

COMPSCI 230 Assignment 3
2015 S1
Marking schedule

Part 1:

Open the document containing the student's Test Designs. Instruction is for this to be .pdf, but the student may submit .ods or .doc. Open the document containing some test design example answers **compsci230Ass3(TestDesignExampleAnswers)**. You will see some test cases in the left column and some tests for each test case in the right column.

Students are to be awarded up to 1 mark for each test identified from the example answers, up to a maximum of 15 marks. For example, the first test case in the example answers document has 3 tests and so would get 3 marks. However, to be awarded any marks for a test case (left column), it must represent a boundary value situation (see below).

If some tests are incorrect, or are awarded .5 mark, you should keep marking until 15 marks have been achieved (or there are no more tests to mark). Once you have reached 15, you should ignore remaining tests.

Tests do NOT have to be in the order shown. We're interested only in the test cases and appropriate tests being identified.

If students identify test cases not in the sample answers, please use your judgement, or email me (dkir006@aucklanduni.ac.nz).

Marking summary :

- For a stated set of inputs (left column), 1 mark for each correct test in the *right* column.
- For a stated set of inputs, .5 mark for a test in the *right* column that is 'nearly ok' (see below).
- For a stated set of inputs, 0 mark for a test in the *right* column that is not in my sample answers (or contact me if unsure).

Some guidelines :

A test case is a combination of *Method inputs* that represent boundary values. For example, we're looking for adding devices to last position in list, to full list; invalid device IDs (too long, too short, doesn't represent a number, represents a number but isn't all digits); bad values for integer inputs and combinations of integer inputs (but all boundary values).

This means that test cases that e.g. add a device to the middle of the list, test a negative range where the values are not contiguous or 0 (eg 38, 22) are awarded 0 marks.

In the example below, I've identified 3 tests for the test case 'Add device with ID too many digits'.

	Method inputs	Tests
3 marks	Add a device with too many digits in ID ("13119")	1. IllegalArgumentException thrown 2. List of devices contains 0 devices 3. Device list does not contain ID "13119"

You should award 1 mark for each of the 3 tests identified. If the student has identified only 1 test, then 1 mark should be awarded. If the student has identified more tests for this set of inputs, please use your judgement, or contact me.

- If the student states that an exception will be thrown for a specific set of inputs, but the wrong exception is stated, award .5 mark.
- If the wording the student uses to describe the state of a list of devices is not really clear, award .5 mark.
- If student gives input values that are ok but mid-range (e.g. device "2344" to the middle of the list), award 0 marks.

You will have to use your judgement on the wording for the tests. Hopefully it will be clear what the student was trying to say. You should err on the side of the student.

Part 2:

View the *ControllerTest* class in the student's submission. You do not need to run tests, and so you can open the source file in Eclipse or in any other editor.

The task is to check the student's JUnit implementation for the tests that you marked in their design document. Students have been asked to present JUnit test methods in the same order as in the design document. This means that you should find the tests you are looking for by starting at the top of the JUnit class, and working down until you have marked all methods corresponding to the 15 marks awarded in the design.

Ignore tests that do not appear in the design document. A single test method may relate to a single test, or the student may group tests for a single set of inputs in the same test method. For example, a suitable method to test if a valid device is successfully added to an empty list would be as in the first test in my JUnit tests. As all three tests correspond to my design document, I would be awarded 3 marks for this.

Some students have used JUnit 3 and some have used JUnit 4 and so you will find some syntax differences. Please ignore these – you should award the mark if the test clearly aims to implement the design.

You should award 1 mark for each implemented test. A sample test class is included for reference. However, you should base your marking on consistency with the student's design document.

Once 15 marks have been awarded, ignore remaining tests. The students were encouraged to implement more tests than required to support their efforts to find defects.

Part 3:

Check the *Controller* class in the student's submission. The five seeded defects are described in the document **compsci230Ass3(InjectedBugs)**. For each of these, award:

- 1 mark if the student has identified the defect and made an effort to resolve it
- .5 mark for including the change in the javaDocs at the top of the file
- .5 marks if you believe the change is of 'high quality' i.e. code and javadocs are tidy (you should use your judgement on this).

Below is an example of what is expected in *Controller* class header.

```
* @version 1.1 : May 20 2015
*   Two defects addressed:
*   1. Add ! to 'deviceID.matches()' to ensure exception is thrown if ID doesn't match mask.
*   2. Add check for list already full when adding device.
```