

Assignment 3

CSE 4313 3.0 Software Engineering Testing, Winter 2013

Due: Monday, March 4, 2:30pm.

Format: Individual.

Testing with JUnit

The purpose of this assignment is to give you experience in creating automated structural test code with JUnit. Your task will be to create a test suite in JUnit for the non-GUI portion of BORG Calendar, produce bug reports, and submit a written report describing your testing.

What to do

Create a development environment similar to the one for Assignment 2.

Select at least two different non-trivial classes for your testing, so that you can demonstrate at least one example of using dataflow testing and path testing (you should not select classes with already developed test cases).

Define a detailed specification for every selected class based on your understanding of the purpose of the class (an example is shown on Page 3).

Develop a test suite for the selected classes. You may use any testing strategy you believe is appropriate including the ones discussed in class. You must demonstrate examples of using dataflow and path testing. You are free to create as many test classes as you like.

For each implemented test case, develop a formal description of the test case. You should also provide a justification for the developed test case. Both the test case description and test case justification will be a part of your report.

If you believe that the implementation fails to meet the specification in some way, create bug reports. The bug reports will be part of your written report, so you can create them in whatever format you prefer.

What to Submit

Before the deadline, submit electronically the test code package you created. To submit, navigate to the directory that contains the `working` directory (e.g. `winter2013`), and give a command like the following:

```
submit 4313 a3 winter2013
```

You also need to submit a written report. This report must include:

- The specification of the selected classes
- A description of the testing strategies you applied, as well as the test cases you created. The marker will not read your code in order to see what you tested. You have to describe it.
- Give one detailed example of path testing and dataflow testing for one of the methods. Construct the DD-Path graph for the selected method. Demonstrate McCabe's Basis Path method and estimate complexity. Demonstrate an example of du-Path and provide detailed steps of using slice-based testing.
- The bug reports you created.

Submit a hard copy of the written report to the 4313 drop box by the deadline. Attach the last page of this handout as the first page of the hard copy submission. Fill in your name and student number. Submit the same report electronically before the deadline as well (PDF format is preferable, but other formats will also be accepted). To submit, give:

```
submit 4313 a3 a3.pdf
```

where `a3.pdf` is your report. The marker will mark the hard copy submission. Electronic submissions will be used for remarking purposes.

Grading

This assignment will be marked out of 100. The marks will be divided as follows:

- Bug finding power of your testing: 20%
- Coverage of your testing: 20%
- Description of your testing approach: 40%
- Quality of bug reports: 10%
- Presentation/English: 10%

Presenting your thought processes in English is an important skill for a software professional. If you have trouble writing English, have somebody proof-read and correct your prose. You might find the services offered by The English as a Second Language Open Learning Centre useful: <http://www.yorku.ca/eslclc>

Following is an example of specification for a Java class (not related to the BORG calendar project):

The GraphGenerator class contains the following two methods:

1. `public GraphGenerator (int numberOfNode, int maxDegree, boolean isConnected, String outputFileName)`

The first argument is the number of nodes that the generated graph should have. The second argument is the maximum degree for any node.

If the boolean argument is true, then the produced graph must be connected in the non-directed sense. If the boolean argument is false, then the graph may or may not be connected.

Finally, the String argument is the name of the file that will contain the produced output in the XML format.

2. `public void createGraph() throws Exception`

This method generates the random graph and outputs it in the designated file.

If the number of nodes or the maximum degree is less than 1, then a `NegativeInputException` is thrown.

Assignment 3 Grade Sheet

CSE 4313 3.0 Software Engineering Testing, Section M, Winter 2013

Student Name:

Student Number:

Bug finding power

Coverage

Description of testing approach

Quality of bug reports

Presentation / English

Total

Letter grade