SOFTWARE TESTING: Project

This page describes the practical for the Informatics Software Testing course. It will be marked out of 25 points of the assessment of the course. This project will be undertaken in groups of 2, and will be assessed on the basis of the group submission.

DEADLINE

The project is comprised of 1 task with the following issue date and deadline,

Issued: 3 April 2021.

Deadline : 5 May, 11 pm.

BACKGROUND

In this project you will consider the specification and Java program available at the following github repository.

TOOLS

You can choose either to use the Eclipse IDE or just to use JUnit and other tools standalone; I have no strong preference - many people find the tools available in Eclipse useful (if you haven't used Eclipse before maybe now is the time to give it a try). You will need some of the following:

1. If necessary you can download JUnit from here . If you are using Eclipse it is probably already installed in the IDE. This article is a reasonable introduction to using JUnit with Eclipse, but bear in mind its age: in particular it's focused on JUnit 3. Here's a good introduction to JUnit 4 (free registration required).

SETTING UP

Preparation:

Please use JUnit 4 as your testing environment.

TASKS

TASK 1: FUNCTIONAL TESTING (25 MARKS)

In this task you will implement JUnit tests using the specification provided in the Github repository. The repository also provides the implementation as a JAR file, `ST_Coursework.jar` , so you can execute your JUnit tests and observe test results.

The specification is described in detail, with helpful examples where necessary in the `Specifications.pdf` file.

Functional testing is a black box testing technique, so use the specification file to derive tests and **not** the source code. The jar file under the `jar` directory can be used to execute the tests derived from the specification. We have also provided a sample JUnit test case, `CommandLineParserTest.java` file, to illustrate a typical test case for the implementation in `ST_Coursework.jar` .

All the files referred to above can be found at the Github repository.

In giving a grade for this part of the practical I will take into account the performance of your test set on a collection of variants/mutants of the specification.

Deliverables:

1. A file `Task1_Functional.java` that contains your JUnit tests.