COMP1022Q
Introduction to Computing with Excel VBA

Introduction to Looping

David Rossiter, Gibson Lam, Eddie Chan

What is Looping?

- A loop is a set of code which repeats many times

- Looping is a very useful feature in programming languages such as VBA because it makes repetitive work easier

- For example, you can use a nested loop to generate a chess board

- In this presentation we will look at two types of loops:
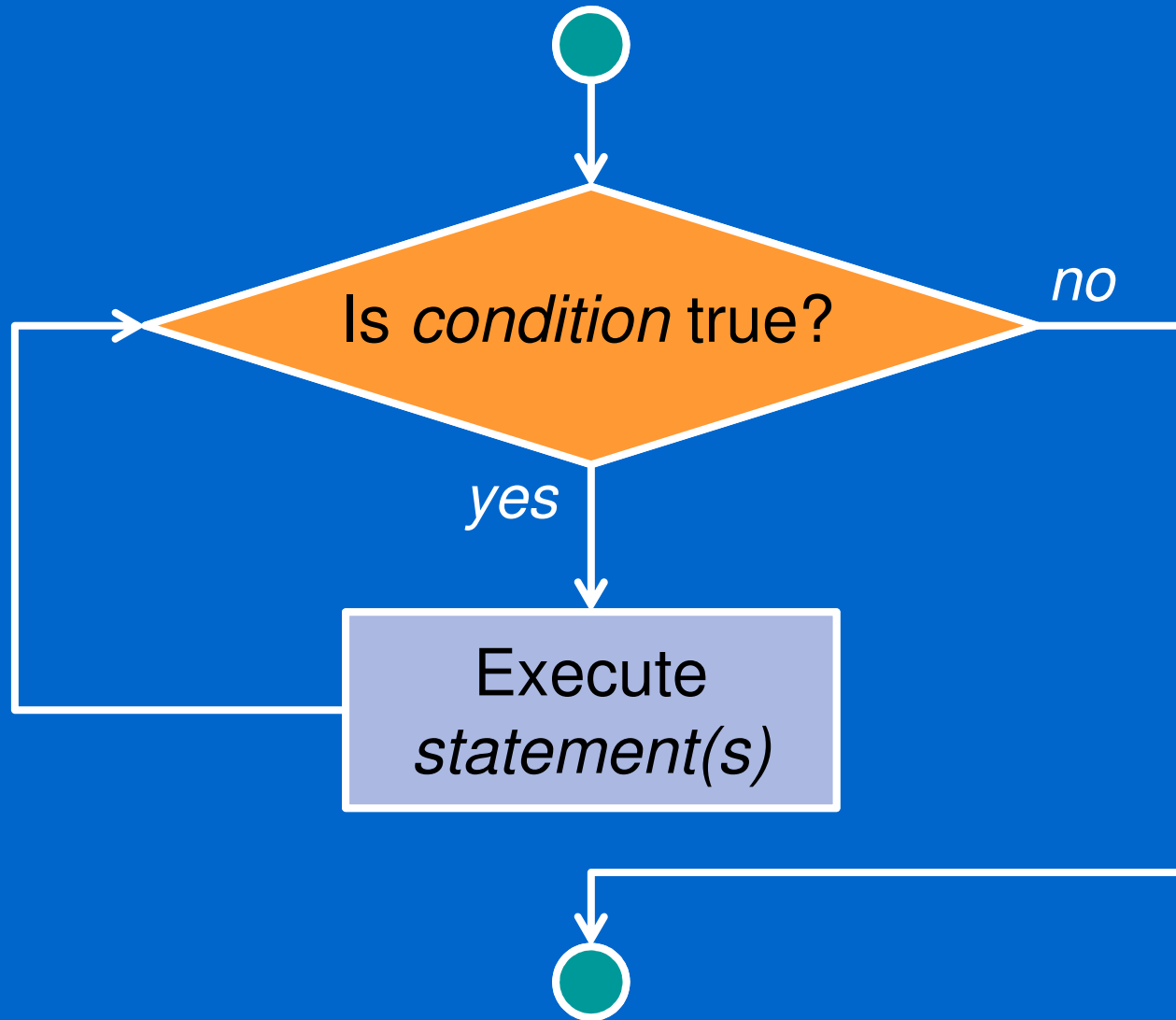
  - While Loops

