CA06

COMP 5700/6700: Comprehensive Assignment 06

Date: Apr 10, 2015 - Apr 19, 2015

Marks: 100

Teacher's Name: Munawar Hafiz

This is an individual assignment. Everybody has to submit

We want to hit two targets (with one stone/assignment).

1. Write test cases and debug code for correctness.
2. Make you learn about good coding practices and modify code to adhere to good coding practices.

Read the stock market price matching solution that I have posted. Then follow these steps.

Step 1: Test Cases (50 points):

You have already written test cases for the functions in the previous assignment. Since, you and I have the same API, most test cases should port seamlessly.

- **To Complete Test Cases Tab in the Excel File:** Port the test cases to this project. Enter info about these in the Excel file.
- **To Complete Test Cases Tab in the Excel File:** I have also created some private methods to break some tasks. These are pretty straightforward. Write new unit tests for these. Enter info about these in the Excel file.
- The Client program is also a test case. Think of it as a large integration test. Use this in the next (debugging) step.
- **To Complete Main Code Modification Tab in the Excel File:** Run the test cases and debug through the execution. Step through the code and see if the code is doing what it is supposed to do. You may be surprised by a few logical flaws in the code I have given. The flaws are about how the stock order data is stored and manipulated in the data structures. Identify and fix the flaws. Modify the main code. Keep track of these modifications in the Excel file.
- **To Complete Debug tab in the Excel File:** Run the integration test (client code) and debug through the steps, specifically step into the trade() method. Put in the Excel file the values of the main data structures (buyOrderTempArrayList (horrible name), sellOrderTempArrayList (horrible name), orderCumulative), as you are going through each line of code [Start from line 58 in the OrderBook class, step into the `calculateCumulative` function, and start keeping note]. Note that there will be many lines of entry in the Excel files. Skip the lines for which none of the three values change. However, if any one of the three values are modified, you have to list that entry.

Step 2: Good Coding Practices (50 points):
As you will identify bad coding practices in my code and fix them, run the test cases made in the previous step each time to ensure that you are not breaking anything with the changes you made.

You have seen the video discussing the good coding practices from the Clean Code book.
https://www.youtube.com/watch?v=HZJxjlvBbVA

You also have access to the good coding guidelines.
(Posted under 16-Coding Guidelines, Document names starting with 00 and 01... One is general good coding guidelines from the Code Complete book, the other is coding guidelines for Java programmers)

Follow these guidelines.

- Start from the code you produced in step 1. It contains my original code and a few modifications that you made to the code to fix some logical flaws.
- List as many problems in the code as you can find. The problems can be small. For example, for each bad variable name you found, you can have a separate entry for them. For each place, you are adding (or removing) a comment, you can list them as separate entries. Some fixes can take multiple steps, as in fixing a long function with a lot of code in it; list the multiple steps and describe what change is made in each step. **Look for about 50 problems.**
- Fix the problems.
- **To Complete Good Practices Tab in the Excel File**: Describe the problems, and how you fixed the code.

Deliverable:

1. The Excel File.
2. The Modified Project: Test Cases and Stylistic/Logical Issues Fixed.