Civil Engineering Computation

VBA Fundamentals - Objects

"What you see with your eyes closed is what counts."

Lame Deer, Lakota Sage

VBA

- BASIC - Beginners All purpose Symbolic Instruction Code
- An interpreted language developed to be an introductory simple language to help people learn to program

VBA

- Compiled languages like FORTRAN and C are converted into executable code and that is what is run
- Interpreted languages execute from the original code each time they are run

VBA

- BASIC was the original language available on PCs
- Tended to be very slow and clumsy with lots of problems due to the simplicity of the language
- Fine for writing very simple things but almost useless for significant computation

VBA

- Wasn't adopted for much technical use
- Remained a "kid's" language
- Surpassed by PASCAL as a teaching language

VBA

- Multiple versions of BASIC were developed with the most popular being QuickBASIC or QBASIC
- Finally "grew up" in the 90's with tighter coding requirements and the introduction of objects into the language specifications.

VBA

- Really came into its own with the introduction of Visual Basic
- The development environment became visually oriented making the writing and debugging of code simpler than before
- Language requirements was tightened to make for better code

VBA

- Microsoft adapted Visual Basic as the language to develop code within their Office Suite (Word, EXCEL, and PowerPoint) and named the version VBA (Visual Basic for Applications)

VBA

- Visual Basic has now spread to be the language to write programs in ArcGIS, AutoCAD, and a number of web programs
- It is probably the most appropriate language for manipulation of data within applications that support it

VBA

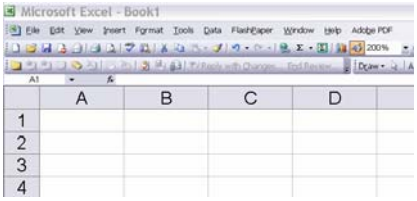
- It is not the most appropriate language for computational intensive applications
- That still remains as the domain of FORTRAN and C/C++

VBA

- Now its time to go under the hood of EXCEL and see what we are starting with
- Open EXCEL on the laptop

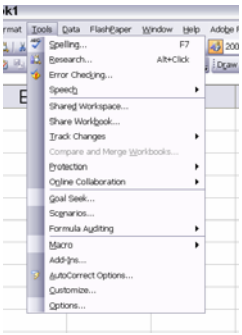
VBA in EXCEL

- You should have something that looks almost like this (my machine may be configured differently)



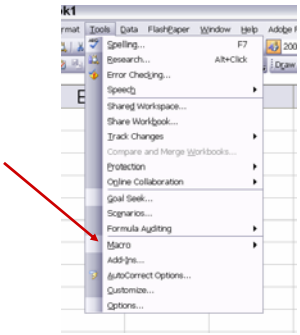
VBA in EXCEL

- One way to get under the hood is to go through the Tools Menu



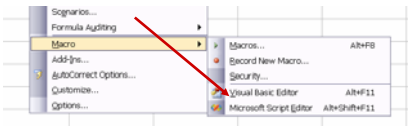
VBA in EXCEL

- Again, this may be a bit different on your version of EXCEL
- Notice the Macro option in the Tools menu



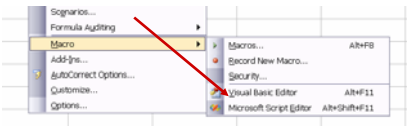
VBA in EXCEL

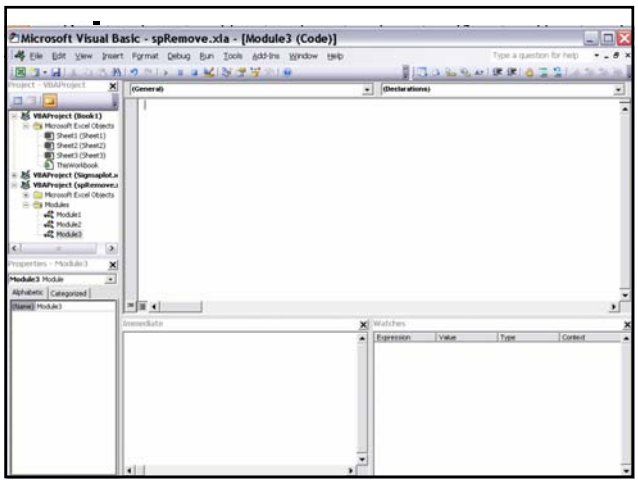
- An option under the Macro option is the Visual Basic Editor

