FIT1013 - Week 4 Resources

Developing an Excel Application

Week 4 Resources

1. Objectives 2

2. Create an Excel application 2

3. Use defined names in formulas 2

4. Create validation rules for data entry 3

5. Working with Macros 8

6. Practice and Apply 14

Reference:

Microsoft Excel 2016, New Perspectives Series, Parsons, Oja, Carey, Desjardins Comprehensive Edn., Cengage Learning, **Module 7**

1. Objectives

- Create an Excel application
- Use defined names in formulas
- Create validation rules for data entry
- Learn about macro viruses and Excel security features
- Create and run a macro
- Edit a macro using the Visual Basic Editor
- Assign a macro to a keyboard shortcut and a button
- Save and open a workbook in macro-enabled format

2. Create an Excel application

It is extremely important to plan your application beforehand and avoid development 'on the fly'. Simply creating code as you go along not only makes it hard for others to understand your work, it may not necessarily be what your client requires. During your course, you will learn more about the Systems Development process and the importance of planning but in this unit, we will focus on the features of the business application tools.

Recall from Lecture:

- You can use a **defined name** to assign a meaningful, descriptive name to a cell or range
- A **defined name** enables you to quickly navigate within a workbook to the cell or range with the defined name
- You can use defined names to create more descriptive formulas

3. Use defined names in formulas

Sometimes it is useful to choose a cell or range of cells and assign a name to those cells. This is called a range name and allows you to refer to the cell or a range of cells by their name instead of their cell references. For example, we might assign a range name, Expenses, to a group of cells that represent their expenses. Then when you want to calculate with those cells, you simple put their Range Name in the formulas instead of the cell reference. In addition, you can also, add defined names to existing formulas.

See Ex 403 and Ex 404 of textbook on examples for using defined names in Formulas.

4. Create validation rules for data entry

Excel provides features that will help validate data as it is entered into a worksheet. You can specify the type of data that is allowed and/or a range of acceptable values. If a value is entered that does not meet the requirements, an error message is displayed. Setting a rule is a preventative measure, allowing you to validate data upon entry. You can set up various rules relating to data entry as well as providing an input message to aid the user in entering data.

Table below shows the type of data and the range of acceptable values (called **validation criteria)**

Type	Acceptable Values
Any value	Any number, text, or date; removes any existing data validation
Whole Number	Integers only; you can specify the range of acceptable integers
Decimal	Any type of number; you can specify the range of acceptable numbers
List	Any value in a range or entered in the Data validation dialog box separated by commas
Date	Dates only; you can specify the range of acceptable dates
Time	Times only; you can specify the range of acceptable times
Text Length	Text limited to a specified number of characters
Custom	Values based on the results of a logical formula

Summary of Data Entry Validation:

- Creating an Error Alert Style and Message
  - An error alert determines what happens after a user tries to make an invalid entry in a cell that has a validation rule defined
  - The three error alert styles are:
    - **Stop**: Prevents the entry from being stored in the cell
    - **Warning**: Prevents the entry from being stored in the cell unless the user overrides the rejection and decides to continue using the data
    - **Information**: Accepts the data value entered, but allows the user to choose to cancel the data entry
- Creating an **Input Message**
  - One way to reduce the chance of a data-entry error is to display an input message when a user makes the cell active
  - An input message provides additional information about the type of data allowed for that cell
  - Input messages appear as ScreenTips next to the cell when the cell is selected
  - Can add an input message to a cell even if you don't set up a rule to validate the data in that cell
- Creating a **List Validation Rule**
  - Use the data validation feature to restrict a cell to accept only entries that are on a list you create
  - Create the list of valid entries in the Data Validation dialog box or use a list of valid entries in a single column or row

