

DIT635 - Structural Testing Exercise

First, if you have not finished either activity from Lecture 8 (Structural Testing), do so!

In a previous exercise, you wrote unit tests for a meeting planner system. Those were based on your ideas regarding functionality. Now, we want to finish the job with test cases based on the code structure. The goal for your unit tests is to achieve 100% statement coverage (also known as line coverage) of the code for all classes except Main.

1. Measure coverage of your current test cases for the meeting planner classes with a tool, such as Emma, Cobertura, or JaCoCo.
 - a. If you are using Eclipse, we recommend EclEmma: <https://www.eclEmma.org/>
 - b. If you are using IntelliJ, you can use the bundled IntelliJ IDEA code coverage runner: <https://www.jetbrains.com/help/idea/code-coverage.html>
2. If portions of code are not already covered, add additional unit tests targeting that code.
3. If you find portions of the code that cannot be covered by a test case, explain why those elements are not covered.

(Note - it is possible to test Main by simulating keyboard input. However, you do not need to do so for this activity)

If you finish early, try fleshing out your unit tests for Homework Assignment 1 with those that add additional statement coverage.