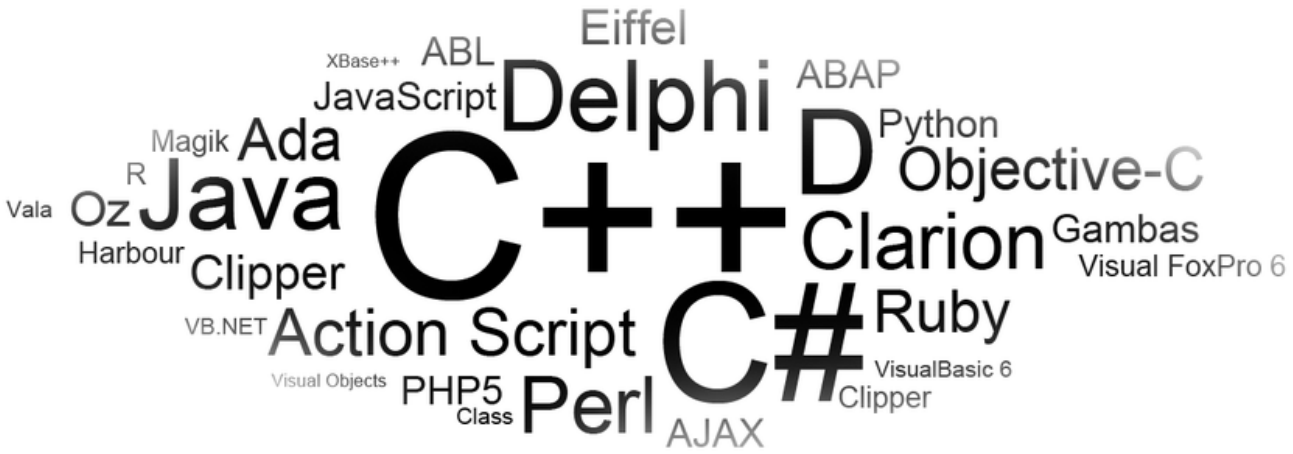


Day 2 – 26 May 2015 (3hrs)

VBA Programming for MIF

Siraprapa Watakit
Technical Instructor,
BSc(Computer),MSc(Finance), PhD Student(Finance)

Day 1 - RECAP

- **What have we learnt last time?**
  - Record macros
  - Implement SUBROUTINE vs. FUNCTION
  - Datatype in VBA
    - Primitive: Integer, Single, Double and etc..
    - Object: Workbook, Worksheet and etc..
  - VBA **I**ntegrated **D**evelopment **E**nvironment, aka IDE(ALT+F8)
    - Project Explorer(Sheets, Modules, Class, Forms and etc..)
    - Properties Windows
    - Immediate Windows
    - Watch Windows
  - Debugging code with F8(Step into), breakpoint, watch window and etc..

