

CSC207: Unit Testing

To earn the credit for this lab, you must be present in the lab class.

1 Overview

This week, you are going to implement a JUnit test suite.

2 Log in and get things set up.

For this lab there is no submission. You must complete the work and show it to your TA and get a grade on the spot. Please note you may be in a room different from other labs, due to Phase I presentation. Team up with a student sitting next to you and divide roles into driver and navigator. Log in and get the starter code from the course web site.

s1 drives and s2 navigates.

Now s1 logs in and opens a new terminal window.

1. Open up Eclipse. Point your workspace to the folder Lab8. You may have to create it first.
2. Using Eclipse, create a new Java project called Lab8. Make sure you open the Java perspective.
3. Create a new package called lab8.

3 The FlightDataRecorder class.

Commercial aircraft are required to contain a flight data recorder (FDR), colloquially known as a black box, for recording data such as aircraft altitude and velocity. An FDR can store up to 25 hours worth of data; to allow continuous recording, FDRs record on a loop; this means that for a recorder that can store 25 hours worth of data, at the beginning of the 26th (51st, 76th, etc.) hour of operation, the recorder is overwriting the data from the 1st (26th, 51st, etc.) hour of operations. The FDR code in FlightDataRecorder you will be testing in this lab only keeps the last 10 entries of data.

4 Chosing test cases.

- No driver/navigator. Brainstorm with your lab partner.
- View the Javadocs posted on the course website for the FlightDataRecorder class.
- Create a list of test cases by creating a table similar to the one below.

Purpose of test	Expected Behavior
Check the behaviour of <code>average()</code> on a new <code>FlightDataRecorder</code> .	<code>average()</code> returns 0
...	...

5 Writing unit tests

s2 drives and s1 navigates.

- Select `FlightDataRecorder.java` in the Eclipse Package Explorer pane on the lefthand side of the screen.
- From the File menu, select New → JUnit Test Case. Note that Eclipse has named the class for you.
- If Eclipse warns you that JUnit 4 is not on your build path, let it add it for you.
- Write at least one test method for each test case in the table. Remember to follow naming conventions by prefixing the name of each test method with `test` and to include the `@Test` annotation before the method definition. Do not forget to document your test methods. Switch roles halfway through writing your test methods.
- Click run. If Eclipse asks you which launcher you would like to use, click "Use configuration specific settings" and select the Eclipse JUnit Launcher. Hopefully, you will have discovered some errors.
- Click on individual test cases in the JUnit pane on the left of the screen for more details about each test case.
- Most of the methods in the provided code contain bugs.
- Identify at least three bugs. The lab is marked out of three points, one for each dicovered bug.
- Show your work to your TA.