

Due the week before the final week in May.

1. Pick at least two hot stocks (say the FAANG group, i.e. Facebook, Amazon, Apple, Netflix, and Google) and download their historical daily trading prices. You will use API to download the data in JSON format if possible. You may try to download such data for free from many websites. One of them could be <https://www.alphavantage.co/>. If that is not possible, there are some paid website also available. If that's the case, let me know and I will see what I can do.
2. You will store the data in the most efficient data structures you have learned so far for the following operations:
 - a. The user will be able to search for the daily price of each or all of the stocks included in your dataset. If the markets are closed on that day, the program will report that fact. Also, if the day specified is prior to the day the company went public, you would also report so as well.
 - b. The user is also allowed to add new data manually into the data structures whenever new data become available.
 - c. The user can search for data for any stock for a specific range of dates and the program will report the average price and the highest and lowest prices and the associated dates within that range.
 - d. (Extra credit) The program can display the historical prices within any specified range using graphs.
 - e. The program must have GUI interface using JavaFX. You may build the GUI by hard-code it or by using SceneBuilder.
3. This kind of tools is used to design trading strategies. Play your program to see if you can come up with a strategy that can potentially make you money by beating the plan old buy-and-hold strategy favored by Warren Buffett, arguably the best investor of all time. Even if you cannot, you need to make some attempts and how your attempts failed to beat the market.

Here are the basic requirements:

1. Design the program so it is concise, easy to understand, and efficient by using the most appropriate data structures for the task.
2. All the Object-Oriented programming principles should be followed. Classes should be as pure as possible. Methods should be refactored so they are as short as possible and named as appropriately as possible. Utility classes and methods should be separated from the real business logic components of the program.
3. Use Java API for the data structures rather than developing your own.
4. GUI should look reasonably nice and neat.
5. Write an article to explain your thinking process while working on the project, why you choose the data structures, and overall what you've learned. Write it in such a way so someone -- your brother, sister, or boy/girl friend -- who has no idea about programming can understand. If your writing is good, you may post it on Medium.com to get extra credit as well.