COMS 417: Software Testing (Fall 2017)

LAB 3

White Box Tests and Code Coverage

Due October 17, 12:30 pm

Objective and Outline

In this lab you will use the EclEmma code coverage tool. You will use coverage results from EclEmma to guide your selection of test cases while building a white-box test set for the countOs() method from Lab 2. This time, however, you will have the source code for the method. The Lab involves these steps:

1) Install coverage tool EclEmma;
2) Replace the “counter.jar” used in lab2 with the source code provided;
3) Run code and conditional coverage on the Part II test suite which you turned in with lab2 (capture the results in a screen shot);
4) Reusing your Parameterized test class from Lab 2 (see the related instructions below), and a new test data file with only the test case “O 1”, rerun the test, paying attention *only* to conditional statements that do not show full coverage.
5) Based *only* on the "missing" CONDITIONAL coverage in step 4, select more tests, add them to the test data file, and repeat until your test suite covers completely as many of the conditional statements as you can. When finished, record the final coverage in a second screen capture.
6) Now write separate non-parameterized tests (in a separate test class) for any requirements that have not been covered by your Parameterized tests. If you wrote such tests for Lab 2, Part II, you may reuse them here. Name and document these tests so that it is clear to us what fault you mean for the test to expose.
7) Submit your documented results. NOTE!! There are specific documentation and submission requirements in the following instructions. READ and FOLLOW.

General Instructions

EclEmma is a code coverage plug-in for Eclipse. The plugin can be easily installed from its update site at http://update.eclemma.org/ on any Eclipse installation of version 3.5 or above. If EclEmma is correctly installed, your workbench should include a button for launching in EclEmma coverage mode.

Note: If this button is not visible, select **Window>Perspective>Customize Perspective**, expand the "**Launch**" item, check "**Coverage**", and click **OK**.

For more information on how to use EclEmma please see the user guide
http://www.eclemma.org/userdoc/

To “replace” the “counter.jar”,

- select project>properties>build path>libraries;
- in the list of libraries, select the Counter.jar line; then
- click the “Remove” button and close out of the dialog.

With the jar removed, Eclipse should complain that class Counter is not defined. At this point if you copy the Counter.java file into the ….lab2.part1 package, everything should compile properly.

The source for Counter.java is available in the assignment section of BlackBoard.

Documentation Instructions

Create a top-level “Doc3” folder within your project. Place all the documentation described in this section in that folder and submit the project (minus standard libraries, bin directories, and class files) as a zip file (see below).

Place the screen captures from above in the Doc3 folder.

Create a report that answer the following questions:

1. Were you able to find more bugs using the conditional coverage? Why?
2. What kind of “conditional coverage” does EclEmma report?
3. Would an EclEmma report showing 100% condition coverage satisfy an FAA requirement for MC/DC? Why or why not?
4. Were there lines you were unable to execute using only the Parameterized tests? Why?
5. Which do they think would be a more time-consuming strategy for producing a white box test set for countOs();
    a. using conditional coverage guidance to identify an adequate test set, or
    b. using CFG analysis to identify an adequate test set?
6. Which of the above two alternatives do you think would work best to select tests for a 100,000 line piece of software? Why?
7. What are the limitations or weaknesses of test sets developed using only branch coverage? (think about which bugs you didn't find using conditional coverage and about what conditionals might look like in more complicated software.)

Submission Instructions

You should submit your complete project (minus bin directory and standard jars) as a zip file, with file hierarchy. We will take off points if we need to reorganize files or supply missing files or deal with unnecessarily large zips. Name the zip “<last name>_<first name>_417lab3.zip”. Submit via BlackBoard.