VBA PROGRAMMING FOR EXCEL

FREDRIC B. GLUCK

FGLUCK@MADISONCOLLEGE.EDU
FBGLUCK@GMAIL.COM
608-698-6304
Class Website for Excel Resources and downloadable files is at: learn.fbgluck.com
Syllabus-lite

● Very quick review and check in of your experience with Excel (low, medium, high) and with programming in other languages
● Guidelines of the class
● Expectations
  ● Attendance at all classes
  ● Cell phones
  ● Homework / Labs / Exams / Projects
About You…

- Your Excel experience
- Have you ever done any programming (in any computer language?)
- How do you use Excel today?
- How would you like to be able to use Excel after completing this class?
What We Will Learn

- This is a class that will teach you computer programming as it applies to the Microsoft Excel environment.
- We start with simple examples and work up to writing complete Visual Basic programs using control structures to accomplish programming tasks.
# The Excel Programming Environment

<table>
<thead>
<tr>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System (MS Windows)</td>
</tr>
<tr>
<td>Programs</td>
</tr>
<tr>
<td>MS Excel</td>
</tr>
</tbody>
</table>

## Hardware

- **Operating System (MS Windows)**

## Programs

- **MS Excel**

## Excel Visual Basic for Applications (VBA)

- **Excel VBA Projects**
  - **Excel VBA Module**
  - **Excel VBA Object**

## Excel Functions

## Excel Macros

## Excel VBA Objects

### Excel VBA Tools

- **Excel VBA Object**
- **Excel VBA Module**
VBA (Visual Basic for Applications)

- Excel macros, functions and programs are written in the VBA programming language
  - VBA can be used for ALL Microsoft Office applications
    - Some of what we learn here (Excel) can be used in other office apps
  - VBA lineage is from BASIC
    - added MS office specific functions
    - added object-oriented programming
Getting Things Automated in Excel

- User entered keyboard commands
  - Recorded commands (MACROS)
    - Add: program flow control (logic)
    - Excel Custom Functions (extend commands and can be used in Macros)
      - Add: additional statements and multiple functions
      - VBA programs

Level of automation vs. Level of complexity
Enabling Macros and VBA

- Developer options are “hidden” from the everyday user
  - FILE > OPTIONS > CUSTOMIZE RIBBON
  - Check “DEVELOPER” BOX to turn on Developers Tab
Relative vs Absolute Referencing (Recording)

- Absolute Referencing:
  - Excel records the absolute value of the cells.
  - Example: Excel records you selected Cell “E4”

- Relative Referencing:
  - Excel records you moved “down three cells from the current cell”

Question: Which should you use?
Macro Security

- Macros can carry malicious code
- The default security settings are “do not run macros”
- Protection Schemes...
  - **Create Trusted Locations** (folders) to store workbooks:
    - DEVELOPER > MACRO SECURITY
  - **Adjust Global Security Settings** for Macros:
    DEVELOPER > MACRO SECURITY > MACRO SETTINGS
- Macro Enabled Workbooks - SAVE AS ....
Excel File Types

- **.xlsx**: An Excel file in XML form. Cannot store Macros
- **.xlsm**: An Excel file that can store macros
- **.xlsb**: An Excel file in binary format. Designed for very large spreadsheets (1.1 million rows)
- **.xls**: An Excel file that can be read by versions prior to 2010. Can contain macros

You can customize Excel to always produce xlsm files.
- FILE > EXCEL OPTIONS > SAVE > SAVE FILES IN THIS FORMAT
Macro Security

- Workbooks in trusted areas can always run macros regardless of other settings
- Macro settings control macro access for workbooks outside of trusted locations
  - The “Enable All Macros” is not recommended
  - “Disable All Macros With Notification” is recommended
Terminology

- Macro
- Visual Basic
- Module
What is a “Macro”

- Recorded keystrokes that can be played back to automate (repeat) a task.
- Macros have no flow control.
  - Keystrokes are always played in the same order
- Examples where Macros are useful:
  - Reformatting data by removing spaces or replacing a string of characters.
  - To repeat “find and replace” commands
  - To run “document setup” actions (e.g. insert a header, footer or other document id information)
- Macro Code is "written" by recording keystrokes
Modules

- A container for storing code (Macros)
- Attached to (part of) an Excel Workbook
The Macro Recorder
How to Create A Macro (VBA SubRoutine)

Macros are “written” by recording the series of keystrokes that are used to accomplish a task.