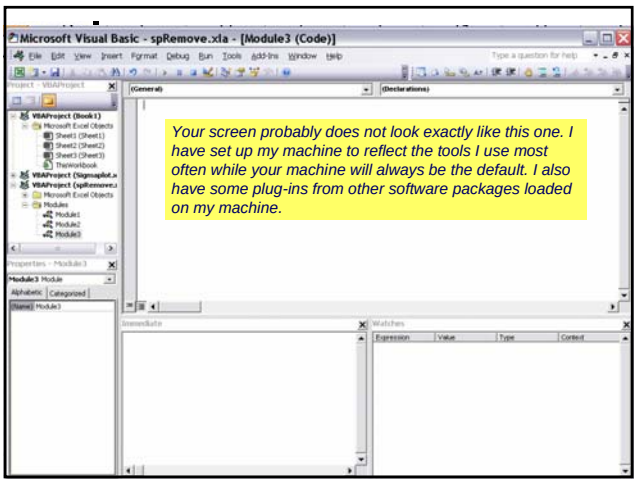


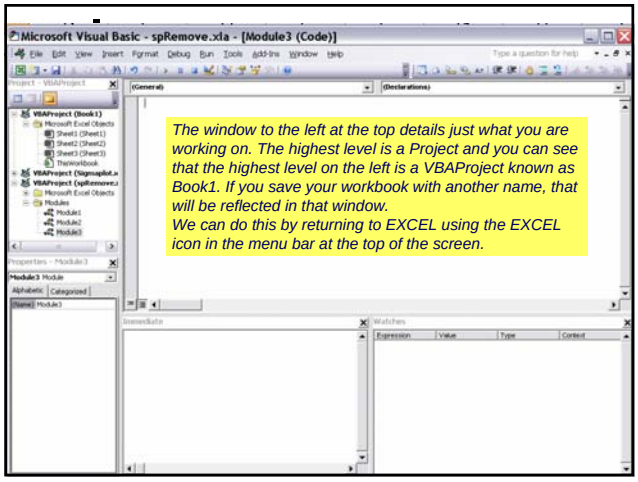
VBA in EXCEL

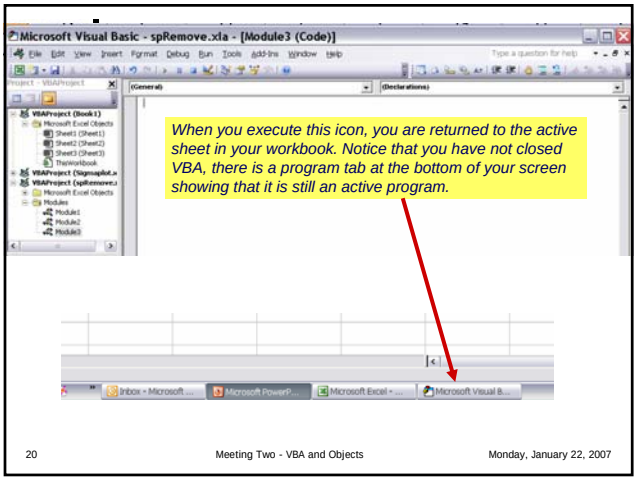
- When you select this Visual Basic Editor option, you open up the development environment for VBA programs in EXCEL




Microsoft Visual Basic - spRemove.xla - [Module3 (Code)]

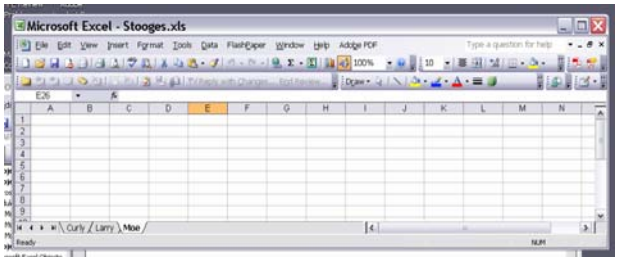

Microsoft Visual Basic - spRemove.xla - [Module3 (Code)]
Your screen probably does not look exactly like this one. I have set up my machine to reflect the tools I use most often while your machine will always be the default. I also have some plug-ins from other software packages loaded on my machine.