  - Do Loops
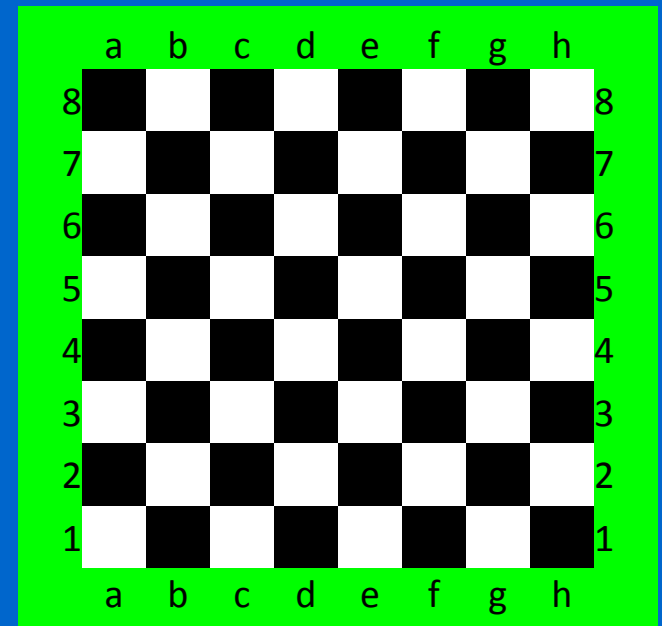
While…Wend

`While` *...condition...*

      ....statement(s)...

`Wend`

- While *condition* is true, repeatedly execute *statement(s)*
- When *condition* is false, *While…Wend* finishes

The Flow of While…Wend

Reading/Writing Worksheet Cells Using Range()

- You have seen the use of `Range()` to read and write a cell value, for example:

```
' Put the value of cell B4 into MyVar
MyVar = Range("B4").Value

' Put the text "Hello" into cell C4
Range("C4").Value = "Hello"
```

- `Range()` uses a cell reference, e.g. `A4`
- It can also refer to multiple cells like this:

```
' Copy the content of five cells in column A
Range("A6:A10").Copy
```

Alternative Way to Read/Write Worksheet Cells

- Another way to read/write a cell is to use `Cells()`
- For example, to read cell `B1` with `Cells()`, you need to do this:

  ```
  MyVar = Cells(1, 2).Value
  ```

  The row and column number of the cell

- The good thing about `Cells()` is that you only need to give the row and column number of the cell you want
- However, that means you can only refer to one single cell using `Cells()`

Working with
the Currently Selected Cell

- Sometimes you want to do something with the currently selected cell but you don't know which cell is currently selected

- In VBA, `ActiveCell` is the currently selected cell

- For example, the following line of code changes the font of the currently selected cell to bold:

```
ActiveCell.Font.Bold = True
```

Useful Things You Can Do With a Cell

- Find out the row number or column number of the cell

```
Range("B4").Row      ' Return 4
Range("E8").Column   ' Return 5
```

- Change the currently selected cell

```
Cells(5, 4).Select  ' Select cell D5
```

- Refer to a cell relative to another cell

```
Range("D7").Offset(1, 0)   ' Return cell D8
                           '  (move down one row)
Range("B1").Offset(2, 0)   ' Return cell B3
                           '  (move down two rows)
Range("A5").Offset(0, 2)   ' Return cell C5
                           '  (move right two columns)
Range("E6").Offset(0, -2) ' Return cell C6
                           '  (move left two columns)
```

An Example of While…Wend (1/2)

```
' Select the first cell in the 4th row
Cells(4, 1).Select
```

Loop condition

```
While   ActiveCell.Row <= 10
```

Loop body

```
        ' Put the row number into the first cell
        ActiveCell.Value = "I am row " & _
                        ActiveCell.Row



        ' Move the selection down one cell
        ActiveCell.Offset(1, 0).Select
Wend
```

This line of code selects the cell on the row following the currently selected cell

An Example of While…Wend (2/2)