Day 2 – Agenda

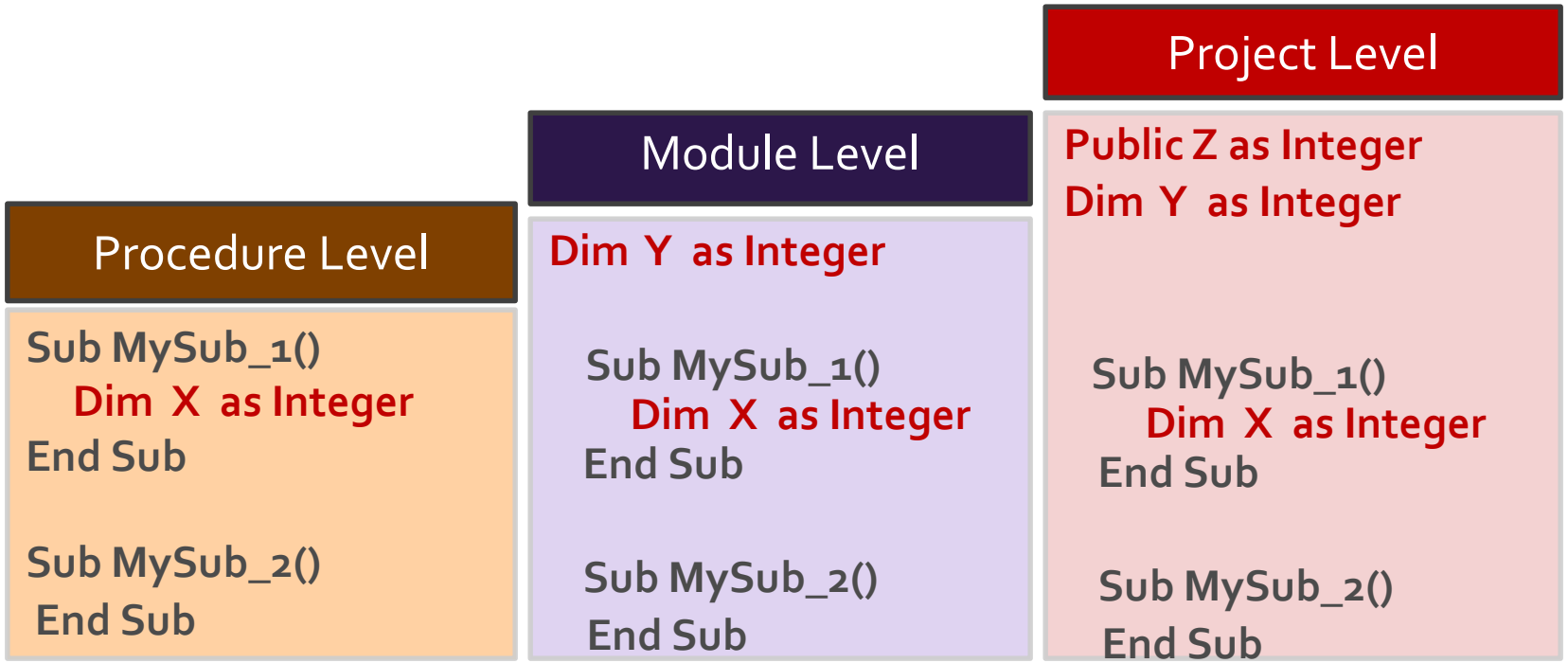
Module 2 – Part 2.1 and Part 2.2

- **Fundamental VBA Programming Concept**
  - Fundamental of Programming
    - More on Structural Programming
      - If, If-then-else
      - For Loop, While Loop
  - Workshops
    - Working with Excel Objects

Module 2
Fundamental VBA Programming Concept
Part 2.1
Structural Programming

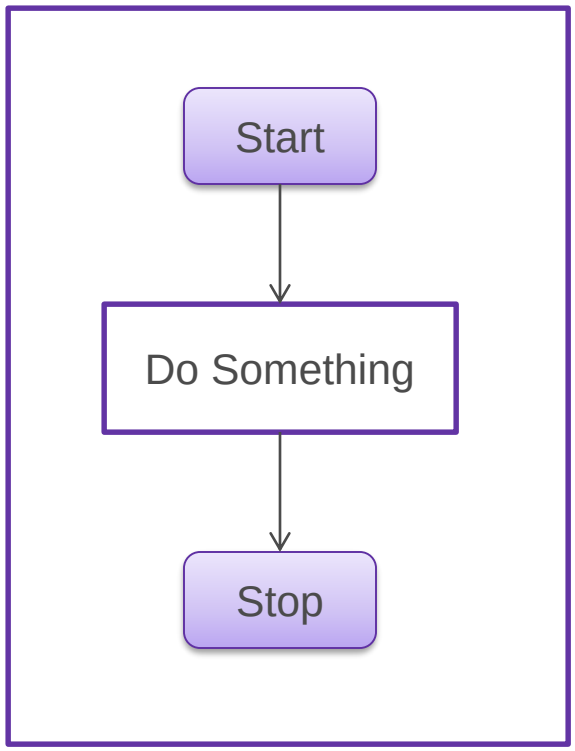
Scope and Lifetime of a Variable

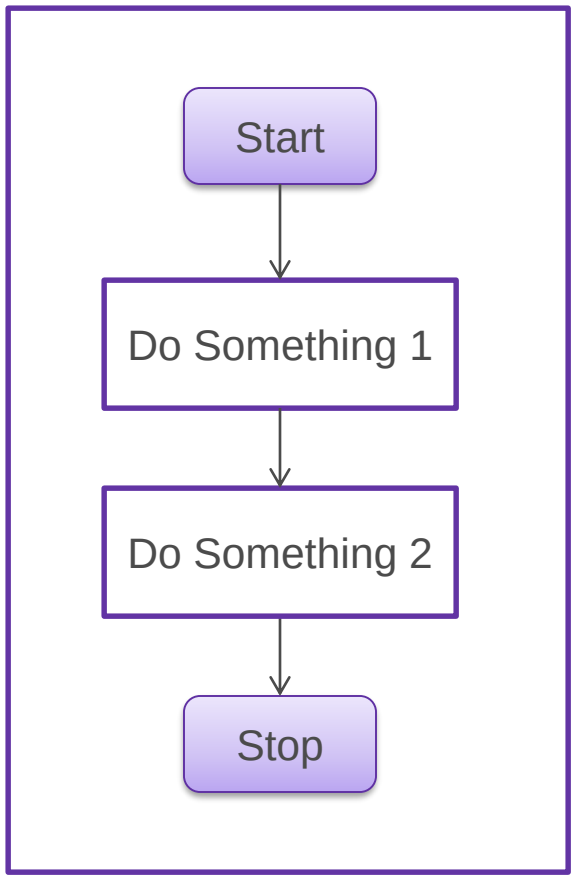
- **Visibilities**
  - X is visible only in **MySub_1**
  - Y is visible in every **sub/fnc in that module(MySub_1,MySub_2)**
  - Z is visible in ***everywhere***, **including other sub/fnc in *other modules***



