**VBA research**

A programming language used for

o Most commonly automation and repetition of tasks in office applications

o Extension to user interaction. Ex: you might want to prompt users to take some particular action when they open, save, or print a document

o Interaction between office applications. Ex: moving data from excel to a set of PowerPoint slides

Bianca

### **Terms**

object/class = class is the template used to instantiate an object

### **Methods**

ActiveDocument

* allows to change the properties of the current object
* Application.ActiveDocument.Save
  + saves the active document
* Application.ActiveDocument.SaveAs ("New Document Name.docx")
  + saves as new file

### **Properties**

* there are properties in each object that you can manipulate

Application.ActiveSheet.Range("A1").Select

Application.Selection.Value = "Hello World"

* this changes all of the A1 cells to say hello world
* application is the object
* range refers to a select group of cells

### **Visual Basic Editor**

* developer tab is where you can code VBA and it must be enable  
  **To enable the Developer tab**
  1. On the **File** tab, choose **Options** to open the **Options** dialog box.
  2. Choose **Customize Ribbon** on the left side of the dialog box.
  3. Under **Choose commands from** on the left side of the dialog box, select **Popular Commands**.
  4. Under **Customize the Ribbon** on the right side of the dialog box, select **Main Tabs** in the drop down list box, and then select the **Developer** checkbox.
  5. Choose **OK**.
* now you should be able to see visual basic and macros tabs
* gotta use the .docm extension

### **Macros**

* to record a macro
  1. Open Excel to a new Workbook and choose the **Developer** tab in the ribbon. Choose **Record Macro** and accept all of the default settings in the **Record Macro** dialog box, including **Macro1** as the name of the macro and **This Workbook** as the location.
  2. Choose **OK** to begin recording the macro. Note how the button text changes to **Stop Recording**. Choose that button the instant you complete the actions that you want to record.
  3. Choose cell B1 and type the programmer's classic first string: Hello World. Stop typing and look at the **Stop Recording** button; it is grayed out because Excel is waiting for you to finish typing the value in the cell.
  4. Choose cell B2 to complete the action in cell B1, and then choose **Stop Recording**.
  5. Choose **Macros** on the **Developer** tab, select **Macro1** if it is not selected, and then choose **Edit** to view the code from Macro1 in the Visual Basic Editor.

### **Variables**

Dim MyStringVariable As String

MyStringVariable = "Wow!"

Worksheets(1).Range("A1").Value = MyStringVariable

* this is how variables look in VBA

### **Branching and Looping**

Sub Macro1()

If Worksheets(1).Range("A1").Value = "Yes!" Then

Dim i As Integer

For i = 2 To 10

Worksheets(1).Range("A" & i).Value = "OK! " & i

Next i

Else

MsgBox "Put Yes! in cell A1"

End If

End Sub

Ali

**VBA RESEARCH**

* Office Visual Basic for application is an event driven programming language that allows you to extend your office application.
* The great power of VBA programming in Office is that nearly every operation that you can perform with a mouse, keyboard, or a dialog box can also be automated by using VBA.
* The most common use of VBA was to automate mundane tasks in Office applications
* In addition to the automation of everyday tasks, you can also add new functionality to your regular office applications. For example, you could write some VBA code that displays a pop up message that reminds users to save a document to a particular network drive the first time they try to save it.

Rachael

* Long but useful video on macros and VBA (why we might use VBA, examples of how to use VBA, coding macro functions using VBA): <https://www.youtube.com/watch?v=KPh13bDRIkM>