

College of Information Technology
Information network department
Programming with Visual Basic II
lecture 2

=====

Do –Loop:

Use a Do loop to execute a block of statements and indefinite number of times. There are several variations of Do...Loop statement, but each evaluates a numeric condition to determine whether to continue execution. In the following Do..Loop, the statements execute as long as the condition is True.

Do While ..Loop

The formats are **Do While *condition* Block of one or more VB Statement Loop** When Visual Basic executes this Do..Loop, it first tests condition. If condition is False, it skips past all the statements. If it's True, Visual Basic executes the statements and then goes back to the Do while statement and tests the condition again. Consequently, the loop can execute any number of times, as long as condition is True. The statements never execute if initially False.

For Example: Loop counts from 0 to 100.

```
Dim num As Integer, Total
num = 0
Do While num <= 100
Total=Total +num
num = num + 1
Loop
Print Total
```

Do...Loop While:

Another variation of the Do..Loop statement executes the statements first and then tests condition after each execution. This variation guarantees at least one execution of statements. The formats are

Do

Block of one or more VB Statement

Loop *condition*

For Example: Loop counts from 0 to 100.

```
Dim num As Integer, Total
num = 0
Do
Total=Total +num
num = num + 1
Loop While num <= 100
Print Total
```

College of Information Technology
Information network department
Programming with Visual Basic II
lecture 2

=====

Do Until ...Loop

Unlike the **Do While...Loop** repetition structures, the **Do Until... Loop** structure tests a condition for falsity. Statements in the body of a **Do Until...Loop** are executed repeatedly as long as the loop-continuation test evaluates to False. The formats are

Do Until *condition*

Block of one or more VB Statement

Loop

For Example: Loop counts from 0 to 100.

```
Dim num As Integer, Total
num = 0
Do until num >100
Total=Total +num
num = num + 1
Loop
Print Total
```

Do... Loop Until

The formats are

Do

Block of one or more VB Statement

Loop Until *condition*

For Example: Loop counts from 0 to 100.

```
Dim num As Integer, Total
num = 0
Do
Total=Total +num
num = num + 1
Loop until num >100
Print Total
```

Existing Loop:

The exit statement allows you to exit directly from For Loop and Do Loop, Exit For can appear as many times as needed inside a For loop, and Exit Do can appear as many times as needed inside a Do loop (the Exit Do statement works with all version of the Do Loop syntax). Sometimes the user might want to get out from the loop before the whole repetitive process is executed; the command to use is **Exit For** To exit a For.....Next Loop

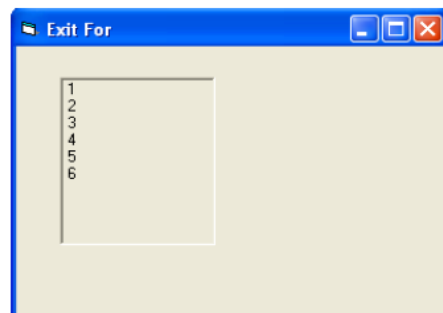
College of Information Technology
Information network department
Programming with Visual Basic II
lecture 2

or **Exit Do** To exit a Do... Loop, and you can place the Exit For or Exit Do statement within the loop; and it is normally used together with the If....Then.....statement.

<ul style="list-style-type: none"> • Exit For <p>The formats are: For <i>counter</i>= start To end step (increment) Statements Exit for Statement Next <i>counter</i></p>	<ul style="list-style-type: none"> • Exit Do <p>The formats are Do While <i>condition</i> Statements Exit Do Statements Loop</p>
--	--

For its application, you can refer to example:

```
1- Private sub Form Load_()
Form1.show
Dim n as Integer
For n=1 to 10
If n>6 then Exit For
Picture1.Print n
Next
End Sub
```



```
1- Private sub Form Load_()
Form1.show
Dim x As Integer
X=0
Do While x < 10
Print x
x = x + 1
If x = 5 Then
Print "The program is exited at x=5"
Exit Do
End If
Loop
End Sub
```