Flowcharts and Structural Programming

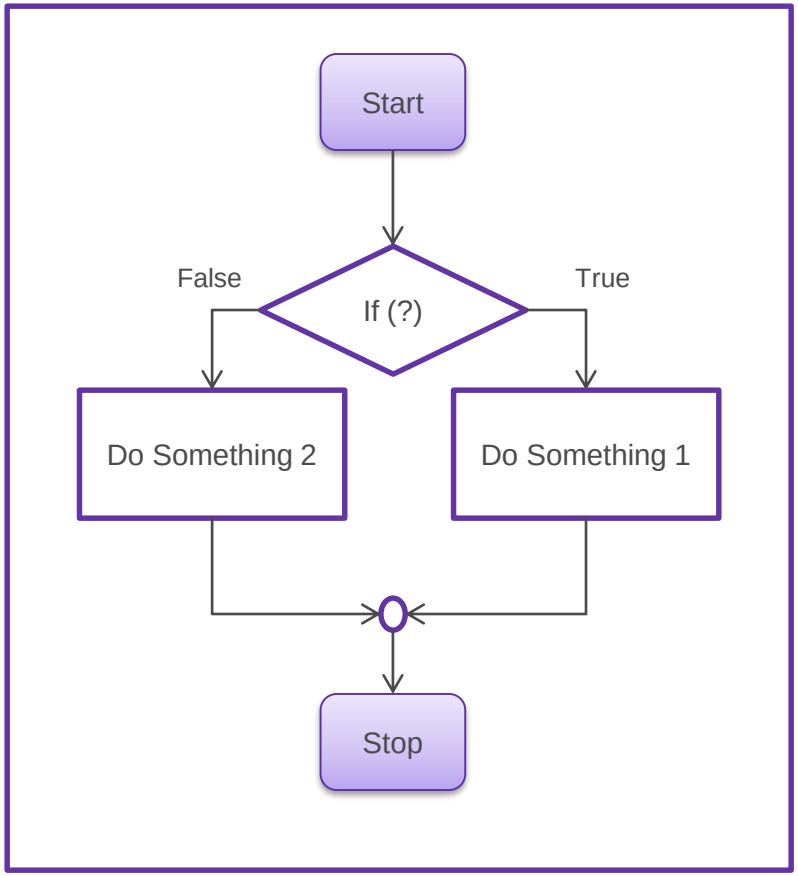
- **Simple Flowcharts**





Flowcharts and Structural Programming

- If…Then…Else




```
If <condition> Then
    <cmd>
Else
    <cmd>
End If
```

```
If <condition> Then
    <cmd>
Elseif <condition> Then
    <cmd>
Else
    <cmd>
End If
```

Exercise – 1

- **Implement a program that will display message as according to these rules**
  - Input: age
  - Output:
    - if age is 62 or above, display "you can apply for senior social security"
    - if age between 22..61, display "you can drink and vote"
    - if age between 18..21, display "you can vote"
    - other than that, display message "you cannot drink and vote"
  - Note: There are at least 2 ways to implement this program!

Flowcharts and Structural Programming

- **Select Case**

```
Select Case <condition>
    Case <condition>,<condition>,…
            <cmd>
    Case <start> To <stop>
            <cmd>
    Case [Is >,>=,<,<=] <condition>
            <cmd>
    [Case Else]
            <cmd>
End Select
```

