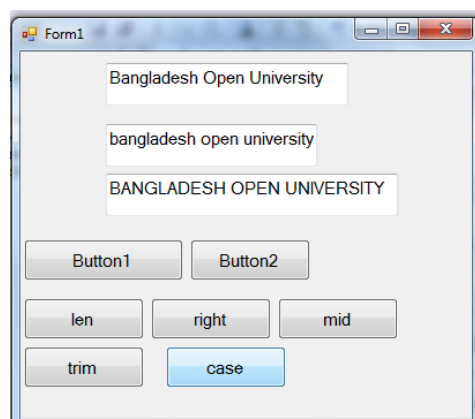
The syntax is

UCase(Phrase)

LCase(Phrase)

For example,

```
Private Sub Button7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
    Dim mtext As String
    mtext = TextBox1.Text
    TextBox2.Text = LCase(mtext)
    TextBox3.Text = UCase(mtext)
End Sub
```





Assessment



Assessment

MCQ

1. Which of the following function returns an integer value which is the length of a phrase or a sentence?

- a) len function
- b) trim function
- c) rtrim function
- d) mid function

Short Questions

1. Which notes of the followings are same?

- i. Len function
- ii. Ucase and Lcase function
- iii. Trim function