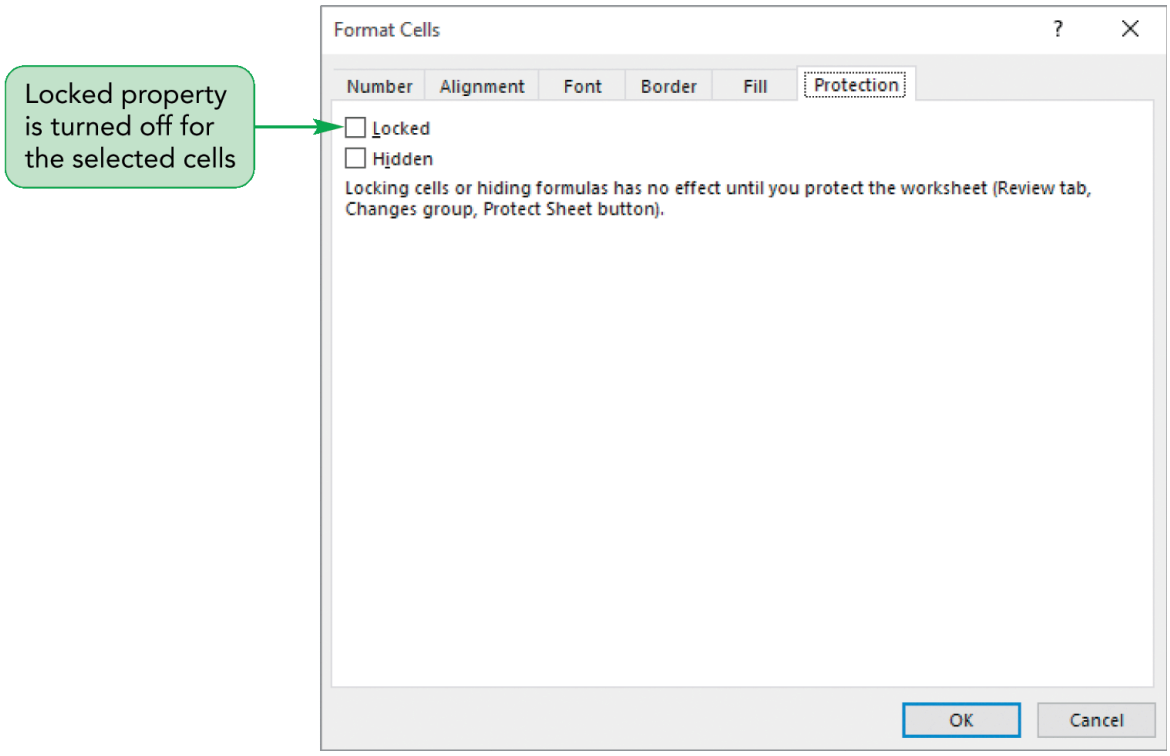
– Once you create a list validation rule for a cell, a list box with the possible values appears

Types of data entry messages available

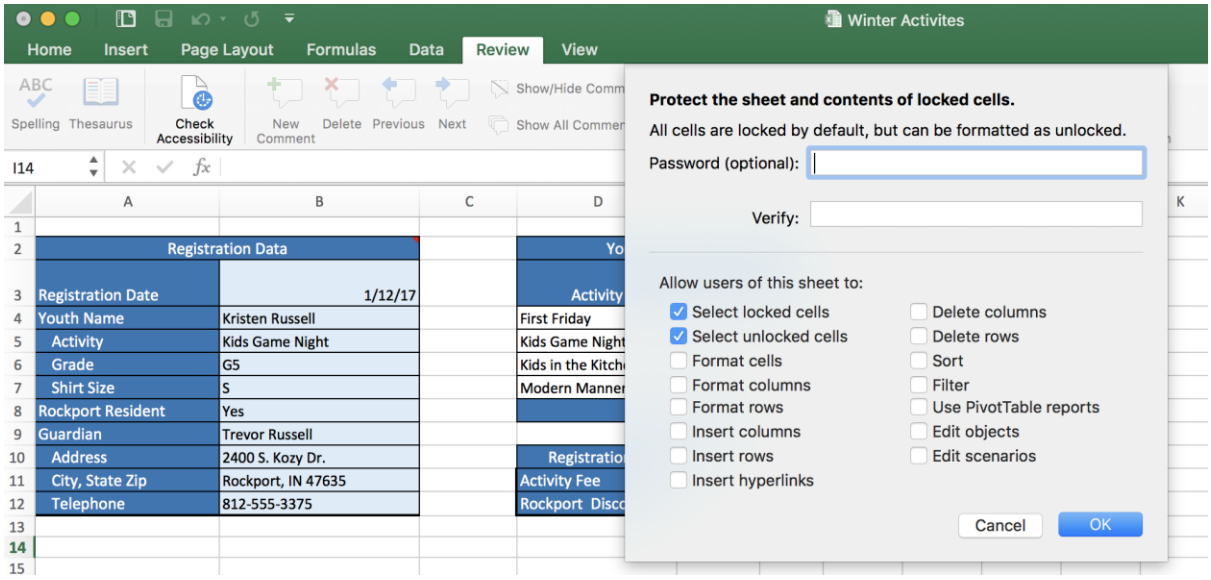
Icon	Type of Alert	Label on Button	Action If Button Clicked
	Warning	Continue Yes	value entered in cell; processing continues
		Continue No	value entered in cell; Excel stops, waiting for you to enter another value
		Cancel	value not entered in cell
	Information	OK	value entered in cell
		Cancel	value not entered in cell
	Stop	Retry	
		Cancel	

