

For updated version, please click on
<http://ocw.ump.edu.my>

COMPUTER PROGRAMMING

REPETITION (LOOPS) – 1

by
LIM KAR SING

FACULTY OF CIVIL ENGINEERING & EARTH RESOURCES
UNIVERSITI MALAYSIA PAHANG

limks@ump.edu.my

Repetition (Loops)

- Do Loops
- For...Next Loops
- List Boxes and Loops

Do Loops

- Pretest Form of a Do Loop (Do While... Loop)
- Posttest Form of a Do Loop (Do... Loop Until)
- A Financial Calculation

Do Loops

- A loop is one of the most important structures in computer programming.
- Used to repeat a sequence of statements a number of times.
- The Do Loop repeats a sequence of statements either *as long as* or *until* a certain condition is true/achieved.

Pretest Do Loop

Do While *condition*
statement (s)

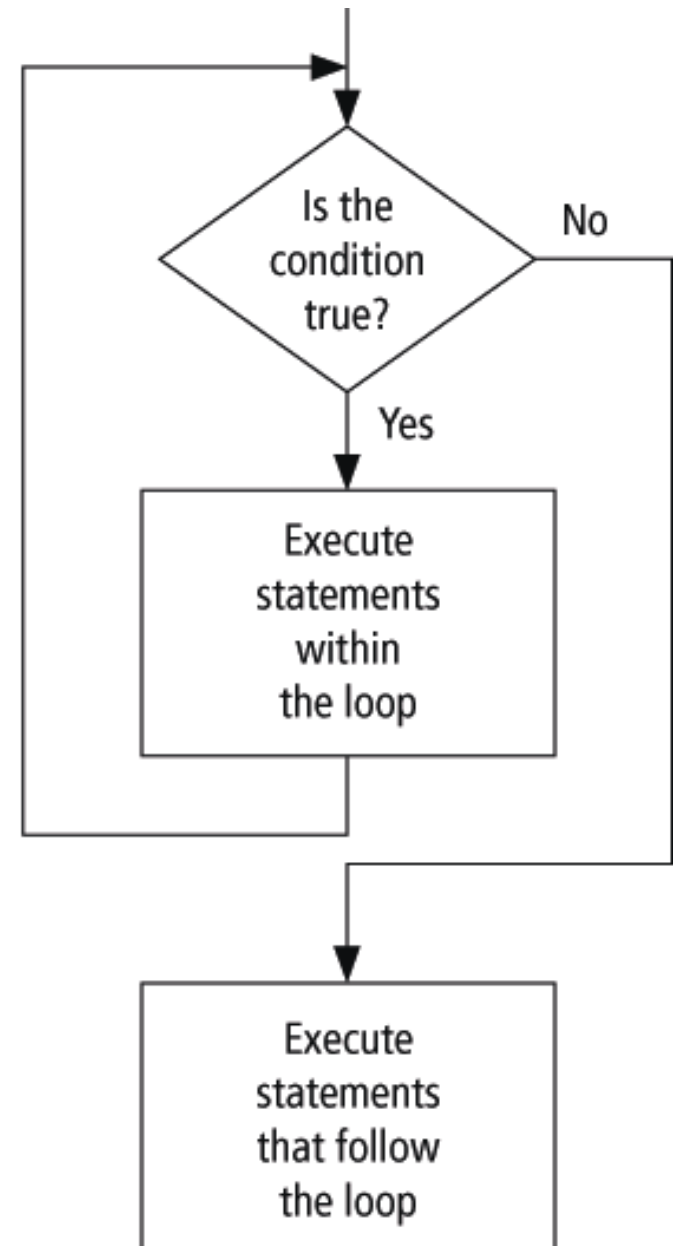
Loop

These statements are inside the body of the loop and are run if the condition above is true.

Condition is tested.
If it is true,
the loop is run.
If it is false,
the statements
following the
Loop statement
are executed.

Pseudocode and Flow Chart

Do While condition is true
Processing step(s)
Loop

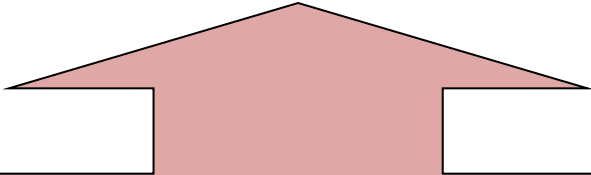


Posttest Do Loop

Do

statement(s)

