OOP and Data Structures	Discussion 2 — testing
CS2110 Summer 2019	Due 6/28 at 5PM

Before working on these problems, search for "testing" on the Java HyperText Read the following entries (1–2 pages each):

- 1. Writing a JUnit testing class in Eclipse, for example, to test the methods in a class
- 2. Testing whether an assert statement is correct: pdf file.
- 4. Eclipse JUnit testing to show code coverage: pdf file.
- Testing is the process of running a program to find errors in it. See this pdf file.
- Also search for "wrapper" and read the short pdf file that comes up.

Note: if you have a question, searching the HyperText is often a good way to answer it quickly! For example, search for loops, String, array, etc.

We prefer that you work together on these discussion questions and only submit one solution as a group. Our graders are also running at 2.5x speed over the summer! However, if you prefer to work solo, you may.

We've uploaded a (poorly written) class `Rectangle` on CMS. This class's implementor has chosen a strange invariant to maintain, and as a consequence, their code is pretty buggy. Luckily, they've provided clear documentation of their intentions, so finding and fixing their bugs won't be too hard.

Do the following:

1. Load `Rectangle.java` into a new eclipse project.
2. Create a JUnit test case for it.
3. Create some black-box tests and some white-box (also called glass-box) tests to find the bugs.
4. Correct the bugs. Do not change the fields or invariants.

Think about, but don't turn in: what would be a better choice of representation / class invariant to simplify the code and solve a lot of these bugs?

Form a group on CMS and submit the files `RectangleTest.java` and `Rectangle.java`.