	A	B	C	D	E
1	**Showing Row Numbers from Row 4 to Row 10**				
2	*This example puts the row number in the first cell of each row from row 4 to row 10. The VBA code is run when you open the worksheet.*				
3					
4	I am row 4				
5	I am row 5				
6	I am row 6				
7	I am row 7				
8	I am row 8				
9	I am row 9				
10	I am row 10				
11					
12					

```
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                    ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```
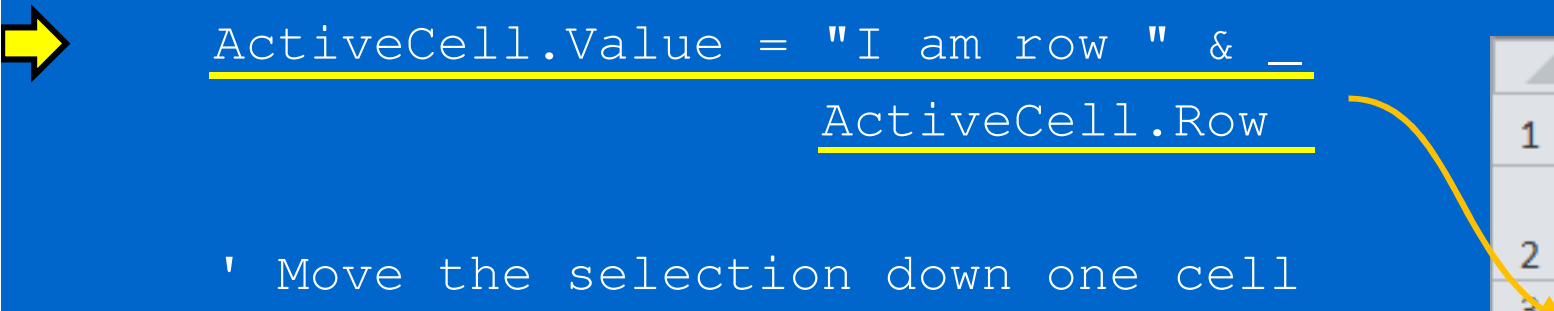
The first cell in row 4 is selected

	A	B
1	**Showing Row N**	
2	*This example puts the* *10. The VBA code is*	
3		
4		
5		
6		
7		
8		
9		
10		
11		

```
' Select the first cell in the 4th row
Cells(4, 1).Select

→ While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                  ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

The condition is true because the active cell is in row 4 (<=10)

```
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                        ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

Running the While…Wend Example (4/10)

```vba
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                    ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```
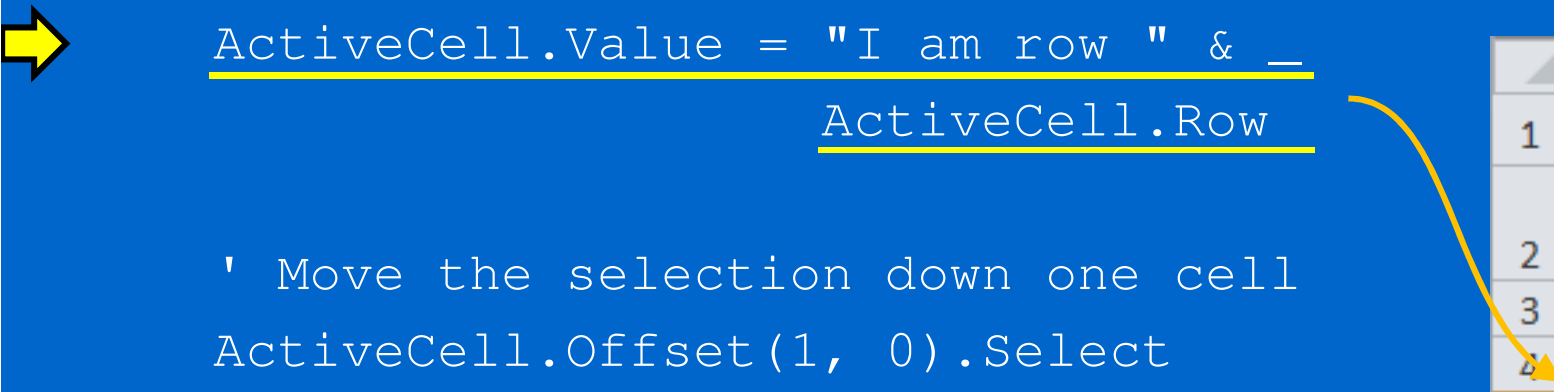
The cell on the next row is selected

```
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                    ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

The condition is true because the active cell is now in row 5

Running the While...Wend Example (6/10)

