Conditional Repeating

Very often we need to perform a task repeatedly in order to achieve our objective. This repetitive process is called looping in programming language. Visual Basic allows a procedure to be repeated many times until a condition is met. There are three kinds of loops in VISUAL BASIC, which are **Do…. Loop** , **While… Wend** and **For…Next**.

Do….. Loop

The structure of a Do Loop command can be written in four different formats as shown below:

a) Do While condition

Block of one or more VISUAL BASIC statements

Loop

b) Do

Block of one or more VISUAL BASIC statements

Loop While condition

c) Do Until condition

Block of one or more VISUAL BASIC statements

Loop

d) Do

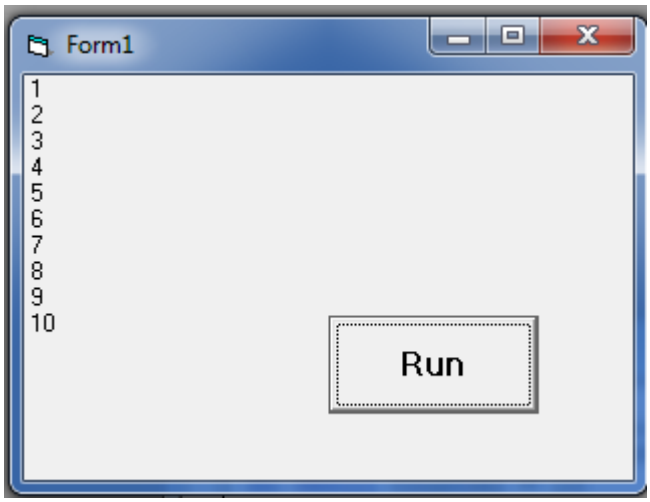
Block of one or more VISUAL BASIC statements

Loop Until condition

Example-1

```
Private sub commond 1_click( )
Do while counter <10
counter =counter+1
 Print Counter
  Loop
End sub
```

In the above example, the value of counter will increase by 1 after each loop and it will keep on adding until counter = 10. The values are displayed in the following figure .



Example-2, Example-3 and Example-4 produce the same result as above.

Example-2

```
Private sub commond 1_click( )
 Do
Counter = Counter + 1
Print Counter
Loop Until Counter = 10
End Sub
```

Example-3

```
Private sub commond 1_click( )
 Do Until Counter = 10
Counter = Counter + 1
 Print Counter
 Loop
 End Sub
```

Example-4

```
Private sub commond 1_click( )
Do
Counter = Counter + 1
Loop While Counter < 10
End Sub
```

Example-5

The following example uses the Do…Loop procedure to find the summation of a sequence of numbers, or in mathematical terms, the summation of an arithmetic progression. In this example, we will attempt to find the summation of 1+2+3+4+……+100. In the design stage, you need to insert a list box into the form for displaying the output, named List1. The program uses the AddItem method to populate the list box. The statement List1.AddItem "n" & vbTab & "sum" will display the headings in the list box, where it uses the vbTab function to create a space between the headings n and sum.

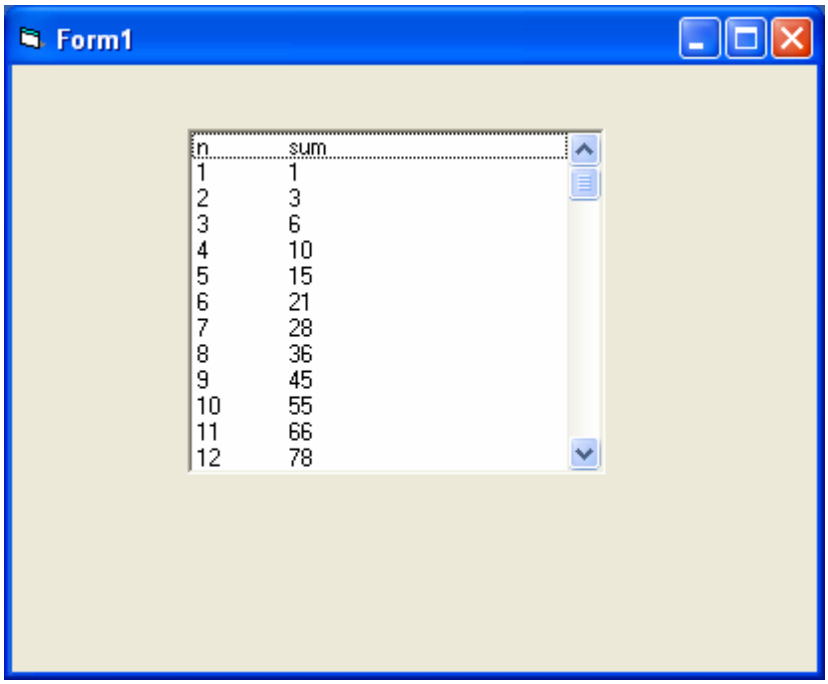
Two variables are declared here, where n acts as a counter and sum is the summation of the numbers. The mathematical logic is very simple. Initially, n and sum are set to 0.

After the first loop, n=1 and sum=1. After the second loop, n will be equal to 2 (n=1+1) and sum will be equal to 3(sum=1+2); and the next loop will produce the result n=3 and sum=6 (sum=1+2+3). Using Do Until n=100, the Program will obtain the summation of 1 to 100. In fact, this program produces the summation at every stage, where the output is displayed in a table form, as shown in following figure.

```
Dim n, sum As Integer

 Private sub commond 1_click( )

List1.AddItem "n" & vbTab & "sum"
Do Until n = 100
n = n + 1
sum = sum + n
List1.AddItem n & vbTab & sum
Loop
End Sub
Private Sub Form_Load ()
n = 0
sum = 0
End Sub
```



The While….Wend Loop

The structure of a While….Wend Loop is very similar to the Do Loop. It takes the following format:

While condition

Statements

Wend

The above loop means that while the condition is not met, the loop will go on. The loop will end when the condition is met. Let's examine the program listed in following example, where it produces the same result as example-5 .

Example-6

```
Dim sum, n As Integer

Private Sub command_click( )

List1.AddItem "n" & vbTab & "sum"

While n <> 100

n = n + 1

Sum = Sum + n

List1.AddItem n & vbTab & Sum

Wend

End Sub
```