CS 350 Spring 2016 **Lab 6: April 7, 2016**

Stock Market Pthreads Programming Exercise

Please write a multi-threaded program that simulates the stock market, using the pthreads library. Your program should take a two integer parameters. The first parameter's value falls between 1 and 1000 and specifies the number of threads your program will use; each thread should simulate a single stock with a floating point value that represents that stock's price. Each stock price should begin at 100.0, and you should initialize the overall market value by multiplying 100 by the number of stocks/threads. Then, in a loop, each thread should continuously generate a random number between 0 and 1, and then either increase or decrease the stock's value by that amount. (You may generate a number between -1 and 1 and add that to the stock price.) After changing the stock's price, the thread should then increase or decrease a shared overall market value.

You should also have two threads that "watch" the market. One thread watches for the market to drop below a pre-determined level, the other thread watches for the market to increase above some pre-determined level. To specify those two levels, a second command line parameter's value, *P*, falls between 1 and 20. If the market drops to *P*% below the market's initial value, then one of the two threads should recognize that, and should print "`Market Down to <Overall Market Value>`", where `<Overall Market Value>` is the final stock market value. Likewise, if the overall market increases by *P*%, the 2nd of the two watcher threads should print "`Market Up to <Overall Market Value>`". One of these two events should trigger all threads to exit.

For example, your program may be invoked as follows:

```
./lab6 55 10
```

Your program would then create 55 *stock threads* plus the two *watcher threads*. The initial market value would be 5500, and if the market rises to 6050, your program might print (for example):

```
Market Up to 6050.34
```

Your program should then print the sum of all the stock prices.

```
Total Market Price of 55 Stocks: 6050.34
```

The market value number should match the sum of the individual stock prices. Use pthreads mutex locks to protect shared data appropriately. Use pthreads condition variables for the two watcher threads.

If this document specifies how to do something (for example, one thread per stock), you must implement it that way for credit. If this document does not specify how to do something, you should decide the best way to do it, using what you have learned about pthreads and multi-threaded programming, and what you can find out on your own. Independent investigation is an important skill for systems programmers! Seek help if you need it!