# CST126

## Lab 2 – Stock Picker

In this lab you will create a program that prints stock information.

### Input

This lab will use a comma separated input file quotes.csv. The format of the file is:

|  |  |  |
| --- | --- | --- |
| Description | **Type** | **Comments** |
| Stock Symbol | 1-4 characters |  |
| Stock Name | Up to 25 characters | The Stock name is surrounded by quotes. IE “Mentor Graphics” |
| Price | Double |  |
| Earnings | Double | Earnings per share |

### Logic

In this program you will practice using classes, reading from an input file and using string object and cstring functions. You will have a new class for stock information. This class should have a separate stock.h for the interface and stock.cpp for the implementation. The .h file should have code to make sure it is only included once.

Your class should have private member data for:

* stock symbol (string)
* name (string)
* price (double)
* earnings (double)

Your class should have the following public member functions:

* A function to print the stock information. (Must be in your classes .cpp file)
* A default constructor that sets the stock symbol and name to blanks and the price and earnings to zero.
* A constructor that takes as input the stock symbol, name, price and earnings.
* A function that returns the price to the calling function.
* A function that updates the price given a new price as a parameter of type double.
* A function that returns the P/E (Price Earnings ratio). The P/E = price / earnings.
* Your program should get rid of the “ around the name before loading it into the stock object.

Except the print function, any of the functions above may be fully implemented in the class definition. A function should go in the class definition only if it is small.

Your main program will have an array of at most 5 stocks. You will read the input file into this array. After you have read and printed the array you will simulate a 10% increase in stock price, and use your update price member function to update the stock price. You will then print this new price.

I’ve done a lot of the work in the main program for you. There are STUDENT TASKS to complete to get it finished up.

### Output

Here is a sample output based on the quotes.csv file. Your output should match this EXACTLY.

Welcome to the Stock Picker

Please enter the input file name including extension : quotes.csv

NWN Northwest Natural Gas Price: 58.30 Earnings -1.95 P/E 29.90

APPL Apple Inc. Price: 174.96 Earnings 1.90 P/E 92.08

GOOG Alphabet Inc. Price: 1029.27 Earnings 17.93 P/E 57.40

LUV Southwest Airlines Price: 55.08 Earnings 5.79 P/E 9.51

If you had a 10% appreciation in price the values would be

NWN Northwest Natural Gas Price: 64.13 Earnings -1.95 P/E 32.89

APPL Apple Inc. Price: 192.46 Earnings 1.90 P/E 101.29

GOOG Alphabet Inc. Price: 1132.20 Earnings 17.93 P/E 63.15

LUV Southwest Airlines Price: 60.59 Earnings 5.79 P/E 10.46

### Strategy

I strongly recommend that you chunk this lab up. Get started now. You won’t know how to do everything when you start and that’s ok. After every major step, compile and run and make sure it works.

Here is one ***possible*** task list that follows along with the lecture material.

1. Read from the file and just print out what is in the file.
   1. Use the temporary version of fill\_stock\_from\_file( ifstream &, int & count) that I’ve provided for you.
   2. Instead of the STUDENT TASK filling the stock, just print what you get from the file.
2. Make yourself a STRUCT for your stock.
   1. Leave it in the main cpp file for now.
   2. Put the minimum variables in your struct.
   3. Make sure your program still compiles and prints out from the file.
3. Pass a single struct into your fill\_stock\_from\_file( ifstream &, int & count) routine
   1. Fill it with the first or last record you read from the stock file.
   2. Print it your new struct in the main part of the program.
   3. I personally would leave the printing of the file in until the very last moment.
4. Create a routine to print your struct.
   1. Replace the print you did in the main program with this new print routine.
5. Move your struct into the stock.h and stock.cpp.
6. Change your struct to a class.
7. Add a constructor to your class and use it.
8. Add a member function to compute the PE ratio and print it.
9. Add a member function to increase the stock value by a percent.
   1. Print the result.
10. Change your program to read an array of stocks.
    1. Note: that you could leave the rest of your program the way it is and just pass in the first element of the array.
11. Change your program to use the whole array of stocks.