Protecting a Worksheet and a Workbook

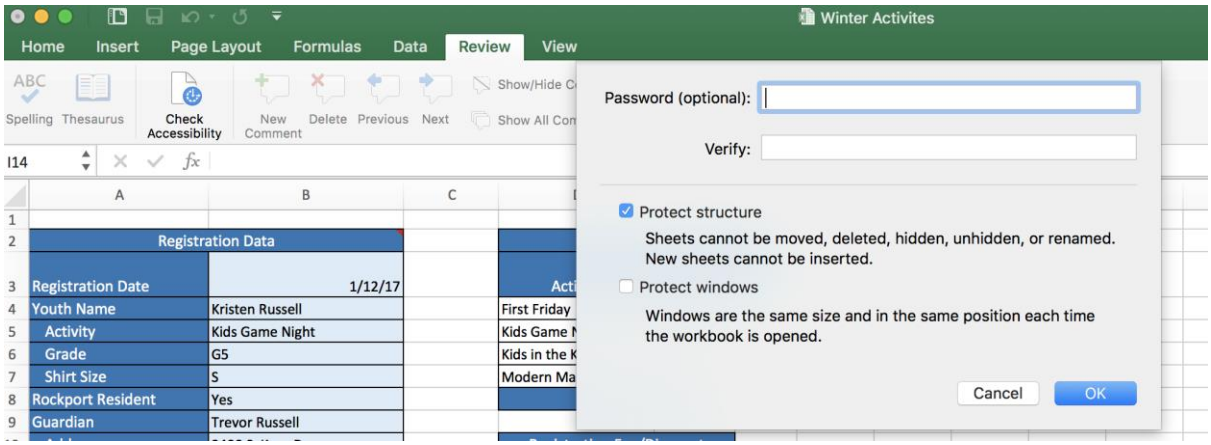
- Another way to minimize data-entry errors is to limit access to certain parts of the workbook
- **Worksheet protection** prevents users from changing cell contents, such as editing formulas in a worksheet
- **Workbook protection** also prevents users from changing the workbook's organization, such as inserting or deleting worksheets in the workbook
- You can keep users from viewing the formulas used in the workbook
- Locking and Unlocking Cells
  - Every cell in a workbook has a **locked property** that determines whether changes can be made to that cell
  - The locked property has no impact as long as the worksheet is unprotected
  - After you protect a worksheet, the locked property controls whether the cell can be edited
  - Unlock a cell by turning off the locked property
  - By **default**, the locked property is turned **on** for each cell, and worksheet protection is turned **off**



- Protecting a **Worksheet**
  - When you set up worksheet protection, you specify which actions are still available to users in the protected worksheet
  - You can limit the user to selecting only unlocked cells, or allow the user to select any cell in the worksheet; these choices remain active as long as the worksheet is protected
  - A protected worksheet can always be unprotected
  - You can add a password that must be entered to turn off the protect



