Coding Standard Document

We are using Google’s CodePro AnalytiX to check our style conventions. The instructions to install CodePro Analytix into Eclipse can be found [here](https://developers.google.com/java-dev-tools/download-codepro).

We have decided to follow the default 94 rules in the rule set, CodePro Core. To set this go to Windows -> Preferences. Under CodePro select Audit. In the Rule Sets tab, select CodePro Core as your default.

Along with the default 94 rules we have decided to add the following rules to our rule set:

* Formatting
  + Line length
    - Maximum line length set to 100
  + One statement per line
  + Space after cast
  + Space around operators
  + Indent code within blocks
  + White space usage
    - Disallows tabs and allows just spaces
  + Brace position
    - Opening braces should not appear on new lines while closing braces must appear on new lines.
  + Space after commas
* Coding Style
  + Avoid nested assignment
  + Badly located array declarations
  + Conditional operator usage
  + Fully parenthesize expressions
* Indents must be 4 spaces; tabs are not allowed.
* For ordering fields in our class files, private fields must be first, the protected fields, then public fields, then private methods, then protected methods, and finally public methods.

For more detail on what each rule is doing, visit [here](https://developers.google.com/java-dev-tools/codepro/doc/features/audit/audit_rules_com.instantiations.assist.eclipse.auditGroup.formatting).

The exported .pref of our rule can be found [here](https://docs.google.com/file/d/0B66sUcs7lhDVZ0ZMM0lZTzZ6RGM/edit?usp=sharing).

We plan to use the CodePro Analytix tool to ensure all members of our group follow the coding guidelines outlined above. Prior to any pull request to the master repository, code must pass these rules. When we do code reviews we will also check each other’s coding styles to make sure we follow the style declared above. (Note: due to our inexperience with using Git, we sometimes ended up pulling/pushing at the wrong time in this process and fixing something *after* it was in the master repository instead of before.)

We will be doing code reviews in person and taking notes as we do in a shared google doc. If there is a small fix that can be done in a short amount of time, the fix will be done right away, during the review. Otherwise, once the review is finished, the coder will go back and fix the appropriate code sections if there are anything major. The coder will then notify the same reviewer to read over the code. If there are still any problems, the reviewer and coder will find time to do another in person code review. This process will be repeated until the reviewer finds that nothing else needs to be fixed.

To provide evidence of these reviews to the grader, the notes we will take during our reviews will be uploaded onto our GitHub Wiki before the due date, with the title “Code Review Document”. Note that because of the flexible, in-person style of our code reviews, some of the smaller errors that are fixed right away will not be in our notes, or some errors mentioned in the notes will seem to never have appeared in the GitHub repository.