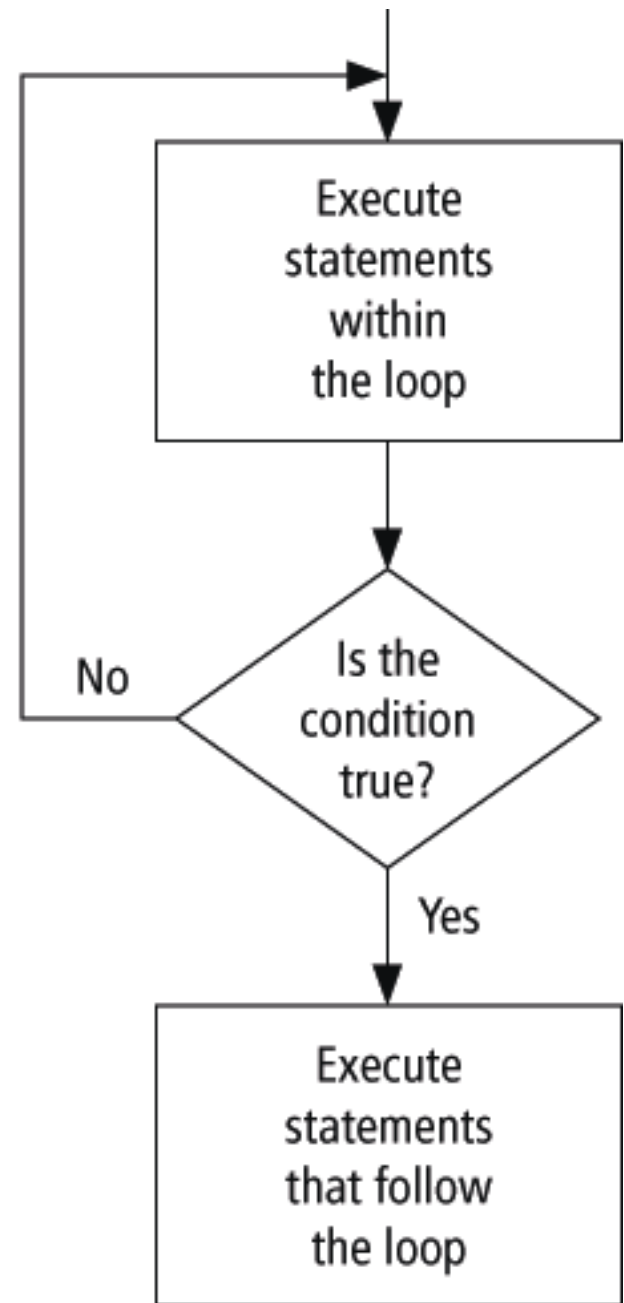
Loop Until *condition*



Loop is executed once and then the condition is tested. If the condition is false, the loop is run again. If it is true, the statements following the Loop Until statement are executed.

Pseudocode and Flowchart

Do
statement(s)
Loop Until condition is true



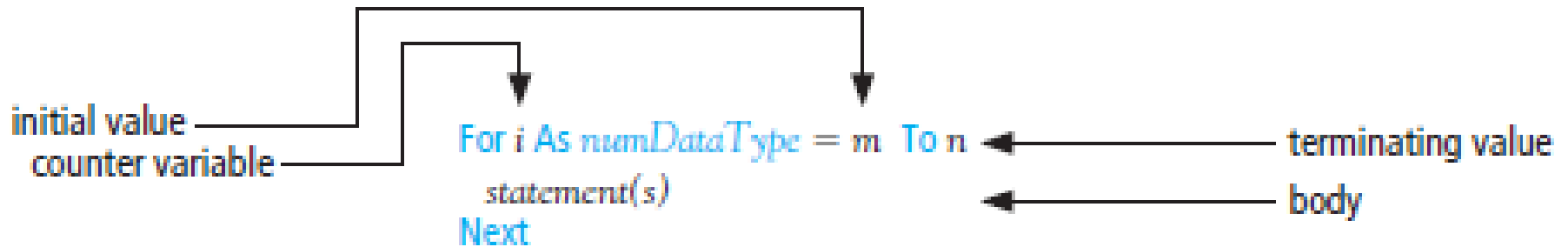
Comments

- Be careful to avoid **infinite** loops – loops that never end.
- Visual Basic allows for the use of either the **While** keyword or the **Until** keyword at the top or the bottom of a loop.

For... Next Loops

- General Form of a For...Next Loop
- Step Keyword
- Nested For...Next Loops
- Local Type Inference
- Used when we know how many times we want the loop to execute
- A counter controlled loop

For... Next Loops Syntax



For **VariableName** As **DataType** = **Start_Value** To **Stop_Value** Step **Increment**

Label1.Text = counter

ListBox1.Items.Add(counter)

Next

Sample Syntax

```
For i As Integer = 1 To 5  
    lstTable.Items.Add(i)  
Next
```

The loop counter variable, *i*, is

- initialized to 1
- tested against the stop value, 5
- incremented by 1 at the Next statement

Similar Do While Loop

```
Dim i As Integer = 1
Do While i <= 5
    lstTable.Items.Add(i)
    i += 1
Loop
```

Step Keyword

- Normally after each pass the value of the counter variable increases by 1
- If **Step s** is appended to the For statement, the value of **s** will be added to the counter variable after each pass.
- If the value of **s** is a negative number, the value of the counter variable will decrease after each pass.

Example with Negative Step Value

```
For j As Integer = 10 To 1 Step -1  
    ListBox.Items.Add(j)  
Next  
ListBox.Items.Add("Time")
```