

**Assignment 1:** You are hired to fix a program that someone else wrote. The company fired this person and hired you to take over their work. Luckily the program is small, simple and almost done.

### The Incomplete Stock Analysis Application

Companies and people often buy and sell stocks. Often, they buy the same stock for different prices at different times. Say a person owns 1000 shares a certain stock (such as Google) he/she may have bought the stock in amounts of 100 shares over 10 different times with 10 different prices.

We will analyze two different methods of accounting, FIFO and LIFO accounting used for determining the "cost" of a stock. This information is typically calculated when a stock is sold to determine if a profit or loss was made. In our version of FIFO accounting, the price of a stock is averaged starting with the first purchase of that item. Say we sell 250 shares of a stock, according to this method the purchase price is determined by averaging the prices on the **first** 250 shares bought. In our version of LIFO accounting, the price of a commodity is averaged starting with the last purchase of that item. Say we sell 250 shares of a stock, according to this method the purchase price is determined by averaging the prices on the **last** 250 shares bought.

We decided to use a queue for storing data for FIFO accounting, and a stack for LIFO accounting. We used a linked list for the implementation of the stack and queue. We decided to test the system with just two stocks, Google and Amazon. We built a simple command line interface to test our program design.

Both our stack and queue have stock objects with the following fields:

- The name of the stock
- The number of shares of the stock
- The purchase price

You can assume that the first element of the structure is the stock bought first, the second was bought second, etc.

Bob who just finished his part wrote the user interface. He made it so the user is able to enter information about various stocks, the number of shares, and the price. The user can then enter a query about a

certain stock and the cost according to the LIFO and FIFO accounting methods for a certain number of shares.

It's a simple command line interface that goes something like this: "Enter 1 for Google stock or 2 for Amazon, 3 to quit:" Next, the system will prompt, "Input 1 to buy, 2 to sell:" After that we get a prompt to enter "How many stocks:" the user wants to work with.

If the user is placing an order to buy, they will enter the number of shares and the price.

If the user wants to sell, the user needs to enter the number of shares and the selling price. Bob forgot to ask for the selling price. He is only asking for the number of stocks. He left out a lot of the calculations as well. We fired him because he didn't put any error checking in the program. A user can attempt to sell a stock they haven't bought. Right now, this is causing a runtime exception. We hired you to fix these issues and make some minor modifications to the features. Specifically, here is what we want you to do.

1. Add the missing functionality in "controller.java" to finish the "sellLIFO" and "sellFIFO" methods.
2. Provide error checking so users can't sell a stock they don't own or try to sell more stock than they own. Eliminate that runtime exception.
3. We think its sort of limited to only work with just two stocks. You need to change the user input, so the user enters the stock name they wish to buy or sell instead of just choosing between Google and Amazon. Keep the command line interface since we are only in the testing phase. Eventually we will have you create a GUI for this program but right now, we prefer a simple command line interface.
4. We like the layout and design of the application so don't change any of the other methods or objects. Only edit the Controller.java file. We have given you the other files (Stock.java and Driver.java) for your reference and testing. DO NOT make any changes to these two support objects.
5. When you are done, just give me the finished Controller.java file.

Email your source code files to [barbara.hecker@csueastbay.edu](mailto:barbara.hecker@csueastbay.edu) with the subject line of [your last name] + " - Prog1" for example, "Hecker - Prog1."