Microsoft Visual Basic - spRemove.xla - [Module3 (Code)]
The window to the left at the top details just what you are working on. The highest level is a Project and you can see that the highest level on the left is a VBAProject known as Book1. If you save your workbook with another name, that will be reflected in that window.
We can do this by returning to EXCEL using the EXCEL icon in the menu bar at the top of the screen.


Microsoft Visual Basic - spRemove.xla - [Module3 (Code)]
When you execute this icon, you are returned to the active sheet in your workbook. Notice that you have not closed VBA, there is a program tab at the bottom of your screen showing that it is still an active program.
20
Meeting Two - VBA and Objects
Monday, January 22, 2007

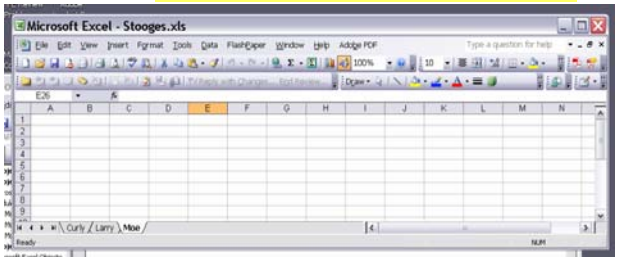
VBA in EXCEL

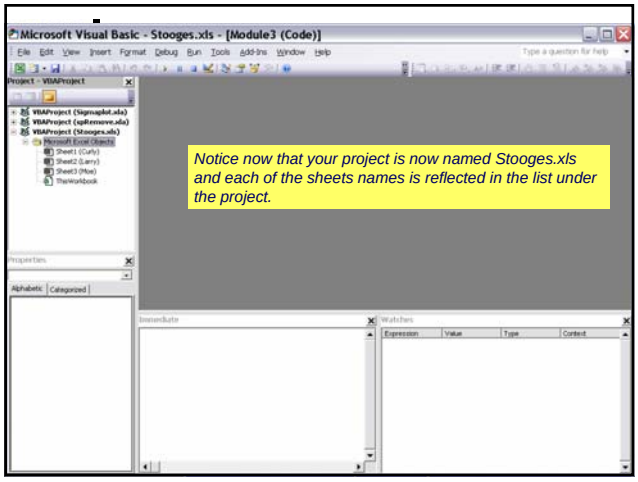
Now rename the sheets in your workbook Curly, Larry and Moe and save the workbook as Stooges.

