

PROGRAM SPECIFICATIONS

In this assignment you will simulate purchasing stock. The stocks in your portfolio will be contained in a singly linked list. You will be able to buy stocks, sell stocks, and calculate your overall stocks value. See menu below for more detail. The initial set of stocks will come from a file with the name specified by user. The initial file format is explained below:

A stock is comprised of a Company structure (not a pointer just normal), number of shares, and purchase price. A Company is comprised of a company name, and ticker symbol. Information for each class is specified below.

Company

Company contains the following:

- Company Name – char *
- Ticker Symbol – char *

Stock

Stock contains the following:

- Company structure (not a pointer just a normal structure)
- Purchase Price – double
- Number of Shares – int

cscd240 s13 hw5Tester.c

- openInputFile – reads the file name and opens the file
 - countRecords – A stock is made of 3 lines – see example input file formatting
 - buildArray – returns a exact size array filled with Stocks
 - menu
- 1) printStockValue – will prompt the user for a file name – this file will contain symbols and current prices – You will determine based on the current prices the overall profit/loss and display this to the screen. This will be done for each stock and the overall total of all stocks.

Example:

Microsoft current price \$35.08

Current value:	\$28274.48
Purchase value:	\$99500.70
Profit/Loss:	(\$71226.22)

Google Inc current price \$908.53

Current value:	\$153541.57
Purchase value:	\$144440.22
Profit/Loss:	\$ 9101.35

- 2) buyStock – Allow the user to purchase more stock
 - a. If it is a brand new stock then a new element will be added
 - b. If it is an existing stock then the number of shares and the purchase price will be updated. Purchase price will be averaged. Number of shares and purchase price are read from the keyboard.
 - 3) sellStock – Allow the user to enter a symbol and number of shares to sell.
 - a. If the stock doesn't exist then display an error message
 - b. If the stock does exist ensure the user is not selling more than they own
 - c. If the stock does exist the user will be able to sell any number of shares up to the maximum owned. If the user only sells a portion, the shares will be appropriately updated. If the user sells all shares of a stock the array will be updated.
 - 4) printStockInfo – prints out each stock in the following format
Name (Symbol)
Shares: # of Shares
Price: purchase price
- cleanUp() – cleans up and gives back all the memory

Here is a small sample input

```
MSFT Microsoft
123.45
806
GOOG Google Inc
856.78
167
MCD McDonalds
34.50
100
GOOG Google Inc
678.98
2
```

OTHER SPECIFICATIONS

- You must validate all ranges
- Main is unchangeable and is in the file cscd240hw4.c
- **Additional information will be provided during class it is your responsibility to keep up to date on the information.**

TO TURN IN

A **zip** file that contains:

- All C and H files to make your code compile
- All input and output files
- makefile – with multiple targets (one for each group of C/H files) - executable target is hw4
- output run named cscd240hw4out.txt
- valgrind run named cscd240hw4valgrind.txt

You should know the zip naming scheme by now.

Get started ASAP!