SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

ENSC 251 D100 – Software Design and Analysis for Engineers (4 sem. hrs.)
Summer 2018

Lab 8 – Word Index

Assigned	Mon July 16, 2018 @ 8:30am
Due	Sat July 21, 2018 @ 9:00am.

This is an individual assignment.

- You may consult with professor and TA about any aspect of the assignment.
- You may consult with other students only in a general way, e.g., about debugging or C++ issues, or questions about wording on the assignment.
- You cannot actively work with another student in this assignment.
- You may not consult with other source for solving this problem.

General Information

- Use the header format. Replace text in GREEN with the appropriate information.

```
/**
 * @file XX.cpp
 * @author FIRSTNAME LASTNAME
 * @Date   DATE
 * @version 1.0
 * @section DESCRIPTION
 */
// I declare that this assignment is my own work and that I have correctly acknowledged the
// work of others.  I acknowledged that I have read and followed the Academic Honesty and
// Integrity related policies as outlined in the syllabus.
//
// _____ (PRINT YOUR NAME HERE) ____      _____(DATE)_______________
//
// ____ (STUDENT ID) ______
//
```

1. Specifications

The following is an example of a text file named “StockPrices.csv”, where csv stands for comma-separated values. This file contain the stock prices of Apple Inc. (AAPL) from the last

10 years. Figure 1 shows an example of the first few lines of this file's content. The data was obtained from YahooFINANCE! [1]

```
Date,Open,High,Low,Close,Adj Close,Volume
7/16/2008,24.314285,24.704287,24.085714,24.687143,16.647926,186947600
7/17/2008,24.871429,24.997143,24.484285,24.544285,16.551584,189381500
7/18/2008,24.074286,24.235714,23.571428,23.592857,15.90999,217103600
7/21/2008,23.842857,23.928572,23.017143,23.755714,16.019814,340117400
7/22/2008,21.285715,23.251429,20.932858,23.145714,15.608453,469898100
7/23/2008,23.57,24.052856,23.08,23.751429,16.016922,265442100
```

Figure 1: Example display of the content in StockPrices.csv

The first line explain the format of the content in the file. We will be only using the data at the third column (i.e. the high prices) only.

For this assignment, you are to write one or more programs that read the contents of the file and perform the following calculations:

a) **Average Price Per Year**: Calculate the average price of gas per year, for each year in the file. (The file's data starts in July 16, 2008, and it ends in July 16, 2018.)
b) **Average Price Per Month**: Calculate the average price for each month in the file.
c) **Highest and Lowest Prices Per Year**: For each year in the file, determine the date and amount for the lowest price, and the highest price.
d) **List of Prices, Lowest to Highest**: Generate a text file that lists the dates and prices, sorted from the lowest price to the highest.
e) **List of Prices, Highest to lowest:** Generate a text file that lists the dates and prices, sorted from the highest price to the lowest.

You can write one program to perform all of these calculations, or you can write different programs, one for each calculation. Regardless of the approach that you take, you should read the contents of the StockPrices.csv file, and extract its data into one or more STL containers appropriate for your algorithm.

Remember, we may use a different test file to test your program (i.e. contain a different time interval), so please design your software accordingly.

2. Submission Instructions

You can use the example zip file from lab 1 as a starting point. Create *.cpp file as needed. Modify the makefile such that it will compile your code into a binaries executable.

a) Create a directory with your name, e.g. "\LastnameFirstname", where Lastname is student's last name and Firstname is the first name.

[1] https://finance.yahoo.com/quote/AAPL/history?period1=1216191600&period2=1531724400&interval=1d&filter=history&frequency=1d

b) Save the files (*.cpp, other files, and makefile) in this directory. Uses these files as a starting point to write the following program.

For example, for student Mary Smith, this is a general example for a homework that has two parts. The directory and files will be as follows:

```
\SmithMary
      \partA
           \*.cpp   [e.g. Patient.cpp]
           \makefile
      \partB
           \*.cpp   [e.g. Car.cpp]
           \makefile
```

Then Zip up the directory “\LastnameFirstname” and the files within this director into a zip file “2018-2-ENSC251-LastnameFirstname.zip.” Submit the zip file to Canvas before the deadline.

3. Resources

- C++ Formatter https://codebeautify.org/cpp-formatter-beautifier
- Vim Basics - https://www.howtoforge.com/vim-basics
- Common Linux Commands http://www.dummies.com/computers/operating-systems/linux/common-linux-commands/

4. Rubric for marking

Criteria	Ratings				Pts
Program Specifications / Correctness	**Excellent** - No errors, program always works correctly and meets the specification(s). 50.0 pts	**Adequate** - Minor details of the program specification are violated, program functions incorrectly for some inputs. 40.0 pts	**Poor** - Significant details of the specification are violated, program often exhibits incorrect behavior. 30.0 pts	**Not met** - Program only functions correctly in very limited cases or not at all. 0.0 pts	50
Readability	**Excellent** - No errors, code is clean, understandable, and well-organized. 20.0 pts	**Adequate** - Minor issues with consistent indentation, use of whitespace, variable naming, or general organization. 16.0 pts	**Poor** - At least one major issue with indentation, whitespace, variable names, or organization. 12.0 pts	**Not met** - Major problems with at three or four of the readability subcategories. 0.0 pts	20
Documentation	**Excellent** - No errors, code is well-commented. 20.0 pts	**Adequate** - One or two places that could benefit from comments are missing them or	**Poor** - File header missing, complicated lines or sections of code	**Not met** - No file header or comments present. 0.0 pts	20

		the code is overly commented. 16.0 pts	uncommented or lacking meaningful comments. 12.0 pts		
Code Efficiency	**Excellent -** No errors, code uses the best approach in every case. 5.0 pts	**Poor -** Code uses poorly-chosen approaches in at least one place. 3.0 pts	**Not met -** Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion 0.0 pts		5
Assignment Specifications	No errors 5.0 pts	Minor details of the assignment specification are violated, such as files named incorrectly or extra instructions slightly misunderstood 3.0 pts	Significant details of the specification are violated, such as extra instructions ignored or entirely misunderstood 0.0 pts		5
				Total	100

-END-