# Exercise 5

Using github, make a fork of this week's repository so you can make changes to it in your own workspace. The repository is:

```
https://github.com/dlambrig/Continuous-Delivery-with-Docker-and-Jenkins-Second-Edition.git
```

1

Run the jacocoTestCoverageVerification test in your pipeline. The pipeline will fail with the error:

```
> Rule violated for bundle calculator: instructions covered ratio is 0.0, but expected minimum is
0.2
```

Describe in your own words the meaning of this error and how you can fix it. Then, modify the minimum limit as specified in file **build.gradle** so that the test will succeed. Commit your change to your forked repository. Show Jenkins logs running the pipeline before and after your change.

2

Add a stage to the lab1 pipeline to run the **jacoco checkstyle** test. Add a publishHTML section to generate a report named "**jacoco checkstyle**". The github repo will fail. Take a screenshot of the generated report to show what failed and describe the failure.

Modify the java file to correct the failure and commit the change to your fork. Then rerun the test, and show it passes with a screen shot.

Submit your pipeline and Jenkins screenshots.

3

Modify the checkstyle configuration config/checkstyle/checkstyle.xml.

Add the rule "ConstantName". You can find documentation here. Commit it to your forked repo.

Run the checkstyle test in the pipeline. It will fail. Take a screenshot of the report generated.

Fix the mistake and commit it to your fork. Rerun and show it passes.

4

Many shops have replaced dedicated testing employees (QA engineers) with employees who do devOps, where most tests are automated within a pipeline. Why? What are advantages and disadvantages to making this change?

5

A unit test checks a given piece of code. Suppose code coverage is very high such that almost all of the code is checked this way. Given high coverage, is this sufficient testing? Why or why not? What else would ensure high quality?