EECS 4313 Assignment 3

- Data Flow Testing, Slice-Based Testing and Mutation Testing

Due: **11:59 pm, April 5, 2018 (Tentative)**

---

1. Assignment goals

The purpose of this assignment is to give you experience on applying data flow testing, slice-based testing and mutation testing approaches. In addition, you will also have a chance to evaluate the goodness of existing test cases. Your task will be to create a test suite in JUnit, produce bug reports (if any), and submit a written report describing your testing and evaluation results. Finally, you will work on some tasks related to executing and analyzing load tests.

2. Task 1 – Borg Calendar

2.1 Getting started

Using the development environment you created for Assignment 2, complete the following tasks:

1. Perform data flow analysis for one of the methods that you have tested. Calculate the value of the following coverage criteria:
   - All-Defs
   - All-Uses
   - All-P-Uses / Some-C-Uses
   - All-C-Uses / Some-P-Uses

   If necessary, add test cases to bring these values as close to 100% as possible.
2. Derive the slices for all A-defs (defined by assignment) and P-uses (used in a decision predicate) in the method. Calculate the percentage of slices that your testing covers. If necessary, add test cases to bring the percentage as close to 100% as possible.
3. Install the PIT mutation testing tool. Evaluate the effectiveness of all your test cases (test cases from A2 and newly created test cases for this assignment) using PIT. Add or modify test cases, accordingly.
4. If you believe that the additional test cases have revealed new bugs, attach bug reports in the written report (see section below for details).

2.2 Code Structure and Testing

We will conduct automated testing to mark your testing code. Make sure you place all the white-box testing Java code that you have created in a folder called eecs4313a3t1. *There should be no sub-folders inside eecs4313a3t1!*

The compilation and the running processes are very similar to the one described in Assignment 2. You should use the following naming conventions for developing and packing your testing code:

- The code package should be called "*eecs4313a3t1*"
- Name the test suite as *EECS4313A3AllTests.java*

2.3 Written Report

You should create a written report (a3.pdf) describing the test cases and test results that you have done. The report must include the following information:

- The data flow analysis you performed and the calculation of the coverage metrics. You must show which test cases are responsible for which dc-paths.
- A description of the test cases you added to improve coverage. If your coverage was already high, discuss how your testing was able to achieve this.
- The slices that you identified and the percentage of slices that your testing covers. You must show which test cases are responsible for which slices.
- A description of the test cases you added to improve slice coverage. If your coverage was already high, discuss how your testing was able to achieve this.
- Evaluate the effectiveness of your test cases using mutation testing. Discuss and address any issues if you have found in your written report.
- Attaching bug reports if bugs are discovered using your testing methods. You should use the same bug report format as in Assignment 1. **Do not** file these bug reports to the project's bug report system.
- An appendix with the specification of the methods you are testing.

Presenting your thought processes in English is an important skill for a software engineer. If you have trouble in writing English, ask someone to proof-read and correct your writing. You can also consult the English as a Second Language Open Learning Centre (http://www.yorku.ca/eslolc) at the York University.

3. Task 2 – JPetStore

3.1 Getting started

In this task, you are required to conduct load test on the performance of an e-commerce website, JPetStore using JMeter:

1. Download the JPetStore deployment war files at: http://www.cse.yorku.ca/~zmjiang/teaching/eecs4313/assignments/a3/jpetstore.war.
2. Deploy the application by dropping the war files at Tomcat's "webapps\ROOT" folder and double check if you can access the JPetStore website (http://localhost/jpetstore/shop/index.shtml).
3. Explore the system a bit and define some realistic, non-trivial test scenarios.
4. Leverage the record-and-replay features to encode the test scenarios in JMeter.
5. Load test the system for a period of time (5 – 30 minutes).
    - Since this task is for you to gain exposure on load testing, please pick some lower request rate (e.g., 1 request/second) and do not overwhelm the system with extremely high load.
6. During the test execution, make sure you also record the performance counters and the logs.
7. Analyze the results of the load test to see if there is anything abnormal:
    - Did the system crash/restart/hang during the test?

- Did all the requests go through?
- Is there unexpected resource usage during the test?

3.2 Written Report

In the second part of your written report (a3.pdf), you should describe the test cases and test results that you have done. The report must include the following information:

- The test scenarios that you have created;
- The request rates and the duration of the load tests;
- The analysis of your load tests and the description of any problems that you have found (if there are any).

4. Submission

You must work in a team with ~~**at least three and at most five**~~ **at most five** partners. *Failure to form a team or a team with the right sizes will result in loss of marks!* **Only 1 submission per group, please!** You should submit the following:

1. Submit a PDF of your written report (a3.pdf). Your report must include the names and the student numbers of all the team members.
2. Submit the test code package (i.e., eecs4313a3t1) for task 1.
3. Submit the testing artifacts (i.e., eecs4313a3t2) for task2, electronically. It should only contain the JMeter test plan (JMX file) and a sub-folder (call "data") containing the recorded logs and counters used for your analysis.

Navigate to the directory where it contains the code package, test artifacts, and the report. Use the following commands to submit:

- submit 4313 a3 eecs4313a3t1
- submit 4313 a3 eecs4313a3t2
- submit 4313 a3 a3.pdf