Assignment HW5

**Cover Page**

Prepared for:

Dr. Mehra Borazjany

Trung Hieu Tran

Prepared by:

Alex Lundin

SE-4367.0U1-Testing

**Assignment Choice:**

N/A for this assignment

24 June, 2018

**Proof of Working Software**

GitHub link:

<https://github.com/AlexLundinEducational/SE-4367-Testing>

Branch Summary:

master – managed by Alex, only fully completed pulls allowed to make TA’s life easy. Master only contains assignment material once they reach completed status.

working – flexible branch for team, ideally, this material should build without causing technical debt during the project.

Commit for grading:

HW5\_Alex-Lundin Complete, Ready for Merge to master and Ready for Grading

Phase 1 Development

Phase 2 Development

Proof of Working Tests (10 seconds)

<https://autode.sk/2taZEfw>

**Documentation Log**

1. Reductions
   1. File 1

* Removed a few import statements in the FactServlet.java that caused compilation errors

1. Refactors
   1. StringConstants.java

* this is a refactoring of the strings used in the JavaServlet.java file

1. Modifications
   1. testRemoveDuplicate()

* Modified the original source code and added toString() method to set an int value of the array before and after removing all duplicates

1. Additions
   1. FactListTest.java

* Used to test the search method in FactList.java
* setup method to create the list before All Tests
* 4 test methods
* Covers all return statements in FactList.java search method

Consider the given *DaysDurationCalculator.java* program, Design and develop at least 10 JUnit tests to evaluate the *cal*().

Do **not** develop tests for the *main*() and the *getN*() methods.

Since *cal*() has a precondition that excludes almost all invalid inputs, you should focus on normal behavior. Note that “data-driven” tests could be very useful for this assignment.

1. A. Create at least one possible fault that you could place into the *cal*() method that your tests **do not** find.
2. B. Create at least one possible fault that you could place into the *cal*() method that your tests **do** find.

Submit the following on paper:

1. A brief write-up (about one page) that describes your test set and the faults.

2. A printout of your JUnit tests.

3. Screen shots showing the results of running all your tests.

**Grading**

We will grade on several factors.

* • (10 pts) Whether you have the required number of tests
* • (50 pts) The overall quality of the test set
* • (10 pts) The clarity of your write-up
* • (20 pts) The two faults that you create
* • (10 pts) Whether your tests ran