CSCE 740 - Project 5 - Testing

Due Date: Tuesday, December 6, 11:59 PM (via Moodle)
No due date extensions will be granted for this assignment, as grades must be submitted on time.

Overview

We've built MEAT. Time to kick back with one of those little drinks with an umbrella in it.

Wait! We should probably test it a little before it ships.

The Assignment

You will create executable unit tests for your MEAT implementation and measure the code coverage achieved by your tests. You will design tests intended to achieve 100% coverage over the code in each class in your MEAT system. Tests will instantiate objects, provide input to their methods, and will verify that the output matches your expected output.

You may find inspiration from your requirements-based tests (from Assignment 2), but may also design tests from the ground-up to reflect changes in your system design since that point. Remember, you will need specific, concrete inputs as well as concrete expected outputs for each test, and you will need to design such tests as unit tests (i.e., tests for a class and its methods). Each test case must be complete, covering all details (setup, test steps, input, and expected output).

Test case implementations should use descriptive names, both for the classes and the methods you implement. We do not want to see test1(), runTest4(), etc. Implementation of test cases must follow the coding standards from the previous assignment (see the assignment 4 description for details).

You will need to have completed homework 4 to complete this assignment. If it is incomplete, finish it now.

Tools

Completing this project requires writing tests in the **JUnit testing framework**, and measuring coverage using **Emma** (you may use the eclEmma plug-in for Eclipse, or run Emma directly from the command-line).

We will not provide tech support for tools. Read the documentation and ask questions of your peers on the Moodle forum.

Deliverables

You are responsible for delivering the following as a single zip via Moodle:

- Unit tests, written as executable JUnit code.
    - If updated, also submit your revised project 4 code.
    - Tests should not be mixed with project code, but should be in their own package.
- README document describing any additional information needed to run tests (i.e., if you used any additional tools or libraries, such as mocking frameworks)
- Test execution report, including:
    - A list of which tests passed and failed.
    - For any tests that failed, note why they failed and what changes were required to fix that code.
- Coverage report
    - You must achieve, at minimum, 80% statement coverage on all classes.
    - If coverage cannot reach 80%, you must justify why it is not possible.