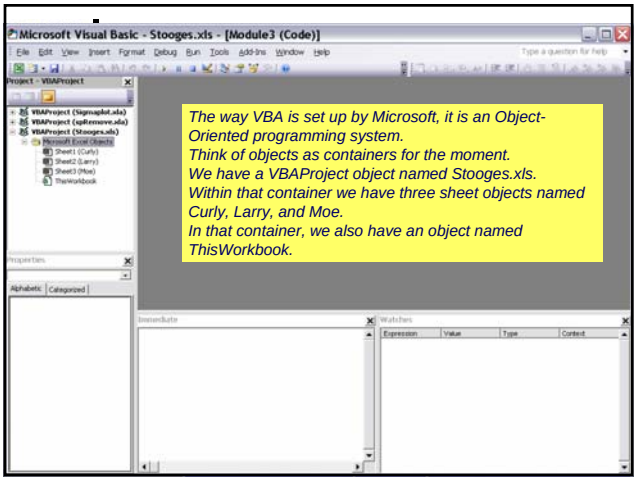


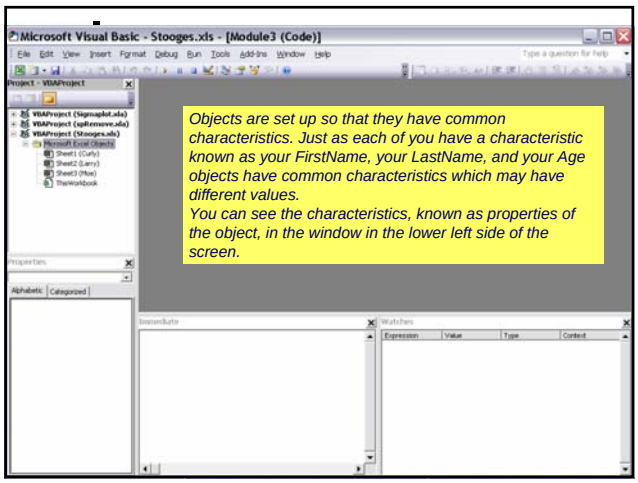
VBA in EXCEL

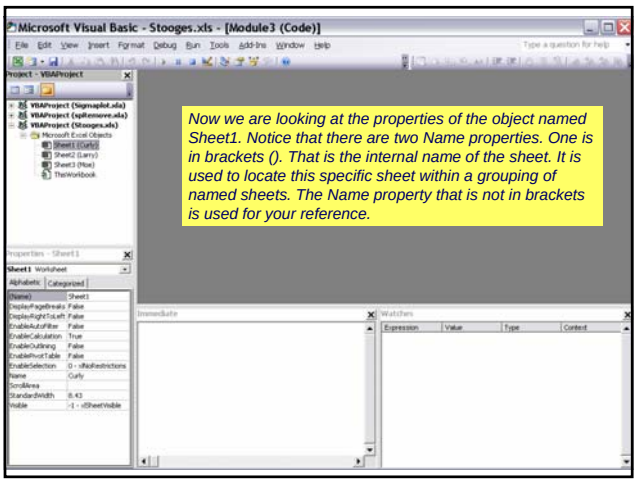
You can retrace the steps we took through the Tools menu to get back to VBA or you can use a keyboard shortcut ALT-F11.

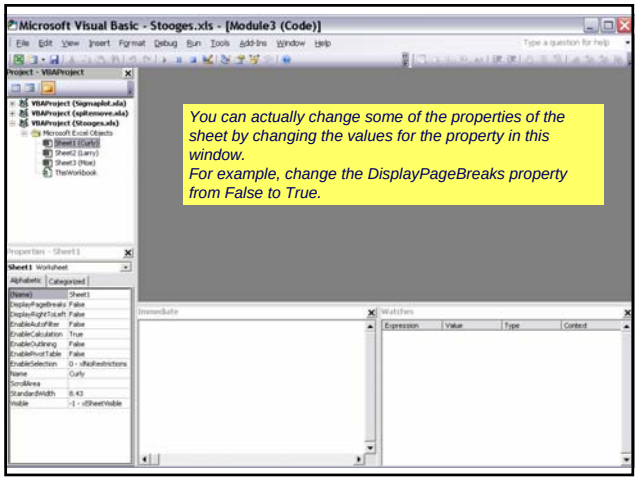


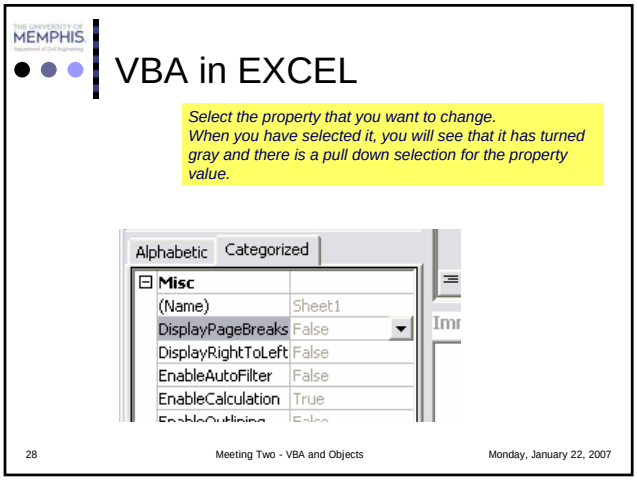





Microsoft Visual Basic - Stooges.xls - [Module3 (Code)]
Objects are set up so that they have common characteristics. Just as each of you have a characteristic known as your FirstName, your LastName, and your Age objects have common characteristics which may have different values.
You can see the characteristics, known as properties of the object, in the window in the lower left side of the screen.