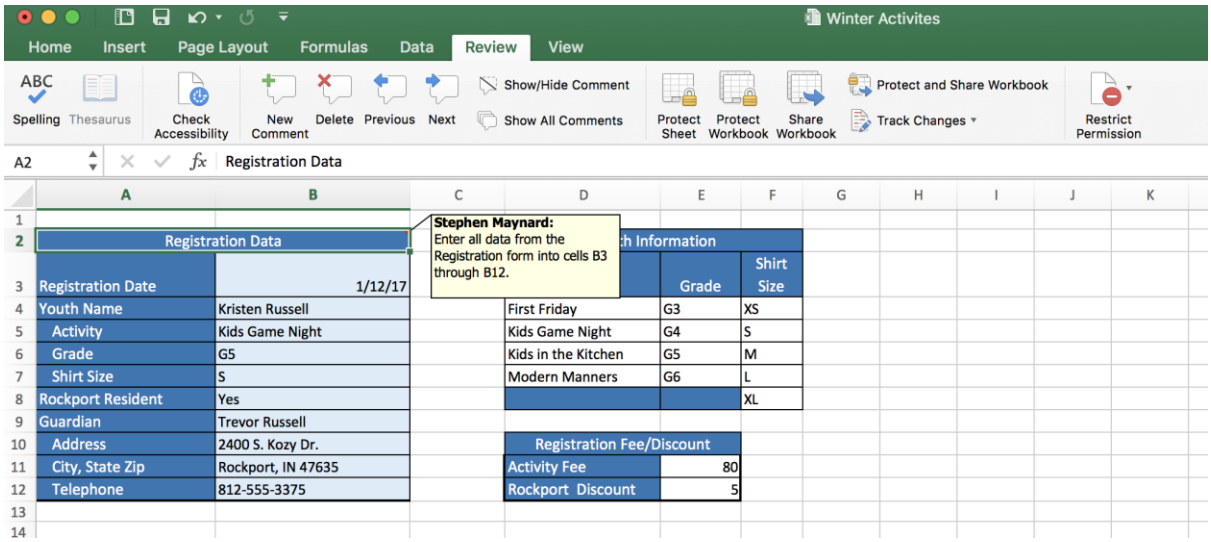
- Protecting a **Workbook**
  - Worksheet protection applies only to the contents of a worksheet, not to the worksheet itself
  - To keep a worksheet from being modified, you need to protect the workbook
  - You can protect both the structure and the windows of a workbook
  - Protecting the structure prevents users from renaming, deleting, hiding, or inserting worksheets



- **Unprotecting** a Worksheet and a Workbook
  - You can turn off worksheet protection at any time
  - You must unprotect a worksheet to edit its contents
  - You can unprotect the workbook
  - If you need to insert a new worksheet or rename an existing worksheet, you can unprotect the protected workbook, make the changes to the structure, and then reapply workbook protection

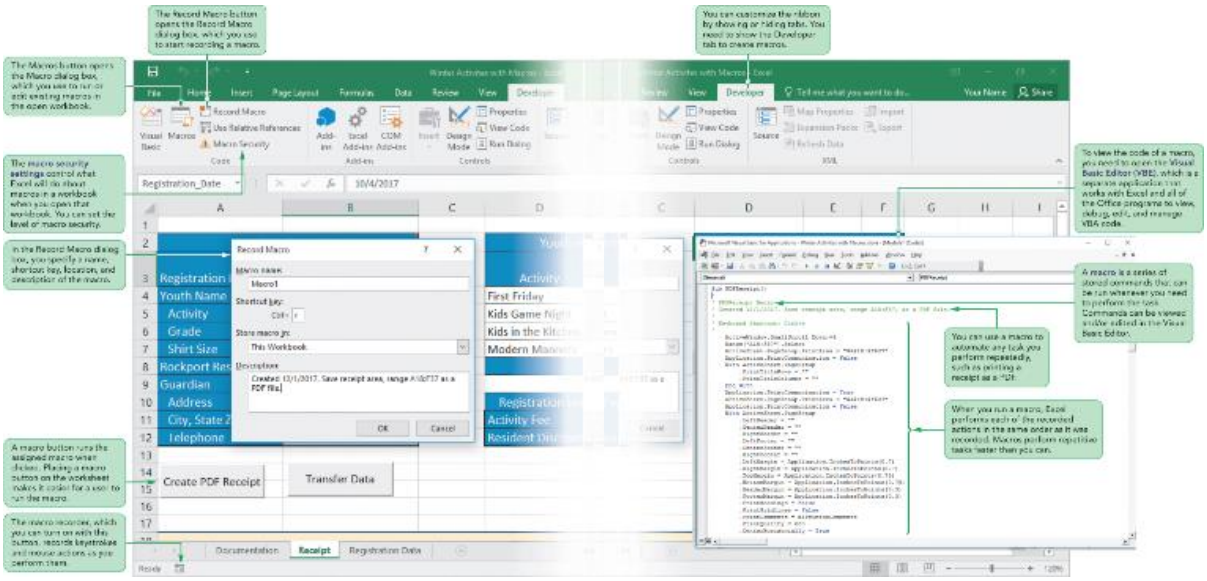
Inserting Comments

- Comments are often used in workbooks to:
  - Explain the contents of a particular cell, such as a complex formula
  - Provide instructions to users
  - Share ideas and notes from several users collaborating on a project
- If you collaborate on a workbook, the top of the comment boxes would show the name of each user who created that comment
- A small red triangle appears in the upper-right corner of a cell with a comment
- The comment box appears when you point to a cell with a comment



