CSCD 240
HW 5

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

Node

Node contains the following:

- void * data;
- struct node * next;

cscd240_s13_hw5Tester.c

- openInputFile() – reads the file name using fscaf, scanf, fgets – uses low level open to open that file for reading.
- buildList(fdRead,3) – This function will:
    - read the entire file in to a static char array of size 4096, using low level read
    - Tokenize that char array into different stock objects using strtok
        - A stock is made of 3 lines – see example input file for how the file is formatted
        - You will need to dynamically allocate a stock which will then be assigned to the void * data inside the linked list node
    - You will build the list in order based on symbol

- 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 then an overall total.

        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 node 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 using high level scanf, fscanf, etc.

    3) sellStock() – Allow the user to enter a symbol and number of shares to sell.

        a. If the stock doesn't exist in the list 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.

    4) printStockInfo() – prints out each stock in the following format

        Name (Symbol)

        Shares: # of Shares

        Price: purchase price

- cleanUp() – cleans up the linked list 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
- All output will be done via high-level printf, fprintf to the screen.
- Main is unchangeable.
- I have provided linkedlist.h for the structure specifications. You can't change any part of the Node
- I have provided stocks.h for the structure specifications. You can't change the types, or pointers, but you can change the variable names.
- All reading from files will be done with low level reads.
- All keyboard input will be high level
- Linked list will only contain list like things
- **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:

- cscd240_s13_hw5Tester.c
- hw5.c
- hw5.h
- linkedlist.c
- linkedlist.h
- stocks.c
- stocks.h
- stocks.txt
- makefile – target of hw5
- output run named cscd240_s13_hw5out.txt
- valgrind run named cscd240_s13_hw5valgrind.txt

You should know the zip naming scheme by now.

Get started ASAP!