# Control Structures

Decision making process is an important part of programming because it will help solve practical problems so that it can provide useful output or feedback to the user. For example, we can write a VB6 program that can ask the computer to perform certain task until a certain condition is met. There are three types of If control structures, namely:

- **If …..Then**
- **If – Then –Else**
- **Select Case**

## 1 If....Then Statement:

This is the simplest control structures 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

**1-1** If ***Condition*** Then ***Statement***

Where, Condition is usually a comparison, but it can be any expression that evaluates to a numeric value, this value as true or false. If condition is True, Visual Basic executes all the statements following the Then keyword.

**Example 1**: Write a program to enter the value of two variables (X and Y). Find and print the maximum value for two variables. Design form window and select all the control objects are used.

**Solution(1):**

```
Private Sub Command1_Click
Dim X , Y , Max
X =Val (Text1.Text )
Y =Val (Text2.Text)
Max=X
If Y> X Then Max= Y
Text3.Text= Cstr (Max)
End Sub
```

**or**

**Solution(2):**

```
Private Sub Command1_Click
Dim X , Y , Max
X =Val (Text1.Text )
Y =Val (Text2.Text)
If X> Y Then Max= X
If Y> X Then Max= Y
Text3.Text= Cstr (Max)
End Sub
```

**1-2** If ***condition*** Then ***Goto n***

Where ***n*** : number of statement ( must be Positive Integer value) for example:

Goto 5 , Goto 16 , Goto 2010

**Example 2:** Used (If-Then Goto) condition to write a program for the previous Example1

**Solution(1):**

```
 Dim X ,Y , Max
  X =Val (Text1.Text )
  Y =Val (Text2.Text)
  Max=X
If X> Y Then Text3.Text= Cstr (Max): Exit Sub
   Max=Y
   Text3.Text= Cstr (Max)
   End Sub
```

**Solution(2):**

```
     Dim X ,Y , Max
     X =Val (Text1.Text )
     Y =Val (Text2.Text)
   If X> Y Then  Then Max=X : Goto 10
     Max=Y
10  Text3.Text= Cstr (Max)
     End Sub
```

**Note:** The statement **Exit Sub** used to stop the program without return to the project window.

## 2 If - Block Statement:

**2.1 (If – Then – EndIf) statement:** The If...Then – EndIf Statement performs an indicated action only when the condition is True; otherwise the action is skipped.

```
If condition Then
VB Expression
End If
```

**For example:**

```
 Dim X ,Y , Max
  X =Val (Text1.Text ) : Y =Val (Text2.Text)
 Max=X
If X< Y Then
Max=Y
EndIf
Text3.Text= Cstr (Max)
End Sub
```

**2.2 (If – Then – Else) statement:** 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 than 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
```

**For example:**

```
Dim X ,Y , Max
  X =Val (Text1.Text ) : Y =Val (Text2.Text)
 If X> Y Then
Max=X
Else
Max=Y
EndIf
Text3.Text= Cstr (Max)
End Sub
```

**2.3 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<br>***VB expression***<br>1 Else<br>If ***condition 2*** Then<br>***VB expression***<br>2 Else<br>If ***condition 3*** Then<br>***VB expression***<br>3 Else<br>***VB expression***<br>End If<br>End If<br>EndIf | If ***condition 1*** Then<br>***VB expression***<br>ElseIf ***condition 2*** Then<br>***VB expression***<br>ElseIf ***condition 3*** Then<br>***VB expression***<br>Else 1<br>***VB expression***<br>End If |

**Example 3:** 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, write a Message Box (Wrong entry, please Re-enter the Mark). Design form window and select all the control objects are used.

**Solution:**

```
Private Sub Command1_click()
Dim Mark As Single , Grade as String
Mark = Val (Text1.Text)
If Mark >100 Then
Msgbox "Wrong entry, please Re-enter the mark", Vbcritical , " Error"
Text1.Text="  " : Text2.Text= " "  :  Exit Sub
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
Text2.Text=Grade
End Sub
```