ADVANCED MS EXCEL COURSE
SESSION 11 & 12
SAMPLE VBA CODES

1. SELECT SHEET

```
Sheets("sheet_name ").Select
```

2. HIDE SHEET

```
Sheets("sheet_name").Visible = False
```

3. UNHIDE SHEET

```
Sheets("sheet_name").Visible = True
```

4. SELECT CELL (Example Select A1 cell)

```
Range("A1").Select
```

5. ENTER A VALUE TO SELECTED CELL

```
Range("A1").value = "VALUE"
```

6. INFORMATION MESSAGE BOX

```
If Range("A1").Value <> "" Then
VBA.Interaction.MsgBox "Your Msg", , "Msg_Heading"
End If
```

7. MESSAGE BOX WITH YES NO BUTTONS

```
If Range("A1").Value <> "" Then
Dim answer As VbMsgBoxResult
    answer = MsgBox("Your_question?", vbYesNo, "Msg_Heading")

If answer = vbYes Then

>VBA Command<
End If

    If answer = vbNo Then
>VBA Command<

End If
```

8. DELETE COLUMNS

```
Columns("A:C").Delete Shift:=xlToLeft
```

9. DELETE ROWS

```
Rows("4:7").Delete Shift:=xlUp
```

10. CLEAR DATA IN SELECTED CELLS

```
Range("A1:C5").Clearcontents
```

11. GO TO NEXT AVAILABLE ROW

```
Range("A" & Rows.COUNT).End(xlUp).Offset(1).Select
```

12. SELECT ALL CELLS IN ACTIVE RANGE

```
LR = Cells.Find("*", Cells(1, 1), xlFormulas, xlPart, xlByRows,
xlPrevious, False).Row

Range("A1:A" & LR).Select
```

13. SELECT ACTIVE CELL RANGE

```
Range("A1").Select
Range(Selection, Selection.End(xlToRight)).Select
Range(Selection, Selection.End(xlDown)).Select
```

14. WORKING WITH OBJECTS

```
With ActiveSheet
      .Shapes("Object_01").Visible = False
      .Shapes("Object_02").Visible = True
End With
```

15. HIDE & UNHIDE COLUMNS OR ROWS

```
Columns("D:F").EntireColumn.Hidden = True
Columns("C:G").EntireColumn.Hidden = False
Rows("10:15").EntireRow.Hidden = True
Rows("9:18").EntireRow.Hidden = False
```

16. PROTECT / UNPROTECT SHEET

```
Sheets("sheet_name").Protect
Sheets("sheet_name").Unprotect
```

17. PROTECT / UNPROTECT SHEET WITH PASSWORD

```
Sheets("sheet_name").Protect password:=”your password”
Sheets("sheet_name").Unprotect password:=”your password”
```

18. STOP / START SCREEN UPDATING

```
Application.ScreenUpdating = False
Application.ScreenUpdating = True
```

1. GET VALUE TO TARGET CELL BY SELECTING A LIST (copy to worksheet)

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)

Dim rangeToChange As Range

Set rangeToChange = Range("E1:E5")

If Not Intersect(Target, rangeToChange) Is Nothing Then

Range("A1").Value = Target.Value

End If
End Sub
```

2. SPELL NUMBERS

```
Option Explicit
'Main Function
Function SpellNumber(ByVal MyNumber)
    Dim Dollars, Cents, Temp
    Dim DecimalPlace, Count
    ReDim Place(9) As String
    Place(2) = "Thousand "
    Place(3) = "Million "
    Place(4) = "Billion "
    Place(5) = "Trillion "

    MyNumber = Trim(Str(MyNumber))
    DecimalPlace = InStr(MyNumber, ".")
    If DecimalPlace > 0 Then
        Cents = GetTens(Left(Mid(MyNumber, DecimalPlace + 1) & _
                "00", 2))
        MyNumber = Trim(Left(MyNumber, DecimalPlace - 1))
    End If
    Count = 1
    Do While MyNumber <> ""
        Temp = GetHundreds(Right(MyNumber, 3))
        If Temp <> "" Then Dollars = Temp & Place(Count) & Dollars
        If Len(MyNumber) > 3 Then
            MyNumber = Left(MyNumber, Len(MyNumber) - 3)
        Else
```

```
            MyNumber = ""
        End If
        Count = Count + 1
    Loop
    Select Case Dollars
        Case ""
            Dollars = "Zero "
        Case "One"
            Dollars = "One "
         Case Else
            Dollars = Dollars & ""
    End Select
    Select Case Cents
        Case ""
            Cents = "Only"
        Case "One"
            Cents = "and One Cent Only"
              Case Else
            Cents = "and " & Cents & "Cents Only"
    End Select
    SpellNumber = Dollars & Cents
End Function

