COMPUTER PROJECT #8

Assignment Overview
In this assignment you will practice creating our own user-defined data structures to be used in a simple stock market simulation. You will utilize what you have learned to date in the process (STL containers, etc.). It is due on Monday, 3/31, two weeks because of the midterm. It is worth 50 points (5% of the overall grade).

Background
We will use some real data from the Dow Jones Industrial Average (http://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average), a set of 30 stocks that are used as an indication of the U.S. stock market. This sample is from 8/31/2012 to 6/14/2001.

Details
You will create a `Market` struct, that has the following features:

- a data member `map<long, vector<double>> stocks` of the stocks in the example file "dow.txt". The key will be the date and the vector will contain the 30 closing stock prices for that date.
- a method `double get_price(string stock, long date).`
  - returns the price of the stock on the date if:
    - the date is a valid date
    - the stock symbol is a valid stock symbol
  - returns a -1.0 otherwise
- a constructor that takes a single string argument, the file name of the stock prices, and fills the underlying map `stocks` .

You will create a `Player` struct that has the following features

- data member `double cash`, how much cash the player has.
- data member `map<string, long> stocks`, where the key is the stock symbol and the long is the quantity of that stock that the player owns.
- a constructor that takes a single parameter, the `double cash` the player starts with.
- a method `bool buy(Market &m, string stock, long date, long quantity).` An attempt to buy a stock by the player from the Market on the specified date.
  - returns true if the player:
    - has enough cash to make the purchase
    - the stock symbol is one of the valid 30 symbols
  - if true, purchase is made and the player info is updated
    - cash reduced, map stocks updated
  - if false, no action taken
- a method `bool sell (Market &m, string stock, long date, long quantity)`
  - returns true if the player:
    - has the stock to sell (can't sell what you don't have)
    - has at least the quantity indicated (can't sell more than you have)
  - if true, player info is updated
    - cash is increased, map stocks updated
  - if false, no action taken
- a method `to_str()`
  - returns a string representation of the player (see example)

Requirements
You will provide the following four files:

- player.h, the class declaration

- player.cpp, the class definition
- market.h, the class declaration
- market.cpp, the class definition

We have provided a `main.cpp` which will test each of your functions (many asserts). Your code must compile and produce correct output using our main **without modification.**

Deliverables
You must use handin to submit only player.h, player.cpp, market.h, market.cpp files.

Output

```
[15:28][579][bill@thub]~/classes/232/SS14/Projects/proj08
>./a.out
Player has:184.57 dollars, Stocks are:
AA, quantity 25
XOM, quantity 1

Player has:1000 dollars, Stocks are:
IBM, quantity 0

[15:28][580][bill@thub]~/classes/232/SS14/Projects/proj08
>
```

Assignment Notes
- Two files are provided
  - dow.txt, the data
  - dowNotes.txt that describe the data in the dow.txt file. In particular this file lists the stock symbols you must use in your program.
- how to map stock symbols to the index in the vector<double> stock values.
  - think about how you would do this. You might want to make a local data structure in the struct to help map this.