CSCI 3901 Winter 2021

Lab 6: Managing software

Due Friday Feb. 12th, 08:30 AST in Brightspace

Objective

In this lab, you will explore managing a system of software with test suites via JUnit and build management with Make.

Working alone or in a group of 2, you will apply your knowledge of JUnit and Makefiles to write a test script for the newest release of the `HfxDonairExpress` online ordering system, to improve their build automation, and to include your test cases in their existing testing setup.

Preparation

- Download and install `GNU Make`. You can alternatively use it on `timberlea`.

Procedure

Set-up

1. Download and unzip the source directory posted on Brightspace.

Lab steps

Part 1 - Unit Testing Write a JUnit test suite for `HfxDonairExpress`, with **one test method for each** of the following cases. You should put these test cases in a file called `UnitTests.java`. Your test cases should check that the method in question (either `order` or `makeOrder`) returns the correct value.

1. `order` method: Ordering a small donair
2. `order` method: Ordering a large donair with the coupon code "YOLO5"
3. `order` method: Ordering a medium pizza with pepperoni and mushrooms
4. `makeOrder` method: Making 1 order
5. `makeOrder` method: Making 2 orders in a row

Part 2 - Understanding Makefiles Answer the following questions about the provided `Makefile`:

1. Which file(s) are included in the `SRC` variable?
2. What file(s) does the `all` target depend on?
3. What is the command used to make the `hfxdonairexpress.jar` file?
4. What does the `test` target do? You may want to try running the command `make test` to see.

Part 3 - Writing Makefiles Add the following features to the existing `Makefile`:

1. Modify the `Makefile` so that your unit test class is compiled and included in `hfxdonairexpress.jar`
2. Modify the `Makefile` to put the compiled `.html` documentation files into a new sub-directory, `docs`
3. Modify the `Makefile` to include a target `clean` which deletes the compiled `.class` and `.html` files. Note: the `rm` command can be used to delete a file

Questions

1. Why is it important to have a portable, reproducible build system that isn't tied to a particular IDE?
2. How often should you run a class' test suite when developing a software system? Explain.
3. Can you see any reason to split test cases up into separate files, or should you just include all of your test cases for every class in the same file?

Reporting

1. In one file, list
   - The members of your team.
   - The answers to the questions in part 2.
   - The answers to the Questions section of the lab.
2. Generate a PDF from the document.
3. Submit the PDF, `UnitTests.java`, and you modified `Makefile` in Brightspace in the Lab/Lab 6 folder.

Assessment

The assessment will be on a letter grade and will reflect how well you have used JUnit and Make, as well as how your answers to the broadening questions demonstrate that you have thought through how unit tests and build automation can help you in maintaining and deploying a complicated piece of software.