CSC207H CODE REVIEW CHECKLIST

This document contains a code review checklist. These are a subset of the items found on checklists posted on the web; they also happen to be the kinds of things that TAs look for in every course that involves coding.
All criticism must be constructive. Don't be rude; don't insult; don't be sarcastic. The point isn't to complain, it is to improve the code.

Style

- □ Are the variable, method, class, and package names named according to the standard Java rules?
- □ Are variable, method, and class names descriptive and meaningful?
- □ Are class names noun phrases?
- □ Are method names verb phrases?
- □ Are abbreviations avoided in instance and static variable names and in method names?
- □ Has the Eclipse Format command been used?
- □ Are chunks of code separated by blank lines to make them easy to identify?

Documentation

- □ Does every class, method, and instance variable — whether private or not! — have a Javadoc comment (starting with `/**`)?
- □ Do methods have `@param` and `@return` tags?
- □ Can you write a call on each method based on *only* its Javadoc information? (If not, it needs to be rewritten.)
- □ Does the documentation match the code? (If not, which is incorrect?)
- □ Are there subsections of code that you don't understand? (If so, there needs to be more internal commenting, or refactoring to make the code cleaner.)
- □ Are comments written for other experienced programmers?
- □ Are there too many internal comments?

Design

- □ Are all instance variables `private`?
- □ Are any method bodies (from the opening brace to the closing one) longer than about 20 lines?
- □ Are there test cases for each `public` method?
- □ Are any getter methods giving unnecessary access to mutable instance or static data? (For example, a method that returns a `List` instance variable allows anyone to modify that list.)
- □ Are there methods that are not used?
- □ Is there duplicate code?
- □ Is there a simpler, clearer design?