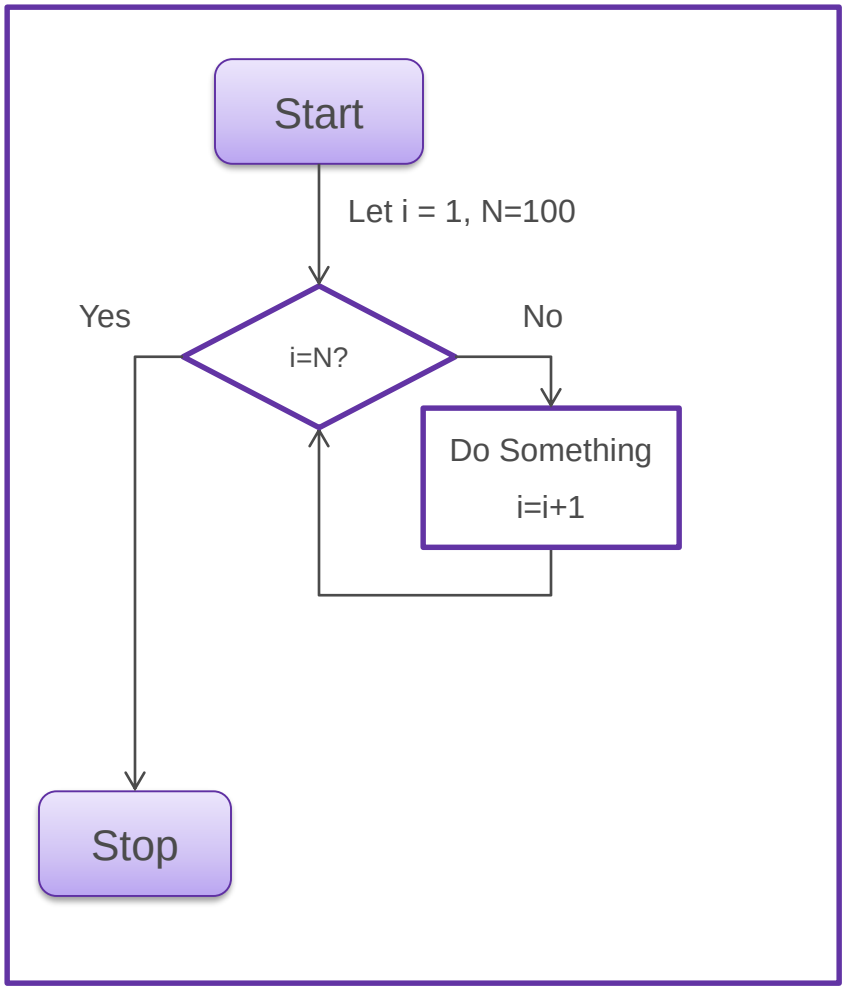
Exercise – 2

- **Implement a program which display message as the following rules**
  - Input: 1 character from InputBox (N, E, S, W)
  - Display
    - If input=N, display "North"
    - If input=E, display "East"
    - If input=S, display "South"
    - If input=W, display "West"
    - Else, display "Unknown Region"
- **According to this grading system, assign grades for each student**
  - SimpleGrading.xlsx

Grade Label	Limit for e.g >=275 gets A
A	275
B	250
C	225
D	200
F	<200

Flowcharts and Structural Programming

- **For Loop**…You know exactly how many rounds to go..




```
Dim i As Integer
For i=1 to 10 [Step 1]
     Debug.Print "i: "; i
[Exit For]
Next
```

```
Dim i As Integer
For i=10 to 1 Step -1
    Debug.Print "i: "; i
    [Exit For]
Next
```

Exercise – 3

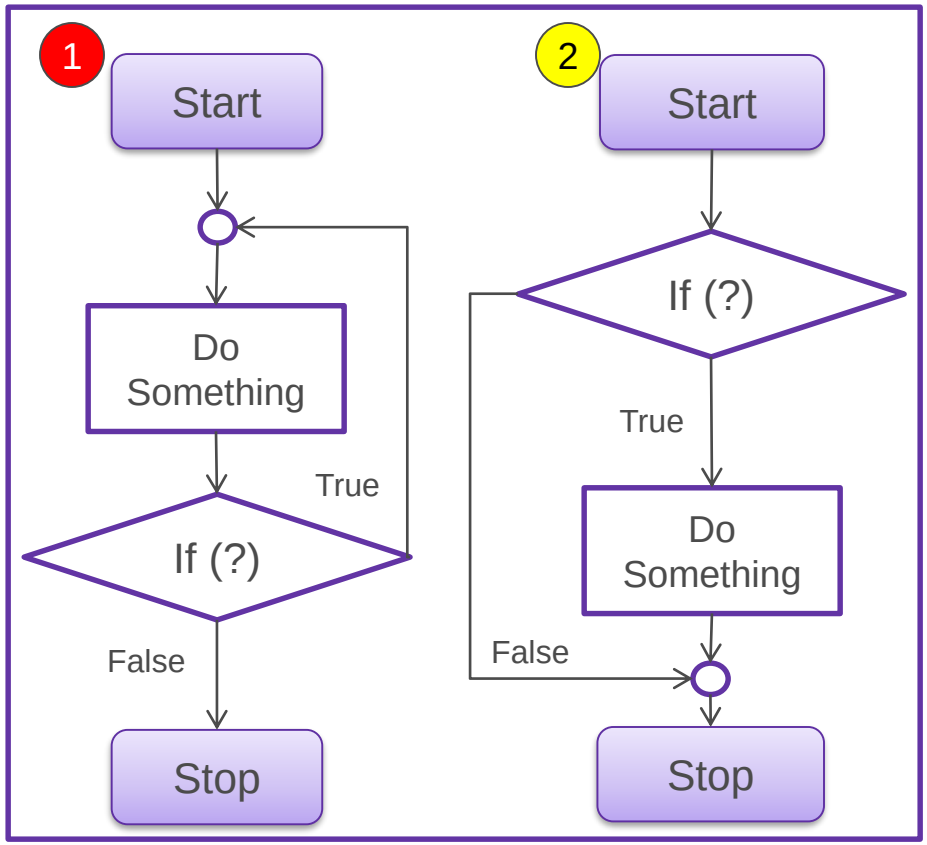
- **Implement a FUNCTION which calculate a factorial value of a number. For example 5!=5x4x3x2x1=120**

- **Implement a FUNCTION which will summaries the number in between. For example, start=2 stop=10, answer=2+3+4+5+6+7+8+9+10=54**

- **Create a FUNCTION that will accept an INTEGER number and determine whether that number is a prime number or not (Definition: Prime number, like 5, can only be divided by 1 and itself i.e. 5)**
  - Hint: For Loop, Mod fnc..

