USING VISUAL BASIC FOR APPLICATIONS (VBA)

Jake Blanchard
University of Wisconsin
Spring 2008
Macros allow one to add significant power to Excel

- They are small programs that can be called from a spreadsheet
- You can create functions or subroutines
- If you want to get fancy, you can add a user interface as well
Using Macros

- Macros are written in a Basic-like language called Visual Basic for Applications
- Excel comes with a separate macro editor
- To create a macro, go to Tools/Macro/Visual Basic Editor, then within the Editor go to Insert/Module
You should get this...
Creating a Function

- Suppose we want to create an Excel function that takes a temperature in Celsius and converts to Fahrenheit.
- We would type the following in a module:

```vbnet
Function ctof(temp)
    ctof = 9 / 5 * temp + 32
End Function
```
Function ctof(temp)
    ctof = 9 / 5 * temp + 32
End Function
Now go to the spreadsheet and type \( =\text{ctof}(100) \)

Or, you can put the value of “100” into cell A1 and then type \( =\text{ctof}(A1) \) into some other cell

In fact, this function can be used just as any built-in Excel function can be used
The Macro Language

- Operators: +, -, *, /, ^, Mod
- Comparison: =, <, >, <=, >=, <>
- Logical Operators: And, Eqv, Imp, Not, Or, Xor
- Intrinsic Functions: Abs, Cos, Sin, Tan, Atn (arc tangent), Exp, Log (natural), Sgn, Sqr (square root), Rnd (random number)
Flow Control

If condition Then
    statements
Else
    statements
End If

If x=0 Then
    f=1
Else
    f=sin(x)/x
End If
Flow Control

For counter=start To end statements
Next

For i=1 To 100
  sum=sum+i
Next
Flow Control

Do Until condition
  statements
Loop

i=1
x=1
Do Until i=50
  x=x*i
  i=i+1
Loop
Flow Control

Do While condition
  statements
Loop

i=1
x=1
Do While i<50
  x=x*i
  i=i+1
Loop
Practice

- Write an Excel function that calculates the sum of cubes of the first N integers
- Then write an Excel function that calculates the sum of cubes of the first N even integers
My solution

Function sumofcubes(N)
    ans = 0
    For i = 1 To N
        ans = ans + i ^ 3
    Next
    sumofcubes = ans
End Function
Another Solution

Function moresumofcubes(N)
    ans = 0
    i = 1
    Do Until i = N + 1
        ans = ans + i ^ 3
        i = i + 1
    Loop
    moresumofcubes = ans
End Function
Function sumofevencubes(N)
    ans = 0
    For i = 1 To 2 * N
        If (i Mod 2) = 0 Then
            ans = ans + i ^ 3
        End If
    Next
    sumofevencubes = ans
End Function
Creating a Subroutine

- Subroutines don’t return values...they carry out duties
- We’ll look at an example
Sub writeit()
    NumPoints = 21
    XNot = 0
    dX = 0.1
    ActiveCell.Value = "X"
    ActiveCell.Offset(0, 1).Value = "Sin(X)"
    x = XNot
    For i = 1 To NumPoints
        ActiveCell.Offset(i, 0).Value = x
        ActiveCell.Offset(i, 1).Value = Sin(x)
        x = x + dX
    Next
End Sub
Running the Macro

- Type the macro into a module in the Visual Basic Editor
- Return to a spreadsheet
- Create an active cell by clicking into some cell below which you don’t mind Excel writing some data
- Go to Tools/Macro/Macros, then click the name of the macro and click Run
Another Way to Run a Macro

- Go to View/Toolbars/Forms
- From this Toolbar, click on the button (row 2, column 2) and then trace out a button on a spreadsheet
- Assign the writeit macro to the button
- Now click the button
First 3 lines define constants
The next 2 lines write column labels back to the spreadsheet (ActiveCell is the highlighted cell in the spreadsheet)
The other lines step through the points, incrementing x and calculating sin(x)
The offset writes each result one row below the previous result
Summary and Conclusions

- VBA is Excel’s macro language
- Functions return values
- Subroutines carry out procedures