```vba
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                        ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

◢	A	E
1	**Showing Row N**	
2	*This example puts the* *10. The VBA code is*	
3		
4	I am row 4	
5	I am row 5	
6		
7		
8		
9		
10		
11		

```
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                    ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

The cell on the next row is selected

Running the While…Wend Example (8/10)

```
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                    ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

The condition is false when the active cell is in row 11

	A	B
1	**Showing Row N**	
2	*This example puts the* *10. The VBA code is*	
3		
4	I am row 4	
5	I am row 5	
6	I am row 6	
7	I am row 7	
8	I am row 8	
9	I am row 9	
10	I am row 10	
11		

```
' Select the first cell in the 4th row
Cells(4, 1).Select

While  ActiveCell.Row <= 10
    ' Put the row number into the first cell
    ActiveCell.Value = "I am row " & _
                       ActiveCell.Row


    ' Move the selection down one cell
    ActiveCell.Offset(1, 0).Select
Wend
```

Since the condition is false the contents of the loop do not get executed, and the loop finishes

|	A	B
1	**Showing Row N**	
2	*This example puts the 10. The VBA code is*	
3		
4	I am row 4	
5	I am row 5	
6	I am row 6	
7	I am row 7	
8	I am row 8	
9	I am row 9	
10	I am row 10	
11		

Do While…Loop

`Do While` ...*condition*...

...*statement(s)*...

`Loop`

- The usage of *Do While…Loop* is exactly the same as *While…Wend*
- The difference is that the keywords of *Do While…Loop* are more 'English' than *While…Wend*

The Flow of Do While…Loop

- This flowchart is the same as the flowchart of *While…Wend*

Is *condition* true?

no

yes

Execute *statement(s)*
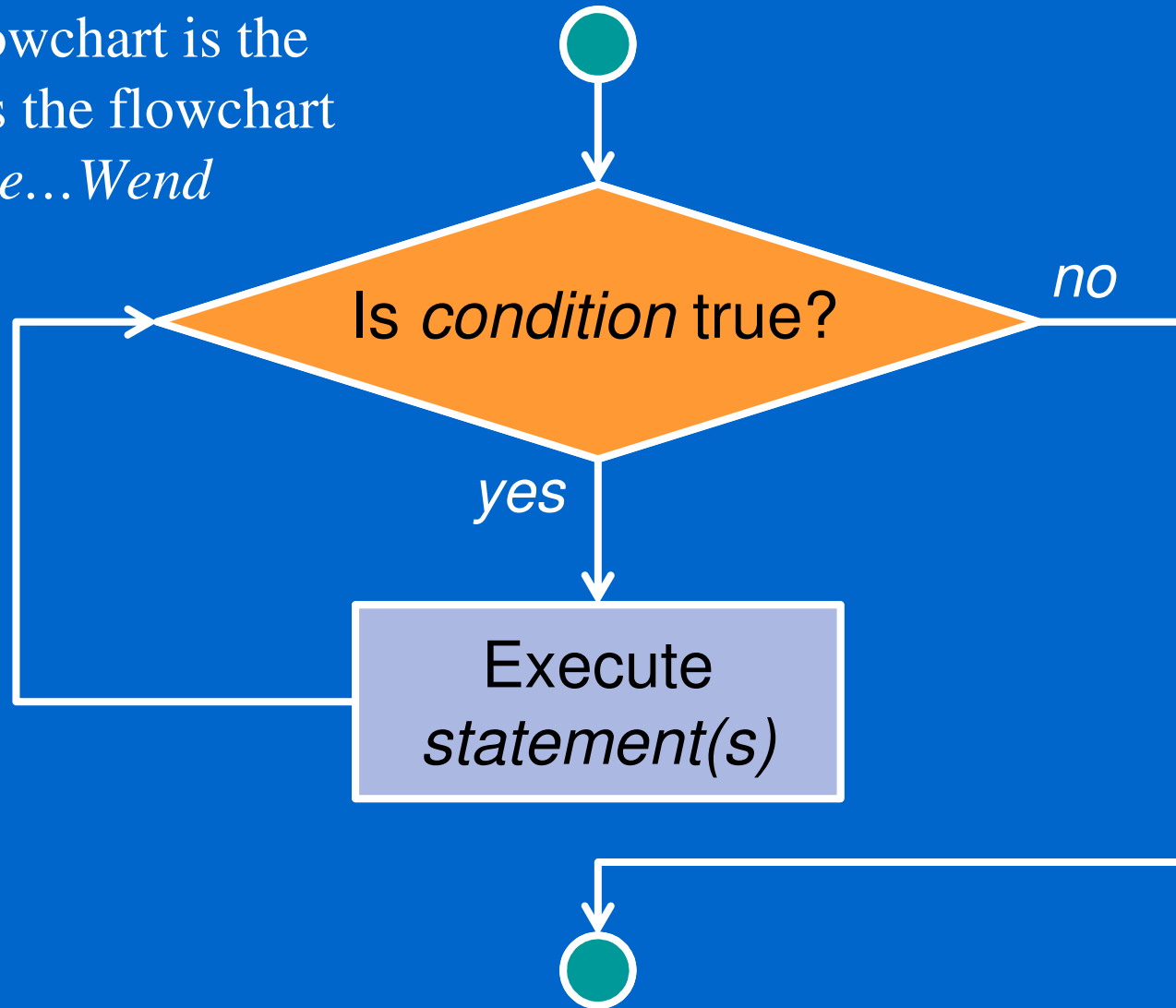
Example of Do While…Loop

