

## A Simple Stock Data Analyzer

### Goals:

- Learn how to choose a data structure for the desired performance.
- Practice with more Java collections, like HashMap and TreeSet.
- Learn how to customize a comparable data structure.
- Learn how to build an in-memory database for data analysis.

**This project can be done with a single partner.**

### Tasks:

\*\*\* Since YahooFinance API has been discontinued, just assign a random value to each stock price.\*\*\*

You would create a *myStock* class to implement the database and provide the following functionalities such that users can conduct simple stock data analysis.

- Users can quickly retrieve the current price and company name using the stock symbol.
- Users can quickly insert and update the records.
- Users can quickly retrieve the stock record with top k prices

To achieve these functionalities, your task is to implement the following methods in

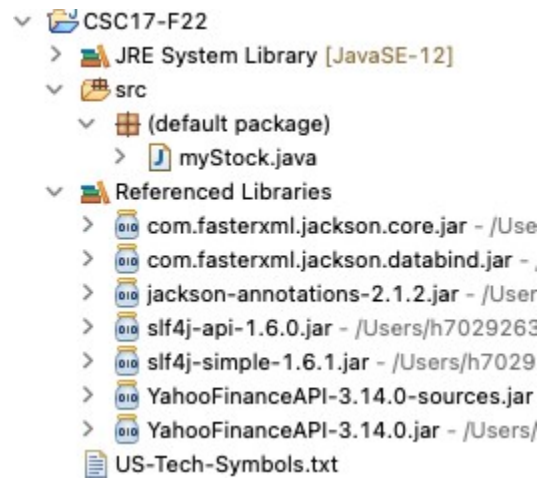
myStock.java which is provided in the starter code. :

```
myStock() // initialize the in-memory database with proper data structures
stockInfo get(String symbol) // get the record using the symbol; runtime is O(1)
void insertOrUpdate(String symbol, stockInfo stock) // insert or update one
record; runtime is at least O(Log(n))
List<Map.Entry<String, stockInfo>> top(int k) //returns a list of records with
top k prices; runtime is O(k)
```

A list of stocks is provided in the file *US-Tech-Symbols.txt* where each line contains a stock symbol and the company name. The real-time price for each stock is a random value within [0, 1000]. According to the test code in myStock.java, the following output is expected (the actual price of the stock may vary).

```
=====Top 10 stocks=====
[1]CCUR CCUR Holdings, Inc. 999
[2]CABO Cable One, Inc. 718
[3]EQIX Equinix, Inc. (REIT) 652
[4]FICO Fair Isaac Corporation 611
[5]AVGO Broadcom Inc. 513
[6]MSCI MSCI Inc. 499
[7]LRCX Lam Research Corporation 454
[8]FDS FactSet Research Systems Inc. 432
[9]TDY Teledyne Technologies Incorporated 413
[10]INTU Intuit Inc. 399
=====Stock info retrieval=====
VMW VMware, Inc. 115
BIDU Baidu, Inc. 96
```

Please closely follow the comments and hints in myStock.java for details instruction when you are implementing the tool. Please make sure to include the library files for YahooFinance API in your project's build path to allow successful compilation. To do that, right-click your project and choose "Build Path -> Add External Archives", and select all library files (i.e., the jar files) from your computer. The project structure should look like the following one if everything is added correctly:

**Submission:**

Please submit your completed myStock.java and a screenshot of your output to Blackboard. You won't receive full credits if you fail to submit the screenshot.