Flowcharts and Structural Programming

- **Do Loop**, you don't really know how many round to go but stop when a condition is met



1

```
Dim ans as Integer
ans=inputbox("input ans=")
Do
    MsgBox "Your ans=" & ans
    ans=inputbox("input ans=")
    [Exit Do]
Loop While ans>0
```

2

```
Dim ans as Integer
ans=inputbox("input ans=")
Do While ans>0
    MsgBox "Your ans=" & ans
    ans=inputbox("input ans=")
    [Exit Do]
Loop
```

The difference btw the 2 program is that, the first one check condition last, the second one check condition first. Which means, the first one will get executed at least 1 time

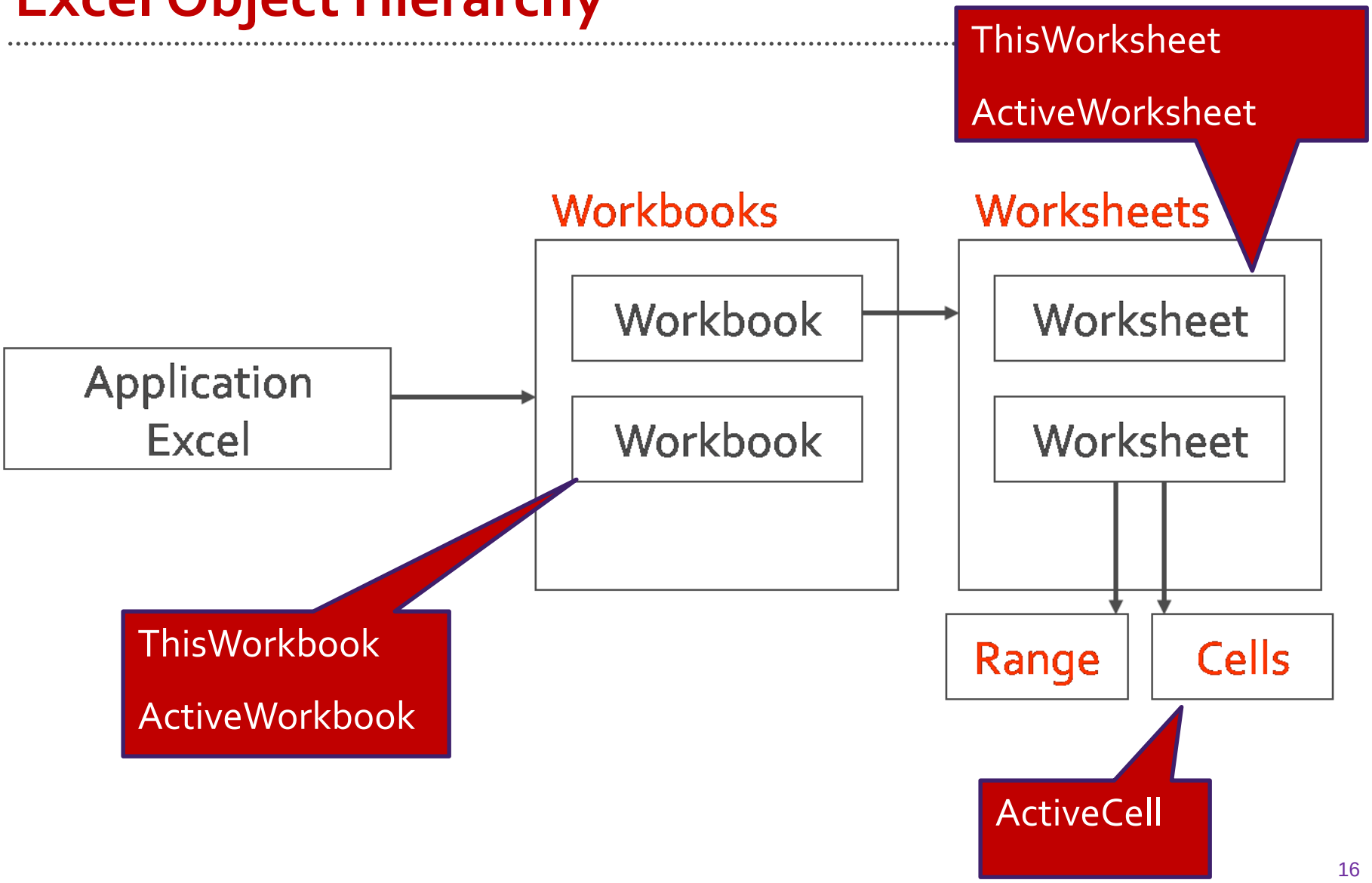
Exercise – 4

- **Create a MAGIC NUMBER game. At first, the program will randomly select an integer between 1..99, then the player will be asked to guess that number, REPEATEDLY. The program will end if either (1) player guesses it correctly, or (2) player quits**
  - Hint: Do While, magic=round(rnd*100,0)
  - Suggestion, write down your ***Pseudo Code and flowcharts*** here…