- For example, we can create a program which does the same thing as the *While…Wend* example using *Do While…Loop*:

```
' Select the first cell in the 4th row
Cells(4, 1).Select

Do While   ActiveCell.Row <= 10
        ' Put the row number into the first cell
        ActiveCell.Value = "I am row " & _
                            ActiveCell.Row


        ' Move the selection down one cell
        ActiveCell.Offset(1, 0).Select
    Loop
```

Loop condition

Loop body
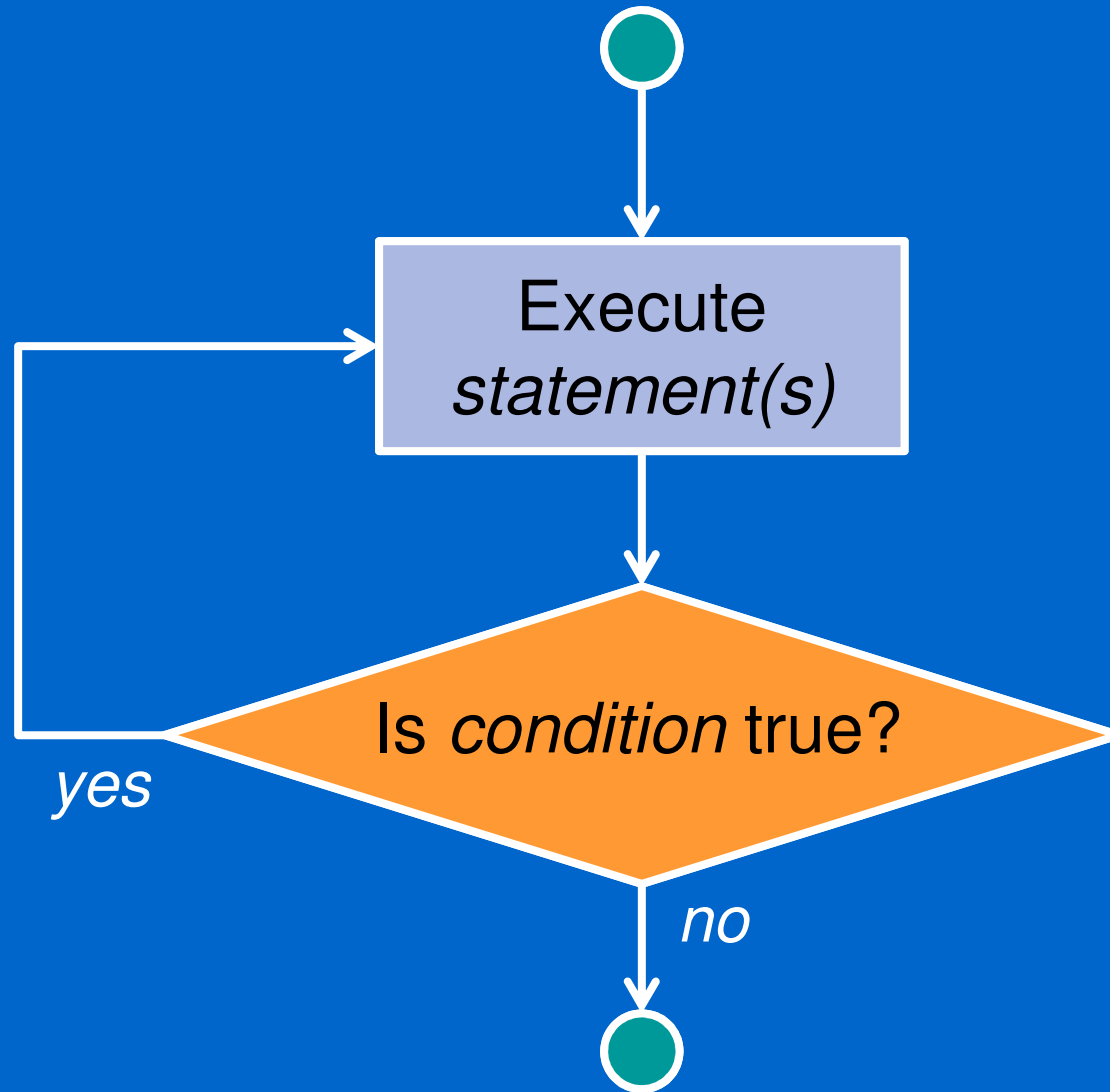
Do…Loop While

```
Do

    ...statement(s)...
Loop While ...condition...
```

