

# Homework 3: Stock Tracker

## Due Date

Friday, March 11 @ 2 hours after 11:59pm (2am Saturday morning)

## Objectives

- Practice with set and map
- Custom Sorting

## Resources

Set: <http://www.cplusplus.com/reference/set/set/> (<http://www.cplusplus.com/reference/set/set/>)

Map: <http://www.cplusplus.com/reference/map/map/> (<http://www.cplusplus.com/reference/map/map/>)

## Stock Tracker

You will be implementing a C++ class that will track stock prices. There are no data structures given for this assignment and it is your task to decide how to store the stock information in order to answer several queries about the data. Your class will be given the total number of shares outstanding for each stock, which will not change, and the current price of each stock which can change often. The assignment will build up to a function that asks for the top k stocks by market capitalization which is its current price multiplied by its number of shares outstanding.

## Registering Stocks (1 point)

```
void registerTicker(string tickerSymbol, int sharesOutstanding);
```

```
bool isTicker(string tickerSymbol);
```

```
int getSharesOutstanding(string tickerSymbol);
```

Implement these functions to register a stock with a certain number of shares outstanding and answer queries on whether a symbol has been registered and the number of shares for each stock. If the number of shares is asked for a ticker that has not been registered, it should return 0. Once a stock is registered, its number of shares outstanding will not change (Note: this is not true in general, but is true for this assignment).

## Tracking Stock Prices (1 point)

```
void updateCurrentPrice(string tickerSymbol, double price);
```

```
double getCurrentPrice(string tickerSymbol);
```

```
double getMarketCap(string tickerSymbol);
```

Implement these functions to track the price of a share for each stock. Unlike the total number of shares outstanding, the price of a stock can change often so `updateCurrentPrice` can be called many times on the same stock. If the price for a ticker has never been set or it was never registered, `getCurrentPrice` should return 0.0 for that symbol.

Market capitalization is the current market value of the company and can be computed by multiplying its price by its number of shares outstanding. If the ticker has not been registered or the price was never set, return 0.0.

## Top k Stocks by Market Capitalization (1 point)

```
vector<string> topMarketCaps(int k);
```

Return the ticker symbols of the  $k$  highest market cap stocks in order of their market cap. The highest market cap stock should be in index 0 of the return vector, the second highest in index 1 and so on. Ties in market cap can be broken arbitrarily for the ordering.

Be very careful not to remove duplicates if using a set or map. The grader will be set up to introduce duplicates by price, shares outstanding, and market cap, but there will be no duplicates on ticker symbol.

## Efficiency (1 point)

```
void updateCurrentPrice(string tickerSymbol, double price);
```

```
vector<string> topMarketCaps(int k);
```

These two functions must run in  $O(\log(n))$  time where  $n$  is the number of registered stocks to earn full credit on the assignment. The runtime for `topMarketCaps` can also depend linearly on  $k$  making its runtime  $O(k \log(n))$ .

Each of these functions will be called repeatedly to check the runtime.

## Code

Download and complete these C++ files with your solutions.

**hw3.zip (homeworkCode/hw3.zip)**

## Submission

Submit: `StockTracker.h`, `StockTracker.cpp`

Submissions must be made individually on the submission website: <https://www-student.cse.buffalo.edu/planetexpress/> (<https://www-student.cse.buffalo.edu/planetexpress/>)

## Solution

hw3-solution.zip (homeworkSolutions/hw3-solution.zip)

## Results



