Assignment 6
Due date: Nov. 2, 11:55PM EST.

You may discuss any of the assignments with your classmates and tutors (or anyone else) but all work for all assignments must be entirely your own. Any sharing or copying of assignments will be considered cheating.

You should not use any features of Java that have not been covered in class. If you have doubt if you are allowed to use certain structures, just ask your instructor.

Problem 1 (60 points): Object Oriented Potions Class with Professor Snape

Complete the Snape's challenges found at http://cims.nyu.edu/~kapp/cs101/potions/.
Submit files only for the final challenge.
Make sure to document your program using Javadoc style comments.

Problem 2 (40 points): Stock Class

In class we looked at an implementation of a 'Stock' class and a test program for the class. (See the worksheet at http://cs.nyu.edu/~joannakl/cs101.06_f15/notes/Java_classIntro.pdf.) Improve this class and its test program.

Your class should set and get methods for all data fields. All methods that can change their value of a data field should verify that the values are correct (prices should always be greater than zero, the symbols should contain only characters and be stored using uppercase letters). There should be a data field that indicates if a given Stock object is valid or not. A Stock object becomes invalid if any of its data fields is deemed invalid.

The class should have a toString method that prints all the information about the stock or the word "Invalid" is a Stock object is not valid.

Improve the test class so that it tests all of the methods of the class with valid and invalid parameters. This test program should print information to the screen (make sure your output is readable) regarding the status of its tests.

Call your files: Stock.java and TestStock.java.
Make sure to document your program using Javadoc style comments.

Grading

Does the program compile? If not, you will lose all the points for that problem.

Is the program properly documented using Javadoc style comments? (worth approximately 20% of each problem)

Proper documentation at this point in the course includes:

- preamble with the name of the author, date of creation and brief description of the program (the description should specify what the program does, not that it is a solution to problem 1 of homework 1);
- method comments - description of what the method does, description of each parameter, description of the return value
- inline comments - comments inside the code describing steps needed to be taken to accomplish the goal of the program;
- appropriately chosen variable names, i.e., descriptive names (a good name for the variable that stores the bonus amount in the last problem is bonus, not x);
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

Is the program well developed? (worth approximately 40% of each problem) Make sure you create variables of appropriate types, use control statements (conditionals and loops) that are appropriate for the task, accomplish your task in a well designed and simple way (not a convoluted algorithm that happens to produce the correct output for some unknown reason). You should also design a friendly and informative user interface.
Is the program correct? (worth approximately 40% of each problem) Make sure that your program produces valid results that follow the specification of the problem every time it is run. At this point you can assume a "well behaved user" who enters the type of data that you request. If the program is not completely correct, you get credit proportional to how well it is developed and how close you got it to the completely correct code.

What and how to submit?

You should submit three source code files combined into a single zip file to NYU Classes. Do not submit all the files that Eclipse creates, just the source code files that have .java extensions. Name your classes as specified in the problems.

If you wish to use your (one and only) freebie for this project (one week extension, no questions asked), then complete the form at http://goo.gl/forms/fpUJJrF64b5

before the due date for the assignment. All freebies are due seven days after the original due date and should be submitted to NYU Classes.

Questions

Post any questions you have regarding this assignment to Piazza under the "homeworks" topic.