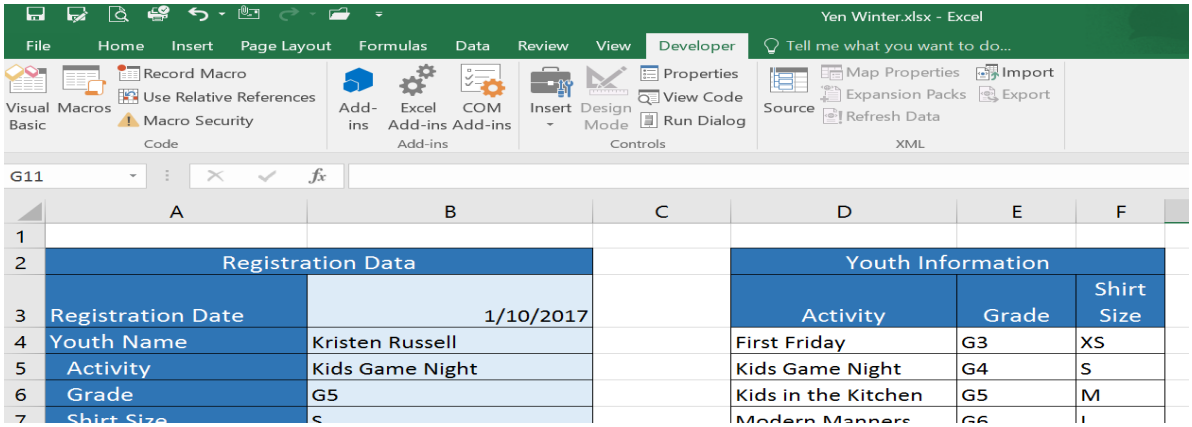
5. Working with Macros

Visual Overview: Working with Macros



Automating Tasks with Macros

- Macros automate any task you perform repeatedly
- Macros perform repetitive tasks consistently and faster than you can.
- After the macro is created and tested, you can be assured the tasks are done exactly the same way each time
- To create and run macros, you need to use the DEVELOPER tab
- The DEVELOPER tab has the following groups:
  - One for code
  - One for add-ins
  - One for controls
  - One for XML



Protecting Against Macro Viruses

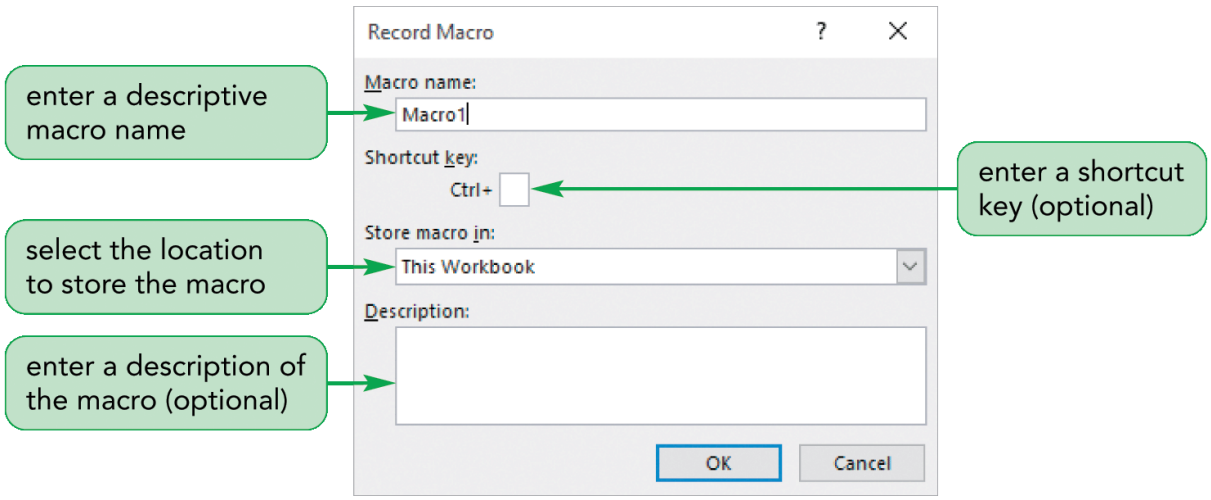
- A **virus** is a computer program designed to copy itself into other programs with the intention of causing mischief or harm
- **Macro viruses** use a program's own macro programming language to distribute the virus
  - Can be destructive
  - Can modify or delete files that may not be recoverable
- Macro Security Settings
  - Control what Excel will do about macros in a workbook when you open that workbook