- This is similar to the previous two loops we looked at but *condition* is evaluated **after** *statement(s)* is executed

- This means that *statement(s)* will be executed at least once before *condition* is evaluated

The Flow of Do…Loop While

An Example of Do…Loop While (1/3)

```
' Execute the loop at least once
Do
     ' Ask a question
     Answer = InputBox("Do you think
          comp1022q is a great course?")


' Check the answer at the end of the loop
Loop While  LCase(Answer) <> "yes"
```
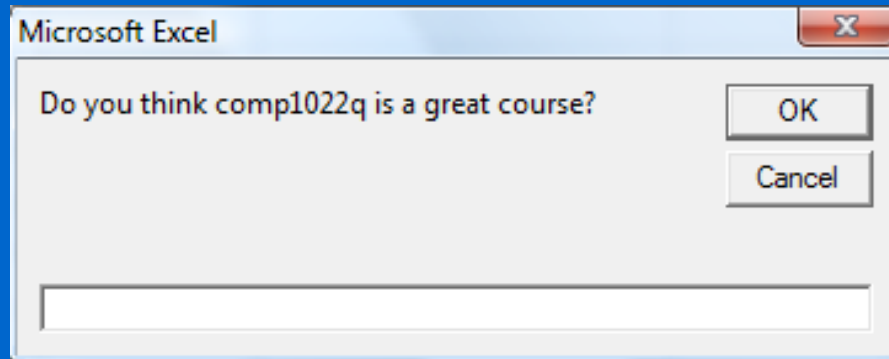
Loop body