Module 2
Fundamental VBA Programming Concept

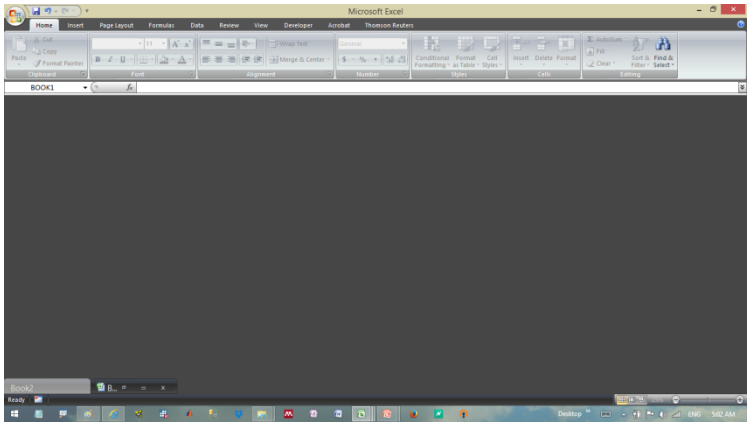
Part 2.2
Excel Objects

Excel Object Hierarchy



Application

- **The Excel Application itself**
- **What can you do with the Excel?**
  - Tell me how many workbooks are currently open
  - What is the active workbook name?
  - Open/Close workbooks
  - **Turn on/off warning alerts e.g. "Do you want to save before exit?"**
  - **Turn on/off screen updating e.g. "Show screen only after calculation"**
  - Display username, program path and etc…



Application

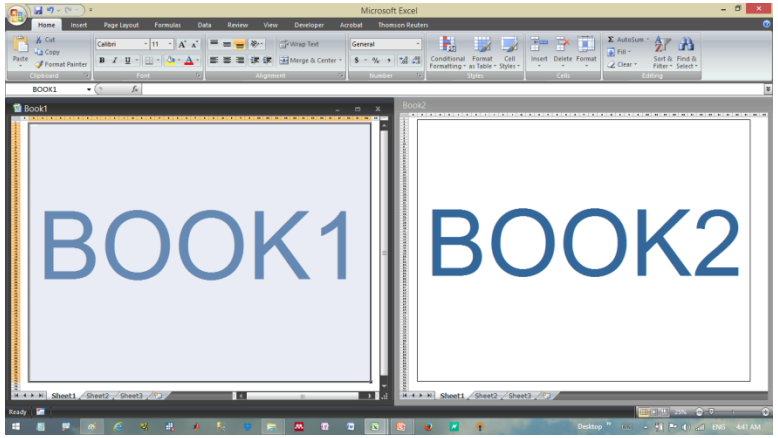
```
Sub MyApplication()
    Debug.Print Application.UserName
    Debug.Print Application.ActiveWorkbook.Name
    Debug.Print Application.ThisWorkbook.Name
    Debug.Print Application.Workbooks.Count
    Debug.Print Application.WorksheetFunction.StDev(100, 200)
End Sub
```

```
Sub TurnOff()
'Turn off all alerts/screenupdate bits
    Application.ScreenUpdating=False
    Application.DisplayAlerts=False
End Sub

Sub TurnOn()
'Once finished, turn everything  back on..
    Application.ScreenUpdating=True
    Application.DisplayAlert=True
End Sub
```

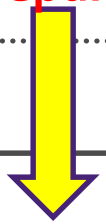
Workbooks

- **The one that is ACTIVE is "ActiveWorkbook";The one that macro is written on is "ThisWorkbook"**
- **What can you do with workbook?**
  - Open, Save, Close
  - Tell me how **many worksheets** in a workbook
  - **Insert/Delete** worksheet
  - Add/Delete **Named Ranges**..



Workbooks

Prepare this



- Open/Save/Close a file

```
Sub OpenAFile()
    Workbooks.Open Filename:="C:\Book1.xls"
End Sub
```

```
Sub SaveAFile()
    Workbooks("C:\Book1.xls").Save
End Sub
```

```
Sub CloseAFile()
   Workbooks("Book1.xls").Close
End Sub
```

Workbooks

- Insert/Delete worksheet

Always make sure you are in ThisWorkbook

```
Sub InsertWorksheet()
    Dim last As Integer
    ThisWorkbook.Activate
    last=ThisWorkbook.Sheets.Count
    ActiveWorkbook.Sheets.Add After:=Sheets(last), _
         Count:=1, Type:=xlWorksheet
End Sub
```

```
Sub DeleteWorksheet()
    Dim last As Integer
    ThisWorkbook.Activate
    last=ThisWorkbook.Sheets.Count
    ThisWorkbook.Sheets(last).Delete
End Sub
```

Workbooks

- Working with defined named ranges

```
Sub AddNames()
    On Error Resume Next
    ThisWorkbook.Names.Add Name:="MyRange", _
    RefersTo:=Sheet1.Range("F10:J10")
    Range("MyRange")=Rnd
End Sub
```

