Guide to GitLab Testing

This is a guide to implementing project/lab testing for CS2, 3, and 24 at Caltech. These courses utilize the Caltech CS * Automated Test Framework , and it may be useful to become familiar with this library.

Step 1: Create the Root Project

Before implementing any testing, create the root directory that will be cloned by the students when they begin the assignment. This includes making sure that all the correct files are present (source code referenced in the spec, test files etc.) and that the tests are configured locally. Once these are done, set up your IntelliJ session as if you were a student starting the project for the first time (usually just have all files closed). Push all of this, and make sure to include the *workspace.xml* file (you can use the *-f* flag), as this captures all the local modifications you've made.

↳ edit Configs
↳ Store as a project file
↳ push test xmls.

Make sure the repository contains:

- A .gitlab-ci.yml file: this specifies the tests to be run.
- A .gitignore file: feel free to copy this over from a past assignment

Step 2: Update the Tests

Locally, build the project. IntelliJ will create an *out* folder (make sure you do not push this), which contains all the *.class* files. Find the files corresponding to the tests and copy those over into the Automated Testing Repository.

Create a merge request from *master* to *prod*. This is the only way to get the runners working.

Step 3: Test… the tests

Create an empty private reference solution repository.

- Go to Settings → CI/CD → Runners
- To test the *prod* branch tests, enable the oobleck-unsecured runner

The tests should run the way you expect them to. Make sure they run appropriately on the source code, before you try to test a pass. Once that looks good, push a solution and make sure everything passes!