CP212 Assignment 2

Weight: 5.1%

Marks: 16

Due: Saturday, February 9, 11:45pm

Read the Assignment Guidelines: http://bohr.wlu.ca/rhenderson/cp212/assignments.html

The file **Customer Orders.xlsx** shows orders by date for a company's customers on the Data worksheet. Many customers have ordered more than once, so they have multiple entries in the list. Write a sub that asks the user for a total (for example, $3000), finds the total amount spent by each customer on the list and reports those whose total is more than $3000 (the amount provided by the user) on a new worksheet called **Report**. Have the program ask the user for a total until the user enters a valid value.

As part of your sub, sort the list on the Report worksheet in descending order by total amount spent. (Hint: The orders in the Data worksheet are currently sorted by date. It might be helpful to use VBA to sort them by Customer ID. Then at the end of the sub, restore the list to its original condition by sorting on Date.)

Make sure your code checks for the Report worksheet before it does any new calculations. Any previous existing Report worksheet should be deleted before a new one is created.

Create your Results worksheet so it looks something like this before your data gets filled in:

	A	B	C
1	**Customers who spent more than $3000**		
2			
3	Customer ID	Total amount spent	
4			

Notes and Tips:

- Be sure to include basic data validation/error checking to avoid data type mismatches and other errors. Have the program ask the user for a total until the user enters a valid value.
- Make a button on the **Data** worksheet to run your subroutine.
- Save the file with the data and your code as *username*_a2.xlsm where *username* is your Novell login.
- Ensure your name and current date are at the top of each module.

Part 2: VBA In Word

This part is done in Word.

Create a new Word document. You will be recording macros in it, so you will have to save the file as a Macro-Enabled Word Document.

For tips on recording a macro read
http://bohr.wlu.ca/rhenderson/cp212/labs/macros_saving_word_macros.html.

(You will notice that a lot of functions become unavailable when you record a macro in Word.)

Record a macro of the following actions:

- Type **Using D2L** in the first line. Press **Enter**.
- Select that text and format it as a **Title** using the *Styles* tool (on the Home Tab in Word, click the Title style). Use your cursor keys to move down to the line below.
- Insert clipart of a student. Search for student in the clipart library and pick one you like.
- Close the clipart task pane.
- Click the title paragraph (Using D2L) and center the text.
- Stop recording.

At the botton of your document, place a button that will run the macro.

Submit both the Word document and the Excel file to the Dropbox in MyLearningSpace before the due date. Do not zip them.

Assignment Rubric

Excel Application Rubric v1

Criteria	Excellent	Good	Amateur / Satisfactory	Unsatisfactory / Needs Improvement
Code Comments	4 points Comments clarify meaning where needed.	3 points Comments usually clarify meaning. Unhelpful comments may exist.	2 points Some comments exist, but may be unhelpful or misleading.	0 points No commenting included.
Execution	4 points Program compiles / runs correctly. Program correctly solves problem in all cases. Shows you know what you are doing.	3 points Runs correctly but some pieces missing. Minor errors occur that may not interfere with the correct solution.	2 points Does not run correctly. Major errors. Some components not working.	0 points The program does not run. The file or application cannot be opened. The instructor is unable to mark anything.
Interface Design	4 points Straight forward, easy to use, visually appealing. Interface is user-friendly. Instructions and final output are clear. Uses informative prompts.	3 points Easy to use. On-screen instructions and output are mostly clear, correct and attractive.	2 points Screens are not very clear, output may be barely readable. Not very user-friendly.	1 point Cannot understand how to use your program, but it looks like it does something. Instructions unclear, not user-friendly.
General Design / Modularity / Efficiency / Usability	4 points Modular design - good use of functions, subroutines, and modules. Enjoyable to use.	3 points Mostly modular - good functions and subroutines but only one or too few modules. Solution is easy to follow.	2 points Confusing code. Major components missing. Some design flaws. Poor modularity.	1 point Poor overall design. Most components missing. Serious design flaws. No modularity.
Overall Score	**Excellent 14 or more**	**Good / Very Good 11 or more**	**Satisfactory 8 or more**	**Poor 0 or more**
	A great assignment. Very little could be improved.	Good piece of software. Solves most of the problem and works well.	Program meets the minimum requirements in most aspects but has some flaws.	Program barely runs if at all. Major components missing.