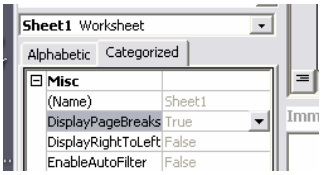

Microsoft Visual Basic - Stooges.xls - [Module3 (Code)]
Now we are looking at the properties of the object named Sheet1. Notice that there are two Name properties. One is in brackets (). That is the internal name of the sheet. It is used to locate this specific sheet within a grouping of named sheets. The Name property that is not in brackets is used for your reference.


Microsoft Visual Basic - Stooges.xls - [Module3 (Code)]
You can actually change some of the properties of the sheet by changing the values for the property in this window.
For example, change the DisplayPageBreaks property from False to True.


VBA in EXCEL
Select the property that you want to change.
When you have selected it, you will see that it has turned gray and there is a pull down selection for the property value.
Alphabetic
Categorized
Misc
(Name)
Sheet1
DisplayPageBreaks
False
DisplayRightToLeft
False
EnableAutoFilter
False
EnableCalculation
True
28
Meeting Two - VBA and Objects
Monday, January 22, 2007

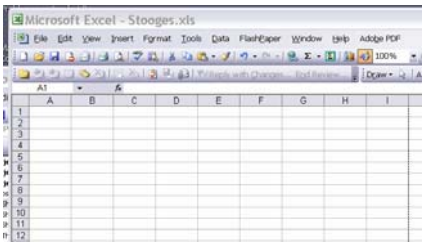
VBA in EXCEL

Select True from the options.
Now you have changed the property of this sheet in your workbook.
Go back to EXCEL and see just what is displayed.



VBA in EXCEL

Notice that there is now a dotted line showing where the page will break based on the printer you had set up.
If you check the other two pages in your workbook, you should not see there lines. You didn't change the properties on those two pages.

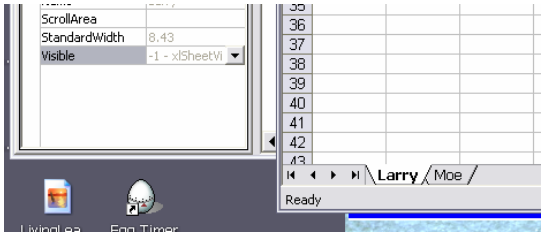


VBA in EXCEL

One interesting property is the Visible property.
You can hide a sheet completely from a user by setting this property correctly.

VBA in EXCEL

If you change this property to xlSheetHidden, this sheet will no longer show up on the tab list at the bottom of the workbook. It is effectively hidden from the user.



Objects in VBA

- The key element that makes VBA so powerful as a system is the use of objects.
- Objects are just containers but they allow the developer to generalize rather then have specific tools for each thing that comes along.

Objects in VBA

- I can divide the class into two types of objects
  - Women and Men
- Now I can build two types of restrooms, one for women and one for men rather than having to build an individual restroom for every person in the class.

Objects in VBA

- Anyone I had to send to the restroom, I would know which one to send them to by considering which object class they belonged to
- But this may be too limited

Objects in VBA

- What if I developed a more general object class called Student
- Each object in the class would have number of properties
  - FirstName
  - LastName
  - Age
  - Gender
  - CECredits

Objects in VBA

- I have a general template to use for each person in class (call an instantiation)
- If I wanted to consider be able to route you to the correct restroom, all I need to do is consider the Gender property
- If I wanted to estimate how many years you have until retirement, I would consider the Age property

Objects in VBA

- The Student object is just the label or the container that holds things
- It can hold different things just as my tags fit on all the students in the class.