Setting	Description
Disable all macros without notification	All macros in all workbooks are disabled and no security alerts about macros are displayed. Use this setting if you don't want macros to run.
Disable all macros with notification	All macros in all workbooks are disabled, but security alerts appear when the workbook contains a macro. Use this default setting to choose on a case-by-case basis whether to run a macro.
Disable all macros except digitally signed macros	The same as the "Disable all macros with notification" setting except any macro signed by a trusted publisher runs if you have already trusted the publisher. Otherwise, security alerts appear when a workbook contains a macro.
Enable all macros	All macros in all workbooks run. Use this setting temporarily in such cases as when developing an application that contains macros. This setting is not recommended for regular use.

- Macro Security Settings (continued)
  - Set macro security in the Trust Center
  - The **Trust Center** is a central location for all of the security settings in Office
  - By default, all potentially dangerous content is blocked without warning
  - If content is blocked, the Message Bar (also called the trust bar) opens below the ribbon, notifying you that some content was disabled
  - You can click the Message Bar to enable content

Recording a Macro

- You can create an Excel macro in one of two ways:
  - Use the macro recorder to record keystrokes and mouse actions as you perform them
  - Enter a series of commands in the **Visual Basic for Applications (VBA)** programming language
- The macro recorder can record only those actions you perform with the keyboard or mouse
- The macro recorder is a good choice for creating simple macros
- For more sophisticated macros, you might need to write VBA code directly in the Visual Basic Editor (VBE)
- By default, the macro is stored in the current workbook (only available with that workbook)
- Another option is to store the macro in the **Personal Macro workbook**, a hidden workbook named PERSONAL.xlsb that opens whenever you start Excel
  - Stores commonly used macros
  - Is most convenient for users on stand-alone computers
- Can store the macro in a new workbook

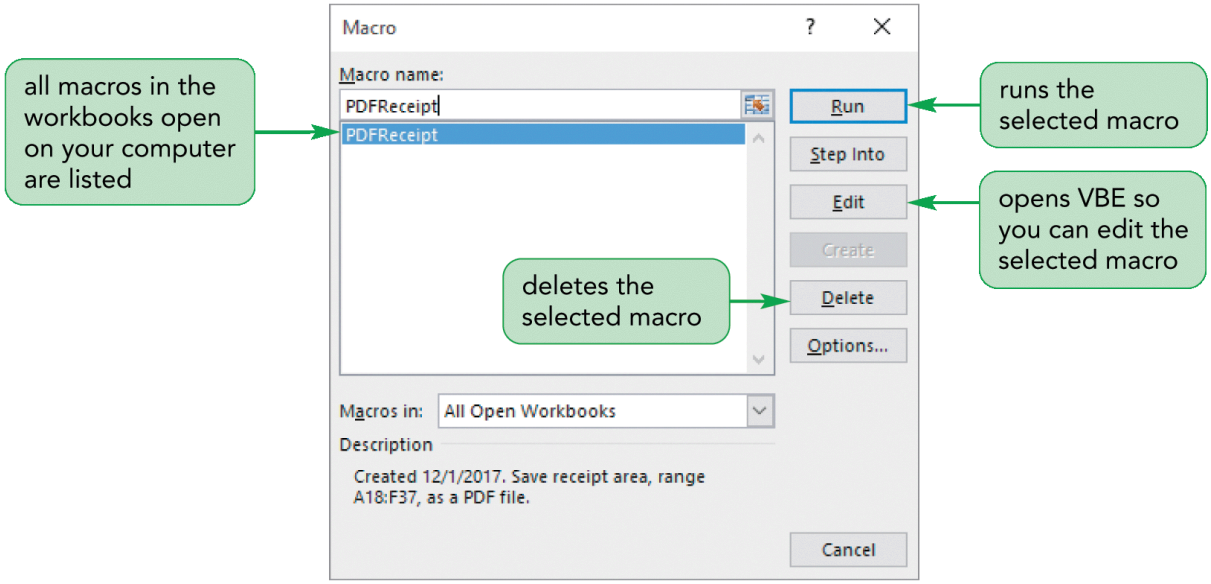


Recording a Macro: https://www.youtube.com/watch?v=MUeL0nHEqkI

Running a Macro

- After you record a macro, you should run it to test whether it works as intended
- Running a macro means Excel performs each of the steps in the same order as when it was recorded
- To run the macro you created, you can either use the shortcut key you specified or select the macro in the Macro dialog box
- The Macro dialog box lists all of the macros in the open workbooks
- From this dialog box, you can:
  - Select and run a macro

