Final Programming Project Assessment (200 Points toward Course Grade)

Instructions: Follow the Software Development Process to complete this project. The following problem can be solved by a program that uses three basic tasks: Input Data, Process Data, and Output Results. To process the data, use file, looping, array, decision, accumulating, counting, find min/max and sorting techniques.

1. **Create** an MS Word document containing a **hierarchy chart** and a **data flow diagram** to organize your program modules. These charts **must** accurately reflect the final C++ program implementation. Safe the MS Word document as a PDF file and submit only the PDF file.
2. **Create** a design flowchart for you program. This flowchart **must** reflect the final C++ program implementation as closely as possible. Use any program you wish to create the flowchart, but copy it to an MS Word document and save it as a PDF file. Submit only the PDF version of your flowchart.
3. Implement (**create**) the final program with **Visual Studio C++**. You MUST use **Modular Programming** techniques by using appropriate functions in your program.

Problem Statement

Cornell Auto Sales has 10 salespeople. Cornell wants to produce a combined monthly sales report for all salespeople. Cornell wants you to write a program that will read from an input data file and produce two reports. Each report is to be sent to both the **console** screen **and** to output **files**. You are to name the two output report files "report1.txt" and “report2.txt”. Download and use the input data file, ***sales.csv,*** provided separately within this assignment in Blackboard. DO NOT EDIT THE INPUT FILE.

The ***Report1.txt*** file output should include:

- A well formatted, unsorted list of the salespeople’s name and the monthly sales for each of them.
- At the bottom of the report, list the following:
  - The salesperson with the lowest sales for the month.
    - **No** hard-coding here. Must use lowest value search algorithm presented in Lesson 6.
  - The salesperson with the highest sales for the month.
    - **No** hard-coding here. Must use highest value search algorithm presented in Lesson 6.

The ***Report2.txt*** file output should include:

- A well formatted, sorted (ascending order) list of the salespeople’s names and the monthly sales for each of them.
- At the bottom of the report, list the following:
  - The total combined sales for all salespeople.
    - **No** hard-coding here. Must use an accumulator.
  - The average sales for all salespeople.
    - **No** hard-coding here. Must do the calculation.

You must submit the Report1 and Report2 output files with your C++ program to show that your program ran correctly.

Proper formatting of your output files is critical in this project. Make them as presentable as possible with column headings, two columns of data and summary information at the bottom. There are plenty of examples in the textbook and learning activities of this course.

Other Requirements:

- Internal Documentation: Model all comments and documentation using the *The Art of Code Documentation.pdf* article provided in the early lessons of this course. Use *block comments* to document ALL functions of your program. Generously provide in-line comments within your code.
- Test and debug your Program: Use the provided data file, run the program, then *check your answers* with a calculator or Excel. If something does not match up, then fix your program.

- Program **must execute** and **produce correct output**.
- Read this document again to make sure you covered all requirements.

Submission Instructions:

- **You will submit 3 files for this project**.
    1. **Hierarchy Chart** and **Data Flow Diagram** in **one PDF** file (both in same document).
    2. **Flowchart** in a **PDF** file.
    3. **Windows Compressed Folder (.zip file)** containing your **Visual Studio C++ solution/project**.
- Each file will be uploaded separately into the submission area for each question shown in the Final Project Self & Peer Assessment.
    1. ZIP your entire project into a Windows Compressed folder (.zip file) and ATTACH it as your submission.
    2. Once you have uploaded your files, be sure to click the submit button. THEN, open the questions back up, download your submitted files, and check to make sure they will open. This way, you KNOW that your assignment was properly submitted and ready for peer reviews.
    3. If you find that the assignment did not submit correctly, submit it again. You can always resubmit your files up until the due date/time noted as the end of the "submission period."