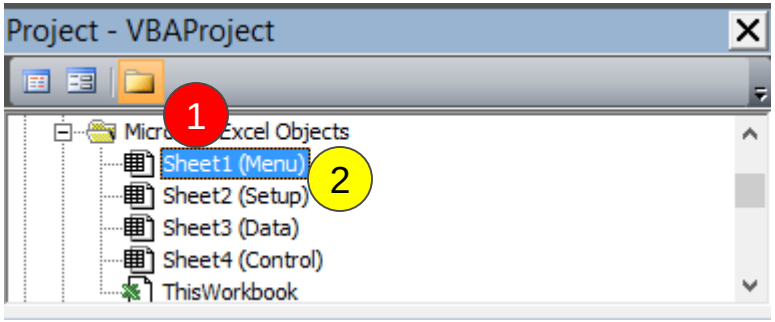
```
Sub DeleteNames()
     On Error Resume Next
    ThisWorkbook.Names("MyRange").Delete
    End Sub
```

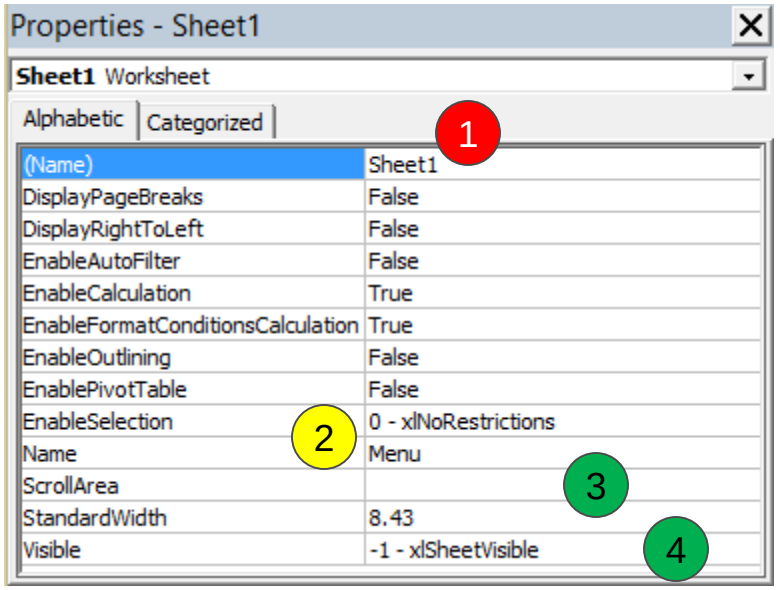
Worksheets

- **Each worksheet contain "Cells", to make a reference to a cell, you can do so by invoke**
  - Range("C3")="Helllo World", or,
  - Range("C5:C10")=" Hi Hi", or,
  - Cells(10,10)="Look Here!!!"
- **What can you do with worksheets?**
  - Activate, or select, a worksheet
  - Select/copy/paste ranges in a worksheet
  - Insert text/number/formula in a cell or range
  - Get the value from a cell
  - Rename worksheets

Worksheets

- **There are quite many ways to access a worksheet,**
  - Via Object Name 1
  - Via Sheet Name 2
  - Via Sheet Index (1…4)





- **Worksheet also comes with useful properties**
  - ScrollArea: A1:Z100 3
  - Visible: xlSheetVisible, xlSheetHidden, xlSheetVeryHidden 4

Worksheets

- Worksheets

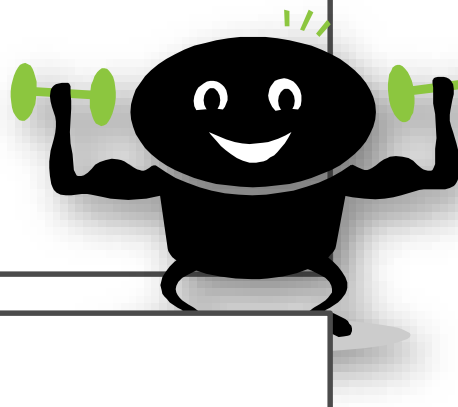
reference by object name

reference by sheet name

```
Sub MyWorksheets()

    MsgBox Sheet1.Name
    Worksheets("Menu").Range("F5:F10")="ABC DEF"

    Range("F5:F10").Select
    With Selection
        .Interior.Color = RGB(255, 64, 38)
    End With
End Sub
```



```
Sub PrintSheetsName()
    Dim i As Integer
    For i = 1 To ThisWorkbook.Worksheets.Count
        Debug.Print Sheets(i).Name
    Next
End Sub
```

reference by sheet index

Worksheets

- Copy and Paste Range

```
Sub CopyPasteRange()
    Sheets("Data").Activate
    Range("C5:D10").Select
    Selection="Dummy Data"
    Selection.Copy
    Sheets("Control").Activate
    Range("C5").Select
    ActiveSheet.Paste
    Application.CutCopyMode = False
End Sub
```

- Clear Range

```
Sub DeleteRange()
    Sheets("Data").Activate
    Range("C5:D10").Select
    Selection.Clear
End Sub
```