- Edit the macro with VBA
- Run the macro one step at a time so you can determine in which step an error occurs
- Delete the macro

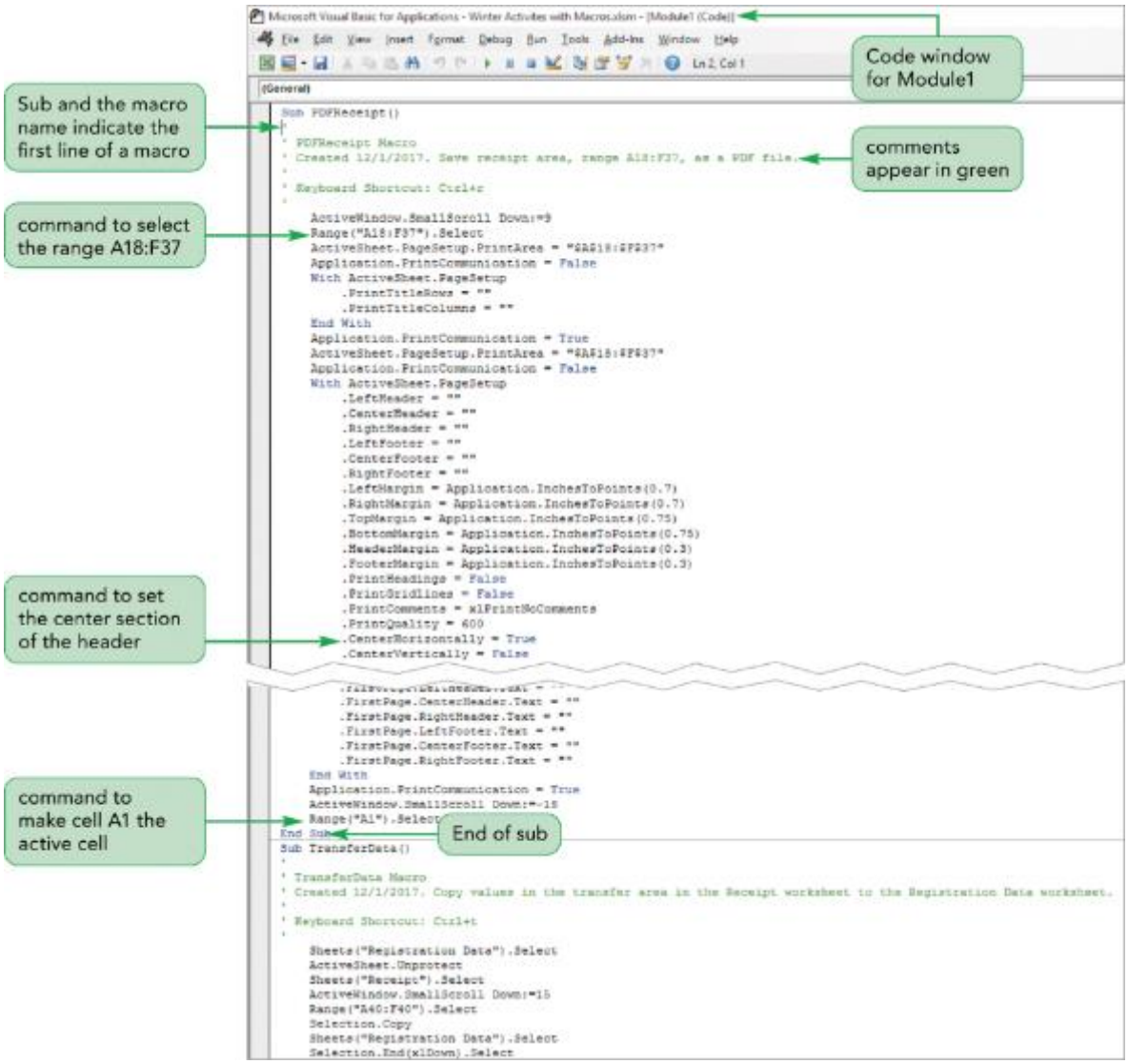


Creating the TransferData Macro

- Fixing Macro Errors
  - If a macro does not work correctly, you can fix it
  - You have the following options to fix a macro:
    - Rerecord the macro using the same macro name
    - Delete the recorded macro, and then record the macro again
    - Run the macro one step at a time to locate the problem, and then use one of the previous methods to correct the problem

