

## Term Project

**Goal:** To choose a class with at least 5 methods from the Java language code base (see <http://openjdk.java.net/>) and build an automated testing harness to test this class. Test cases will be developed from various functional and structural criteria we have addressed in the course. The testing harness will consist of a directory structure and script(s) that will be executed from the command line. You may work individually or in teams.

**Deliverable #1:** Produce a test plan (see [here](#)) that details how you will proceed. In addition, these are topics to include: choice of class with listing or link; specific functional and structural coverage criteria and their application; specifics of how test cases are generated to comply with these criteria; a template for recording each test case (see below); the test cases themselves according to the template (i.e., a set of .txt files).

Template should include instructions for creating a test case file that conforms to the template:

1. test number or ID
2. requirement(s) being tested
3. testing target
4. the relative path to the file your script will execute to perform the test case
5. the input(s) for the test case or a path to the file that contains the inputs
6. the expected outcome for the test case or a path to a file containing the oracle

In future deliverables, expect that your script will process one or many test case files, run the tests, collect the results, and produce an html report that opens in a browser.

Submit your work as a single PDF file: [lastName]\_Deliverable1.pdf by 5PM 3.April.2012.

(continued on next page)

**Deliverable #2:** Build the following folder structure and make sure that all your directory references within scripts, etc. are relative to this structure (be sure not to use spaces in any folder or file name). All resources needed should be included within this structure, excepting the Java language libraries needed to compile.

```
/[firstLast in CamelCase]_TESTING
    /project
        /src
        /bin
        / ...
    /scripts
        runAllTests.(some scripting extension)
        ... other helper scripts
    /testCases
        testCase1.txt
        testCase2.txt
        ...
    /testCasesSrc
        testCase1(may be folder or file)
        ...
    /testCasesExecutables
        testCase1(may be folder or file)
        ...
    /testCasesInputs
        testCase1Input (may be folder or file)
        ...
    /temp (for output from running tests ... be sure to clean at start of runAllTests)
        testCase1results (might be folder or file)
    /oracles
        testCase1Oracle (might be folder or file)
    /docs
        README.txt
    /reports
        testReport.(txt or html)
```

### Requirements:

1. From within the top-level folder, the command to start your framework must be `./scripts/runAllTests.(some scripting extension)`. This script will walk the `/testCases` directory and execute each test case found and capture the required information needed for the testReport file.
2. Your `runAllTests` script must open and display your testReport automatically either in a browser or as a text file. The test report should provide an informational summary of the results of running all the test cases in `/testCases`.
3. Create all these folders even if you do not use them; create additional folders as needed.

Submit your work as a single zip of this folder structure and files named `[firstLast in CamelCase]_TESTING.zip`. You will need to email it to me as `[firstLast in CamelCase]_TESTING.zip.piz`.

Due 5 PM 10 April 2012.

Come to class prepared to demonstrate your project.