

Loops:

Loops provide the ability to repeatedly execute the same block of code, and to each time change values such that each run through the loop produces different results.

Visual Basic provides four main kinds of loops:

- 1- The classic Do-Loop.
- 2- The Do-Until Loop.
- 3- The Do-While Loop.
- 4- The while-Wend loop.
- 5- The For-Next Loop.

Do-Loops:

The most basic form of loop in Visual Basic is the Do-Loop. Its construct is very simple:

Do

(Code to execute)

Loop

This, quite simply, executes the block of code, and when it reaches Loop, returns to the beginning of the Do Loop and executes the same block of code again. The same block of code will be repeatedly executed until it is told to stop executing. The problem here is the program will never stop, this is known as the problem of the infinite do-loop, one that all programmers will experience, and some quite frequently.

So the programmer must use the order (Exit Do) to stop executing the codes of the program.

Exit Do:

It is very important to find a way to escape from the Do-Loop. You could, of course, simply End the program once you have calculated enough values, but what if you still need to perform tasks after you're done calculating? The answer is to use the (Exit Do) statement.

Whenever your program reaches an (Exit Do) statement within a loop, it will exit the current loop. For more understanding let us try to write codes for the examples below:

Ex: Write a program that is able to print the numbers (0, 32, 64, 128, 256, 512, 1024, 2048) on the form vertically, use the font (Times New Roman), (Bold) with the size (14). Use Do ... Loop and Exit Do statements.

- 1- Design part: No design part for such example.
- 2- Coding part: The coding part might be written as below:

```
Private Sub Form_Activate( )
```

```
Font.Name = "times new roman"
```

```
Font.Bold = True
```

```
Font.Size = 14
```

```
    cnt = 1
```

```
    x = 0
```

```
    Print x
```

```
    x = 32
```

```
    Do
```

Print x

If cnt >= 7 Then

Exit Do

Else

cnt = cnt + 1

x = x + x

End If

Loop

End Sub

Ex: Write a program that is able to print the numbers (0, 15, 23, 31, 39, 47, 55, 63, 71, 79, 87, 95, 103, 111, 119) on the form vertically, use the font (Times New Roman), (Bold) with the size (14). Use Do ... Loop and Exit Do statements.

- 1- Design part: No design part for such example.
- 2- Coding part: The coding part might be written as below:

```
Private Sub Form_Activate( )
```

```
Font.Name = "times new roman"
```

```
Font.Bold = True
```

```
Font.Size = 14
```

```
cnt = 1
```

```
x = 0
```

```
Print x
```

x = 15

Do

Print x

If cnt > 13 Then

Exit Do

Else

cnt = cnt + 1

x = x + 8

End If

Loop

End Sub