Working with the Visual Basic Editor

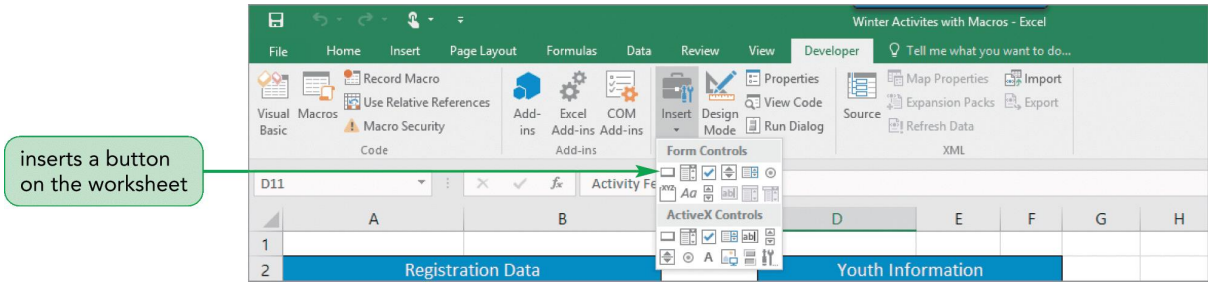
- To view the code of a macro, open the Visual Basic Editor (VBE) - allows you to view, debug, edit, and manage VBA code
- The VBE consists of several components, including:
- The Code window that contains the VBA code
- A menu bar with menus of commands you use to edit, debug, and run VBA statements
- You can access the Visual Basic Editor through the Macro dialog box or the Visual Basic button in the Code group on the DEVELOPER tab



- Understanding the Structure of Macros
  - In VBA, macros are called **sub procedures**
  - Each sub procedure begins with the keyword Sub followed by the name of the sub procedure and a set of parentheses
  - Sub procedures are organized into **modules**
- Editing a Macro Using the Visual Basic Editor
  - The Visual Basic Editor provides tools to assist you in writing error-free code
  - As you type a command, the editor will provide pop-up windows and text to help you insert the correct code

Creating Macro Buttons

- Another way to run a macro is to assign it to a button placed directly in the worksheet
- Macro buttons are often a better way to run macros than shortcut keys
- Clicking a button (with a descriptive label) is often more intuitive and simpler for users than trying to remember different combinations of keystrokes



Saving a Workbook with Macros

- When you save a workbook that contains macros, a dialog box opens indicating that the workbook you are Saving a Workbook with Macros
- trying to save contains features that cannot be saved in a macro-free workbook
- The default Excel workbook does not allow macros to be stored as part of the file and has the .xlsx file extension
- A macro-enabled workbook has the **.xlsm file extension**
- In the Macro warning dialog box:
  - Click the Yes button if you want to save the workbook without the macros; the macros you created will be lost
  - Click the No button if you want to save the workbook with the macros; then save the workbook as a new file—one that allows macros to be saved as part of the file

Opening a Workbook with Macros

- When you open a file with macros, Excel checks the opening workbook to see if it contains any macros
- When the workbook opens the first time, a SECURITY WARNING appears in the Message Bar with an option to enable or disable macros

Removing a Tab from the Ribbon

- If you decide you don't want a tab displayed on the ribbon, you can remove it
- To remove the DEVELOPER tab from the ribbon:
  - Right-click any tab on the ribbon, and then click Customize the Ribbon
  - In the Main Tabs box, click the Developer check box to remove the checkmark
  - Click the OK button

6. Practice and Apply

- Understanding how to create an Excel application
- Understanding how to use defined names in formulas
- Understanding how to create validation rules for data entry
- Familiar with macro viruses and Excel security features
- Understanding how to create and run a macro
- Understanding how to edit a macro using the Visual Basic Editor
- Understanding how to assign a macro to a keyboard shortcut and a button
- Understanding how to save and open a workbook in macro-enabled format
- Complete Tutorial 4 Exercises