Loop condition

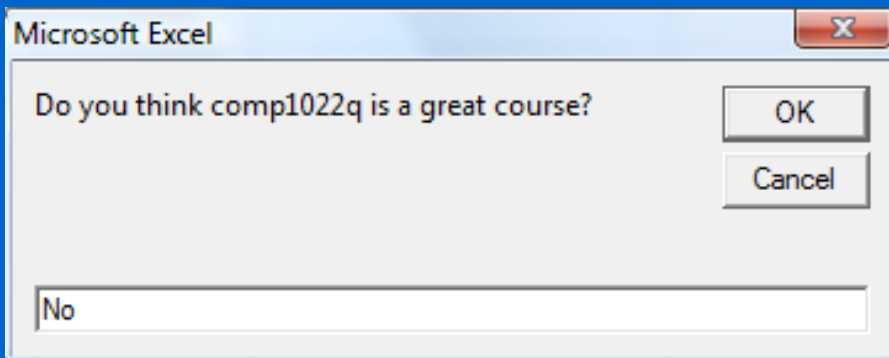*The LCase() method changes all letters in some text into lower case letters, e.g.:*
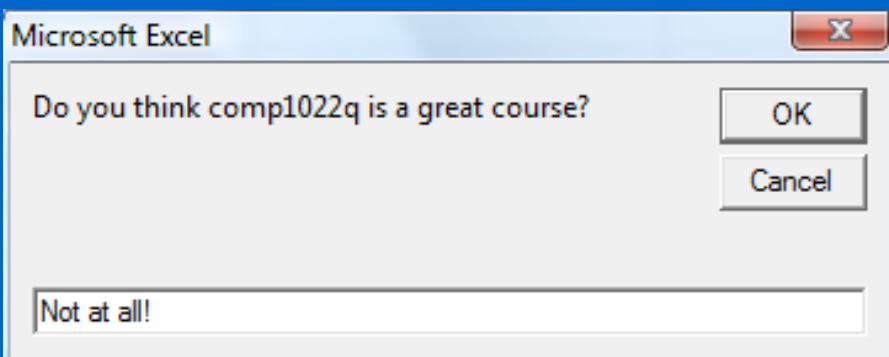
```
LCase("BIG") → "big"
LCase("How Are You") → "how are you"
```

An Example of Do…Loop While (2/3)

Microsoft Excel

Do you think comp1022q is a great course?

OK

Cancel

You need to answer a question in an InputBox

Microsoft Excel

Do you think comp1022q is a great course?

OK

Cancel

No

If you answer 'No' you will be asked again

Microsoft Excel

Do you think comp1022q is a great course?

OK

Cancel

Not at all!

If you answer 'Not at all!' you will be asked again

An Example of Do…Loop While (3/3)

Microsoft Excel
Do you think comp1022q is a great course?
OK
Cancel
I don't think so.
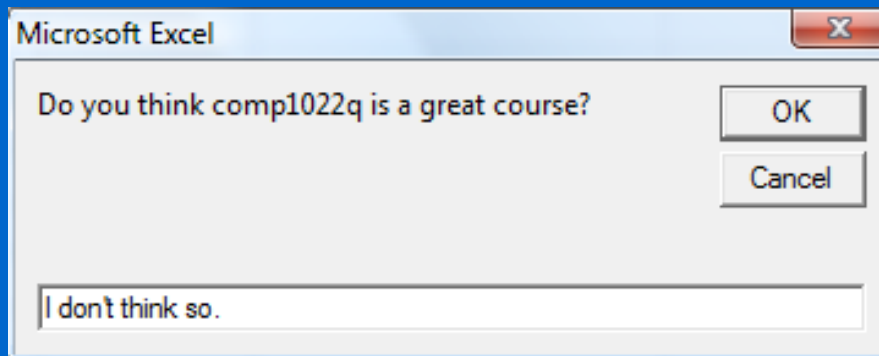
If you answer 'I don't think so' you will be asked again

Microsoft Excel
Do you think comp1022q is a great course?
OK
Cancel
Yes
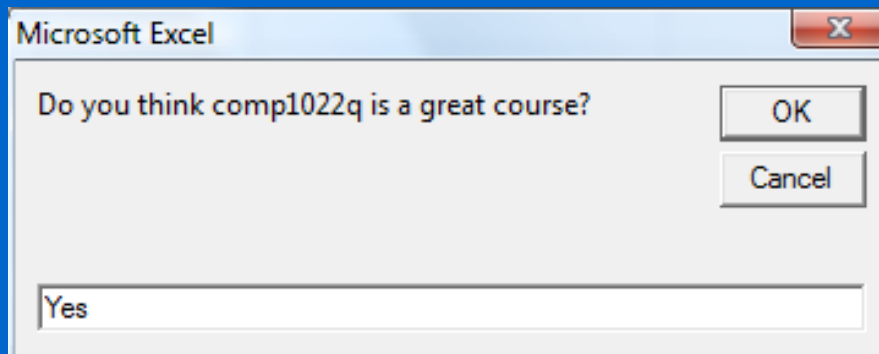
You will not be asked the question again if you answer 'Yes'

- In this example, if you answer 'Yes', 'yes', 'YES' they will all stop the InputBox from showing up again