- Select Range, alternatively..

```
Sub SelectRanges()
    Range(Range("A1"),Range("C10")).Select
    Selection="Blah.."
    Range(Range("A1"), Range("A1").End(xlDown)).Select
End Sub
```

Worksheets

- Get value from a Range

```
Sub GetValue()
    Dim myvalue As Variant
    'Fill in any value or fnc at Range("A1")
    myvalue=Sheet1.Range("A1").Value
    MsgBox "Value is Sheet1.Range(A1) is " & myvalue
End Sub
```

- Working with 2D Matrix - REVISITED

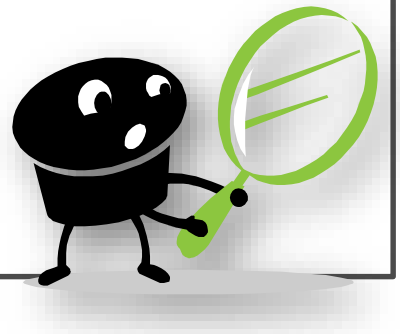
```
Sub PasteMatrix()
    Dim mymatrix() As Double
    Dim i As Integer : Dim j As Integer : Dim size As Integer
    Range(Range("B5"), Range("B5").End(xlDown).End(xlToRight)).Clear
    size = InputBox("Enter Size")
    ReDim mymatrix(1 To size, 1 To size)
    For i = 1 To size
        For j = 1 To size
            mymatrix(i, j) = Rnd
        Next
    Next
    Range("B5").Resize(size, size) = mymatrix
End Sub
```

Worksheets

- **For Each…Next, this will come in handy if you work with excel objects..**

```
Sub ForEach_Range()
Dim o As Range
For Each o In Range("A1:A10")
        Debug.Print o.Address
        Debug.Print o.Row
        o="=Rand()"
Next
Debug.Print Range("A1:A10").Rows.Count
End Sub
```

```
Sub ForEach_Worksheet()
Dim o as Worksheet
For Each o In ThisWorkbook.Worksheets
    Debug.Print o.Name
Next
End Sub
```



Exercise – 5 Monte Carlos Simulation

- **Call/Put Option Pricing**
  - Suppose we assume stock price follow GBM where

$$S_{dt} = S_0 \times e^{[(\mu - \frac{\sigma^2}{2})dt + \sigma\sqrt{dt}\varepsilon]}, \varepsilon \sim N(0,1)$$

$$c = e^{-r_f T} \times E[\max(ST - K, 0)]$$

$$p = e^{-r_f T} \times E[\max(K - ST, 0)]$$

  - If a closed-form solution is unknown to us and Numerical Method can be cumbersome, **Monte Carlo Simulation** can easily be done
    - After all, it is basically a FOR LOOP
    - Source files: **MIF VBA Day 2 - MCSim.xlsm**
  - **Compare with MatLab code?(mcmodel.m)**

Final Programming Notes

- When assign Excel Objects to a variable…always use Set

```
Sub MySub()
    Dim abook As Workbook
    Dim arange As Range
    …
    Set abook = ActiveWorkbook
    Set arange = Range("A1:A10")
    …
    Set abook = Nothing
    Set arange = Nothing
End Sub
```

release memory

Exercise – 6

- **Create a program that INSERT A NUMBER OF SHEETS as according to user input**
  - Then, create a program that **DELETE** those sheets
  - Note, you **cannot DELETE** ALL sheets, **there must be at least 1 sheet left**
- **Create a program that HIDE all sheet**
  - Note, you **cannot** HIDE ALL sheets, **there must be at least 1 sheet visible**
  - Then, create program that **SHOW** those sheets
- **Create a program that OPEN MULTIPLE file**
  - Hint:

```
Dim getFiles As Variant
    getFiles = Application.GetOpenFilename _
               ("Excel Files (*.xls), *.xls*", _
                MultiSelect:=True)
```

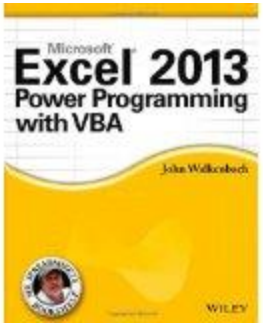
  - Then, create program that **CLOSE** those opened files

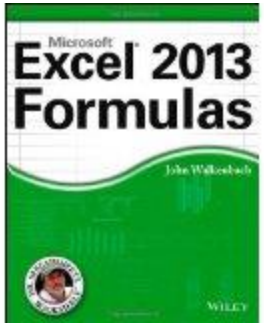
Preparation for Day 3

- **Download and take a look at these RAW DATA**
  - Bloomberg : https://db.tt/QMCTacEF
  - Datastream : https://db.tt/BYRXIzUt
  - Eikon: https://db.tt/dzGQnoxA
- **Self Study Portfolio Optimization and Portfolio Simulation**
  - PortOptSim.xlsx
- **Think! If you are to clean the data, what do you need to do, what are your thought process?**

Recommendations

- John Walkenbach(Mr.Spreadsheet)





- Bill Jelen(Mr.Excel)