

Chapter 7

Controlling Program Flow

-
- ❖ Understanding Conditional and Logical Operators
 - ❖ Using the If control structure with the Comparison Operators
-

In the previous Chapters, we have learned how to write code that accepts input from the user and displays the output without controlling the program flow. In this chapter, you will learn how to write VB2010 code that can make decision when it processes input from the user, and controls the program flow in the process. Decision making process is an important part of programming because it will help solve practical problems intelligently so that it can provide useful output or feedback to the user. For example, we can write a VB2010 program that can ask the computer to perform certain task until a certain condition is met, or a program that will reject non-numeric data. In order to control the program flow and to make decisions, we need to use the conditional operators and the logical operators together with the If control structure.

7.1 Conditional Operators

The conditional operators are powerful tools that can compare values and then decide what actions to take, whether to execute a program or terminate the program and more. They are also known as numerical comparison operators. Normally we use them to compare two values to see whether they are equal or one value is greater or less than the other value. The comparison will return true or false result. These operators are shown in Table 7.1

Operator	Meaning
=	Equal to
>	More than
<	Less Than
>=	More than and equal
<=	Less than and equal
<>	Not Equal to

Table 7.1: Conditional Operators

7.2 Logical Operators

Sometimes we might need to make more than one comparison before a decision can be made and an action taken. In this case, using numerical comparison operators alone is not sufficient, we need to use additional operators, and they are the logical operators. The logical operators are shown in Table 7.2.

Operator	Meaning
And	Both sides must be true
Or	One side or other must be true
Xor	One side or other must be true but not both
Not	Negates truth

Table 7.2: Logical Operators

* Normally the above operators are use to compare numerical data. However, you can also compare strings with the above operators. In making strings comparison, there are certain rules to follows: Upper case letters are less than lowercase letters, "A"<"B"<"C"<"D"...<"Z" and number are less than letters.

7.3 Using the If control structure with the Comparison Operators

To effectively control the VB program flow, we shall use the **If** control structured together with the conditional operators and logical operators. There are three types of **If** control structure, namely **If...Then** statement, **If...Then... Else** statement and **If...Then...Elseif** statement.

7.3(a) If...Then Statement

This is the simplest control structure which ask the computer to perform a certain action specified by the VB expression if the condition is true. However, when the condition is false, no action will be performed. The general format for the if...then... statement is

If condition Then

VB expression

End If

Example 7.1

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

```
    Dim myNumber As Integer
    myNumber = TextBox1.Text
    If myNumber > 100 Then
        Label2.Text = " You win a lucky prize"
    End If
```

```
End Sub
```

When you run the program and enter a number that is greater than 100, you will see the "You win a lucky prize" statement. On the other hand, if the number entered is less than or equal to 100, you do not see any display.

7.3(b) If...Then...Else Statement

Using just If...Then statement is not very useful in programming and it does not provides choices for the users. In order to provide a choice, we can use the If...Then...Else Statement. This control structure will ask the computer to perform a certain action specified by the VB expression if the condition is true. When the condition is false, an alternative action will be executed. The general format for if...then... Else statement is

```
If condition Then
```

```
    VB expression
```

```
Else
```

```
    VB expression
```

```
End If
```

Example 7.2

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

```

    Dim myNumber As Integer
    myNumber = TextBox1.Text
    If myNumber > 100 Then
        Label2.Text = "Congratulation! You win a lucky prize!"
    Else
        Label2.Text = "Sorry, You did not win any prize"
    End If

```

End Sub

When you run the program and enter a number that is greater than 100, it displays a message "Congratulation! You win a lucky prize!" On the other hand, if the number entered is less than or equal to 100, you will see the "Sorry, You did not win any prize" message.

Example 7.3

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

```

    Dim myNumber, MyAge As Integer
    myNumber = TextBox1.Text
    MyAge = TextBox2.Text

    If myNumber > 100 And myAge > 60 Then
        Label2.Text = " Congratulation! You win a lucky prize"
    Else
        Label2.Text = " Sorry, You did not win any prize"
    End If

```

End Sub

This program use the logical **And** operator beside the conditional operators. This means that for the statement to be true, both conditions must be fulfilled in order; otherwise, the second block of code will be executed. In this example, the number entered must be more than 100 and the age must be more than 60 in order to win a

lucky prize, any one of the above conditions not fulfilled will disqualify the user from winning a prize.

7.3(c) If...Then...Elseif Statement

If there are more than two alternatives, using just If...Then...Else statement will not be enough. In order to provide more choices, we can use the If...Then...Elseif Statement. The general format for the if...then... Else statement is

```
If condition Then
    VB expression
ElseIf condition Then
    VB expression
ElseIf condition Then
    VB expression
Else
    VB expression
End If
```

www.knecnotes.co.ke

Example 7.4

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim Mark As Integer
Dim Grade as String
Mark = TextBox1.Text
If myNumber >=80 Then
Grade="A"
ElseIf Mark>=60 and Mark<80 then
Grade="B"
ElseIf Mark>=40 and Mark<60 then
Grade="C"
Else
Grade="D"
End If
End Sub
```

Summary

In this chapter, you learned how to use the If control structure together with the comparison operators to control program flow.

- In section 7.1, you learned how to use the conditional operators in VB2010 such as =, <,>, >=, <= and <>.
- In section 7.2, you learned how to use the logical operators And, Or, Xor and Not.
- In section 7.3, you learned three types of If control structure, i.e. If...Then, If...Then...Else and If...Then...Elseif. Besides, you also learned how to write code involving the use of the If control structure.