CS362-001 W17

Final Project: **Due date is Monday, March. 20th (23:59 pm)**
Late Submission: **No late submissions** will be accepted for the final project.

Part-A:

Overview

How the mutation testing tool works:

1. First, the mutations (i.e., small changes) are applied to the source code (i.e., dominion code)
2. Then, the modified code is compiled
3. Finally, the unit tests are run for each mutation and data is analyzed

In the first and second assignments you created manually and automatically unit tests. The effectiveness of these tests was measured using code coverage criteria such as statement and branch coverage.

1. You are asked to select, download, install and use a Java mutation tool to create mutations of the dominion code.
2. Use the tool to compute the mutation rate for: (a) your own unit tests; (b) a random test generation tool unit tests; OR a search-based software testing tool unit tests.

NOTE: If the mutation rate is under 100% (i.e., not all mutants of the code were killed) for your unit tests, add new unit tests to improve the mutation rate and try to achieve a score that is at least close to 100%.

Part-B:

1. You are asked to pick and test a classmate with a working Dominion implementation.
2. You can use the unit tests, random tests, or the search-based tests (both from assignment-1 and assignment-2), OR manually/automatically generate new unit tests to test code of the classmate you picked.
3. You **must** find and document at least two bugs and produce written bug report (possible including pointers/pictures to code to expose those bugs). The bug report should include the symptom, cause, how found , and how to fix it. In addition, include the <onid-id> of the student you picked to test the code. Submit the bug report to the canvas, titled **BugsReprot.pdf**.

3. **Submitting your solution**

1. **Canvas**: A document "**FinalProject_CS362_001.pdf**" in Canvas and contains **TWO** sections.

   Section Part-A:

   i. The section contains: (a) the mutation rate for each test suite (i.e., your unit tests, random testing **OR** search-based testing generated test suits(s)); (b) commentary on relative scores of the two mutation rate.
   ii. List the tools you looked at before selecting this tool?
   iii. Why this tool and cite sources?
   iv. What problems did you face while using this tool (If any)?
   v. Discuss your overall experience with using the mutation tool.

   Section Part-B:

   vi. The section contains: (a) describing your experience testing Dominion; (b) Document in detail, including code coverage information, the status and your view of the reliability of the Dominion code of your classmate (i.e., the dominion code that you picked to test)**.**
   vii. What problems did you face while generating tests to test your classmate code (If any)?
   viii. What problem did you face to test your classmate code (if any)?

2. **The class github repository**

   1. Submit your complete maven project for **Part-A (**i.e., your new tests for the mutation testing) to your onid-folder on our class github. The common maven project should contain:

   i. src/main/java package

   ii. src/test/java package

   iii. pom.xml