COSC 2100/2010 (Fall 2017)

Assignment IV

Due by Monday, October 30, 2017 6:00 PM CST

**Submission Instructions (Please read):**

* Copy all source files (from both questions) into one a single folder named as **<GroupMember1\_Fullname>\_<GroupMember2\_FullName>\_Project4** then zip that folder and upload to D2L.
* Make sure to **put the full names of group members in the Java files.**
* One submission per group
* Name your driver class (i.e., the class with the main method) as **StockTransaction.java**
* Make sure that your driver class has several test cases.
* **Please use the folder naming format described above. Do not name folders such as src, project4, and myproject.**

In this assignment, you will use **double linked lists to calculate capital gains of stocks**. You are NOT allowed to use any of the Java Collections Framework classes such as Java ArrayList, LinkedList, HashMap, etc to implement this. The DLLNode class is available from your textbook code, support package and partial implementation of DList is given to you in DList class attached to the dropbox folder.

**Calculating capital gains**

When a share of common stock of some company is sold, the capital gain (or, sometimes, loss) is the difference between the share's selling price and the price originally paid to buy it. This rule is easy to understand for a single share, but if we sell multiple shares of stock bought over a long period of time, then we must identify the shares actually being sold. A standard accounting principle for identifying which shares of a stock were sold in such a case is to use a FIFO (First In First Out) protocol--the shares sold are the ones that have been held the longest (indeed, this is the default method built into several personal finance software packages). For example, suppose we buy 100 shares at $20 each on day 1, 20 shares at $24 on day 2, 200 shares at $36 on day 3, and then sell 150 shares on day 4 at $30 each. Then applying the FIFO protocol means that of the 150 shares sold, 100 were bought on day 1, 20 were bought on day 2, and 30 were bought on day 3. The capital gain in this case would therefore be 100\*10 + 20\*6 + 30\*(-6), or $940.

You will track individual shares by their *unique* stock code. You will read two input files : stocks.txt and transactions.txt. Sample files are attached to Dropbox folder.

The stocks.txt file contains stock codes and descriptions in the following format:

AAPL;Apple Inc.

IBM; International Business Machines Corp.

KO; The Coca-Cola Company

FB; Facebook Inc.

SBUX;Starbucks Corp.

The transactions.txt file contains different transactions in chronological order a user conducts on shares. The format will be as follows

AAPL;buy;10;$450

KO;buy;100;$30

IBM;buy;50;$150

FB;buy;10;$30

AAPL;sell;5;$480

KO;sell;50;$50

FB;buy;30;$40

IBM;buy;50;$50

FB;sell;20;$10

SBUX;sell;30;$50

So for example, the first line indicates to buy 10 Apple Inc. shares at $450. The second line indicates to buy 100 shares of The Coca-Cola company at $30, and so on.

Your program will take an input of a stock quote and then simply print the realized capital gain generated for the given input stock. You will generate realized capital gain only when you sell a stock. If the stock code does not exist in the stocks.txt file, you will print an appropriate error message. If the input stock code has not yet been sold, you will also display an appropriate message. Finally, handle any of the exception conditions that you can think of.

Name your main program as StockTransaction.java. Some sample runs of the program are given below (user input is shown in bold) :

Java StockTransaction

--successfully read stocks.txt and transactions.txt

Please enter a input stock quote for realized gain(or loss) for the stock : **AAPL**

Congratulations, your realized gain for Apple Inc. is : $150

Java StockTransaction

--successfully read stocks.txt and transactions.txt

Please enter a input stock quote for realized gain(or loss) for the stock : **FB**

Sorry, your realized loss for Facebook Inc. is : $500

Java StockTransaction

--successfully read stocks.txt and transactions.txt

Please enter a input stock quote for realized gain(or loss) for the stock : **IBM**

Sorry, no realized gain(or loss) reported for International Business Machines Corp.

Java StockTransaction

--successfully read stocks.txt and transactions.txt

Please enter a input stock quote for realized gain(or loss) for the stock : **XYZ**

Sorry, the stock quote does not exist in the system.

Java StockTransaction

--successfully read stocks.txt and transactions.txt

Please enter a input stock quote for realized gain(or loss) for the stock : **SBUX**

Sorry, there is an error condition associated with Starbucks Corp. The number of sold shares exceeds the total buy quantity.

The above problem description, sample tester class, and its output given above should be sufficient for you to complete this assignment. Your programs need to have well defined methods. It is just not enough to get the programs working correctly. You are required to follow all the object oriented principles, consistent coding style and indentation for your programs.

**Notes**

This assignment can be submitted by groups of 1 or 2. You may choose your own group member. If you need a group member, please contact the instructor. You may only choose a group member in your own section (9:00 AM or 10:00 AM) class. **Enroll in one of the available groups in the Assignment 4 group category on D2L.**

**Important Reminder**

This assignment can be done in groups of 1 or 2. All solutions must be group members’ own work. Academic honesty policy is strictly enforced in this class. Violators of this policy will be reported to the Provost's office. There are no exceptions. Please refer to the course syllabus for the academic honesty policy. If you have any questions about this policy, please contact the instructor.

**Grading (for each question)**

|  |  |
| --- | --- |
| **Grading Criterion** | **Points** |
| **Does not compile** | **0** |
| Compiles | 4 |
| Input, Output and Computation is valid | 26 |
| Test cases (The program demonstrates that it works under different test cases) | 10 |
| Coding Style (indentation, variable naming, etc.) | 5 |
| Documentation (comments are provided, author name is given at the top of the program) | 5 |