Expectations on Homework

General considerations

This is a large class, and there are many homework assignments to be graded each week. The course staff has procedures in place to help reduce the time needed to complete the grading of assignments. These procedures rely on students following instructions for preparing and submitting the homework assignments. *If the graders need to handle your homework submission in a special way, you will lose points on that assignment. The penalty will be 25% of the total number of points for the assignment.* The penalty will be applied for any violation of the instructions, including, but not limited to

- submitting the assignment to the wrong project in InstructAssist
- submitting any file type other than .zip
- mistakenly submitting the wrong file (you submit your solution to Lab 2, for example, instead of submitting your solution to Homework 2)
- submitting your .class files

The best way to avoid penalties for these kinds of violations is to complete your homework ahead of the the due date. Mistakes are more likely if you're submitting an assignment with only minutes to spare.

For most assignments, we'll provide a stub file (usually called CompileCheck.java) for checking that your program compiles. *Programs that do not compile against this file will likely break the auto-grader, resulting in zero points for the auto-graded portion of the assignment!*

Assignment Guidelines

Follow these guidelines in preparing your assignments

- Put the names of all students for whom the assignment is submitted at the beginning of one of your testing files (such as in an Examples class).
- Annotate every method with a brief comment explaining what the method does. This is analogous to the 1101/2 purpose statement.
- Provide examples of data for all classes you define. Put these in one of your testing files (such as in an Examples class).
- Provide test cases for all of your methods. We will check that your tests are correct relative to the problem statement. We will also check that your tests are reasonably thorough in checking the various cases of the methods and their inputs. An assignment will often indicate specific methods for which we will grade your test cases particularly closely. For these methods, we will check whether your test cases detect our own set of broken solutions (by having at least one of your tests fail on our broken solution). Our broken solutions will have mistakes in the logic of the computation but will satisfy the input and output types required by the program.
- Use appropriate techniques to reuse common code (this will go into effect after Homework 1).

- Use newlines and indentation to produce clean, well-formatted code (we actually read your code; we don't just run it).
- Follow naming guidelines for methods, fields, interfaces, classes, and abstract classes (the naming guidelines will be covered in class).
- Submit only your .java files, not your .class files.

You do NOT need to:

- Comment individual lines within your code unless something subtle is going on.
- Provide 1101/2-style templates
- Include error checking in your code (until we cover exceptions later in the course).

Separating Tests from Code

Our auto-grading software will look for tests in a file called *Examples.java*. Tests that are left in any other files will not get detected (and hence not get graded). Similarly, the auto-grader will not look for classes or methods in these tests files, so non-test classes in these files will not get graded. Please be sure to separate your code and your tests so that the grading software can find them. You will lose points for failing to set up your files as described here. Your files can be organized into any subdirectories that you wish.