Control Structures

In this lesson, you will learn how to write VB.NET code that can make decision when it process input from the users, and control 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 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. There are basically three types of control structures, namely:

- **If - Then**
- **If - Then - Else**
- **If - Then - ElseIf - Then**
- **Select Case**

- **If - Then Statement:**

This is the simplest control structure which asks 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: Write a program to enter the value of two variables (X and Y). Find and print the Maximum value for two variables.

```
Private Sub Button1_Click()
        Dim x, y, max As Integer
        x = InputBox("x")
        y = InputBox("y")
        If x > y Then
            max = x
        End If
        If y > x Then
            max = y
        End If
        MsgBox(max)
 End Sub
```

- **The If – Then - Else statement**

Allows the programmer to specify that a different action is to be performed when a certain action specified by the VB expression if the condition is True then when the condition is false, an alternative action will be executed. The general format for the **If - Then - Else** statement is:

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

Same pervious example:

```
Private Sub Button1_Click()
        Dim x, y, max As Integer
        x = InputBox("x")
        y = InputBox("y")
        If x > y Then
            max = x
        Else
            max = y
        End If
        MsgBox(max)
 End Sub
```

- **Nested (If – Then – Else) statement**

If there are more than two alternative choices, using just If – Then - Else statement will not be enough. In order to provide more choices, we can use If...Then...Else statement inside If...Then...Else structures. The general format for the Nested If...Then.. Else statement is

Method 1	Method 2
If *condition 1* Then *VB expression* 1 Else If *condition 2* Then *VB expression* 2 Else If *condition 3* Then *VB expression* 3 Else *VB expression* End If End If EndIf	If *condition 1* Then *VB expression* ElseIf *condition 2* Then *VB expression* ElseIf *condition 3* Then *VB expression* Else *VB expression* End If

Example: Write a program to enter the value of variable (Mark). Find the grade using If – Block statement and display the value of grade in a text box. When the value of variable (Mark) exceed 100 or less than 0, write a Message Box (Wrong entry, please Re-enter the Mark). Design form window and select all the control objects are used.

```
Private Sub Button1_Click()
        Dim Mark As Single
        Dim Grade As String
        Mark = InputBox("enter the your degree ")
        If (Mark > 100) Or (Mark < 0) Then
            MsgBox("Wrong entry, please Re-enter the mark")
        ElseIf (Mark >= 90) And (Mark <= 100) Then
            Grade = "Excellent"
        ElseIf (Mark >= 80) Then
            Grade = "Very Good"
        ElseIf Mark >= 70 Then
            Grade = "Good"
        ElseIf Mark >= 60 Then
            Grade = "Medium"
        ElseIf Mark >= 50 Then
            Grade = "Pass"
        Else
            Grade = "Fail"
        End If
        TextBox1.Text = Grade
End Sub
```