

Assignment 7: A List of Stock

CIS 210 – Fall 2016

Point: 100

Project Due: 11:59 PM on Thursday, 10/13/2016

Learning Objectives:

- ❖ To understand the concept of Array.
- ❖ To create and use arrays to store objects.
- ❖ To use sequential search algorithm to find a target value in an array
- ❖ To read from and write to a text file in Java
- ❖ To write your own testing cases to test your program.
- ❖ To use debugger for debugging your Java program.

Problem Statement

You are required to read in a list of stocks from a text file and write the sum and average of the stocks, the name of the stock that has the highest price, and the name of the stock that has the lowest price to an output file. The minimal number of stocks is 30 and maximal number of stocks in the input file is 50. When the program runs, it should do the following:

- Use the Stock Class created in Assignment 6 and modify it as it needed by this assignment.
- The Stock class must have a constructor with 3 parameters that correspond to the 3 fields: Stock Name, Symbol, Price.
 - For the argument constructor, set the value of 3 fields based on the arguments.
- The Stock class must have accessors and mutators for all of its fields.
- The setter methods must protect a user from setting the price into negative number. You can set the values to 0 if a user tries to change them to negative values.
- The program will ask user to enter a name of the input file. If the input file does not exist, the program should print an error message and terminate immediately.
- If the file opens successfully, then the program reads stocks' information and store it into an array of stocks. Each element of the array MUST be a Stock object.
- The program calculate the sum and average of the stocks' prices.
- The program finds the name of the stock with the highest price and lowest price
- The program asks user to enter a name for an output file. Then it creates the output file and write the sum, average, name of the stock with highest/lowest price to the file.

Note: your program must be user-friendly and intuitive. This is a part of your grade. In other words, even if your program does everything the problem statement states, your grade may be reduced because of difficulty to it.

Input

This program requires that you read in the following data values:

- An input file name.
- An output file name.
- An input file that contains a list of stocks with their name, symbol, current price. Each line represents one stock and each field is separated by comma. The file **MUST** contain **at least 30** stocks.
 - E.g., Microsoft Corporation, MSFT, 23.34
 Google Inc, GOOG, 786.79
 Bank of America, BAC, 16.76
 AT&T Inc, T, 34.54

You will use ***interactive I/O*** in this program. All of the input must be validated if it is needed. You can assume that for a numeric input value, the grader will enter a numeric value in the testing cases.

Output

Your program should display the sum and average of the list of stocks, the name of the stock with highest price and the name of the stock with lowest price on the console, and write them to a file.

Use of Methods, Parameters, Modularity, Design, etc.

Part of your grade on this and ALL future programming projects in this course will be determined by how well you use multiple functions and parameter passing appropriately and how well you design a modular and functionally cohesive program using the principles discussed in class. Large grade point penalties can be incurred for not setting up a modular, well designed program structure. This emphasizes good program structure, design, and fundamental software engineering principles.

Miscellaneous

Be sure to read and understand the sections in the Course Syllabus handout on general project requirements. Also be sure to study the style, documentation and formatting guidelines discussed in the *Programming Style Guidelines* handout and in the lecture.

What File To Turn In and How to Turn In Your Work using Blackboard

You must turn in your Java program source files in a zip file. The zip file must include a Stock.java file which defines the Stock class, and a StockList.java file which includes the main() method. The zip file's name must follow this format: hw7_ListOfStocks.zip