Function GetHundreds(ByVal MyNumber)
    Dim Result As String
    If Val(MyNumber) = 0 Then Exit Function
    MyNumber = Right("000" & MyNumber, 3)
    ' Convert the hundreds place.
    If Mid(MyNumber, 1, 1) <> "0" Then
        Result = GetDigit(Mid(MyNumber, 1, 1)) & "Hundred "
    End If
    ' Convert the tens and ones place.
    If Mid(MyNumber, 2, 1) <> "0" Then
        Result = Result & GetTens(Mid(MyNumber, 2))
    Else
        Result = Result & GetDigit(Mid(MyNumber, 3))
    End If
    GetHundreds = Result
End Function

Function GetTens(TensText)
    Dim Result As String
    Result = "" ' Null out the temporary function value.
    If Val(Left(TensText, 1)) = 1 Then   ' If value between 10-19...
        Select Case Val(TensText)
            Case 10: Result = "Ten "
            Case 11: Result = "Eleven "
            Case 12: Result = "Twelve "
```

```
            Case 13: Result = "Thirteen "
            Case 14: Result = "Fourteen "
            Case 15: Result = "Fifteen "
            Case 16: Result = "Sixteen "
            Case 17: Result = "Seventeen "
            Case 18: Result = "Eighteen "
            Case 19: Result = "Nineteen "
            Case Else
        End Select
    Else ' If value between 20-99...
        Select Case Val(Left(TensText, 1))
            Case 2: Result = "Twenty "
            Case 3: Result = "Thirty "
            Case 4: Result = "Forty "
            Case 5: Result = "Fifty "
            Case 6: Result = "Sixty "
            Case 7: Result = "Seventy "
            Case 8: Result = "Eighty "
            Case 9: Result = "Ninety "
            Case Else
        End Select
        Result = Result & GetDigit _
            (Right(TensText, 1))  ' Retrieve ones place.
    End If
    GetTens = Result
End Function

Function GetDigit(Digit)
    Select Case Val(Digit)
        Case 1: GetDigit = "One "
        Case 2: GetDigit = "Two "
        Case 3: GetDigit = "Three "
        Case 4: GetDigit = "Four "
        Case 5: GetDigit = "Five "
        Case 6: GetDigit = "Six "
        Case 7: GetDigit = "Seven "
        Case 8: GetDigit = "Eight "
        Case 9: GetDigit = "Nine "
        Case Else: GetDigit = ""
    End Select
End Function
```

3. CREATE AUTOMATED BACKUP (Copy to This workbook)

```
Sub Workbook_BeforeClose(Cancel As Boolean)

ActiveWorkbook.Save

BKNO = 100

Dim strBakPath As String
  strBakPath = Application.ActiveWorkbook.Path & "\Backup Files\" &
Left$(ActiveWorkbook.Name, InStrRev(ActiveWorkbook.Name, ".") - 1) &
"\"
Dim fsoFSO
  Set fsoFSO = CreateObject("Scripting.FileSystemObject")

  If fsoFSO.FolderExists(Application.ActiveWorkbook.Path & "\Backup
Files\") Then
  GoTo subfold
  Else
  fsoFSO.Createfolder (Application.ActiveWorkbook.Path & "\Backup
Files\")
  End If

subfold:
  If fsoFSO.FolderExists(strBakPath) Then
  GoTo SaveFile
  Else
  fsoFSO.Createfolder (strBakPath)
  End If

SaveFile:

  On Error Resume Next
    Set objFiles = fsoFSO.GetFolder(strBakPath).Files
    If Err.Number <> 0 Then
        countfiles = 0
    Else
        countfiles = objFiles.Count

            If oldfile Is Nothing Then
            oldfile = fil
            End If
        If countfiles >= BKNO Then
        Do Until objFiles.Count = BKNO - 1
        For Each fil In objFiles
```

```
            If oldfile.Datecreated > fil.Datecreated Then Set oldfile
= fil

        Next fil
        fsoFSO.DeleteFile oldfile, True
        Set oldfile = Nothing

    Loop
        End If

    End If

' Save output backup file

  With ActiveWorkbook
    .SaveCopyAs strBakPath & Left$(ActiveWorkbook.Name,
InStrRev(ActiveWorkbook.Name, ".") - 1) & " (" & Format(Now, "dd-mm-yy
hhmm AM/PM") & ").xlsm"
    .Save
  End With
End Sub
```