Example:

- Turn on Macro Recorder
- Step 1
- Step 2
- Step “n”
- Turn off Macro Recorder
- Sub (macro_name)
  ... 
  End Sub
Create Macros with the Macro Recorder

- The Macro Recorder is available only in MS Word and MS Excel
- From the DEVELOPER TAB, choose RECORD MACRO
- Specify the properties of theMacro:
  - The **name** of the macro (No spaces. Begins with letter or underscore. Must be unique to the workbook)
  - A **shortcut key** assigned to the macro
  - Where to **store** the macro
  - A **description** of the macro
Recording Macros

- After the dialog box, recording begins
- **EVERYTHING** is recorded (keyboard, menu selections)
  - Mouse actions are recorded in EXCEL, not in WORD
- Icon at bottom indicates “Recording is on”
- To stop recording:
  - VIEW > MACROS > STOP RECORDING
  - Click on the MACRO icon
Quick Ways to Run Macros

- Short keys are case sensitive
  - “Ctrl+e” is not the same as “Ctl+E”
  - If you use “E”, you need to type “CTL+Shift+E”
- Open (non-assigned) lower case keys are:
  - e,j,m and q
- Assigned (Upper case) keys are:
  - A, F, O and P
Quick Ways to Run Macros

- Macros can be assigned to the Quick Access Toolbar
- Choose the “customize Quick Access Toolbar command” (Dropdown from the quick access toolbar) > MORE COMMANDS >
- From the CHOOSE COMMANDS FROM LIST choose MACROS
- Choose the symbol and complete the display name
Quick Ways to Run Macros

- You can also assign a Macro to a Form Control, Text Box or Shape
  - DEVELOPER > INSERT > Form Control Button
  - Click the top left location of the button
  - Assign the macro
  - Modify the button (text, size etc.)
Where Do I Store Macros?

- Two Options: (specified when you define a macro)
  - In your current workbook
    - Travels with the workbook
    - Only available when the workbook is open
  - In the “Personal Macro Workbook”
    - When you record a Macro -- Store Macro In: Personal Macro Workbook
    - Creates a “hidden workbook” on your machine called “PERSONAL.XLSB”. This workbook is stored in:
      - C:\Users\<username>\AppData\Roaming\Microsoft\Excel\XLSTART
      - Does not exist until you save a macro into this workbook
      - Can be unhidden: VIEW > UNHIDE > PERSONAL
Changing Macro Properties

- Macro Properties
  - Macro Name
  - Shortcut Key
  - Description

- Change these properties by:
  - VIEW > MACROS > (select macro)

- WARNING: Avoid assigning shortcut key that is used by Excel (e.g. ctl+b, ctl+c etc.)
  - Excel will override it’s internal command and run your macro in case of a conflict (see pg 11 in the book for additional information)
Exercises
Exercise #1

- Create a Macro that fills in cells “A2” to “A11” with the values “1” through “10”.
- Store this Macro in the spreadsheet you are working on.
- Assign a shortcut key and an icon on the Quick Access Toolbar to this macro.
- Assign a button on the worksheet called "Count 1 to 10" to this macro.
- Save the worksheet and re-open it. Make sure the macro runs correctly.
Exercise #2

- Create a macro that
  - clears the contents of the workbook
  - moves the cursor to cell A1
  - fills the cells from A1 to A7 with the days of the week
  - Fills the cells from B1 to B7 with the day numbers (assume that Monday is day #1)
- Save the macro to your Personal Macro Workbook
- Make sure to unhide this spreadsheet
- Assign a shortcut key and Quick Access Icon to this macro
- Save and close the worksheet
- Open a new worksheet and run the macro you just defined
Exercise #3

- Create a “setup macro” that you run immediately after you open a new spreadsheet
  - Adds your company name to the workbook header
  - Adds a worksheet
  - Renames the worksheets to “North” “East” “West” and “South”
  - Adds a “page # of # pages” and file name and “prepared by your name” to each section of the footer.
- Save the macro (think of where you should save it)
- Close the spreadsheet
- Open a new spreadsheet, run your setup macro and run the macro you defined in Exercise #2
Exercise #4 – Data Reformat

- Once a week, you get a report of delinquent accounts from the accounting legacy system. You want to use the data to send letters via MS WORD.
- Open the csv file “accounting_data.csv” from the MATC "cloud"
  - i:\FB_Gluck\accounting.csv
- Create a macro that reformats the names, address1 and address2 so that each of the elements is in one column.
  - Hint: running the macro once will take one "set" of data and reformat it. You will have to run the macro once for each individual set of data.
- If you want, try to modify the macro so that you can run it once and reformat the entire sheet.