

Programming Assignment: Writing Unit Tests for the Coffee Maker

✓ Passed · 100/100 points

Deadline The assignment was due on Apr 30, 11:59 PM PKT
You can still pass this assignment before the course ends.

[Instructions](#) [My submissions](#) [Discussions](#)

Task

In this assignment, you'll get some practice at building effective unit tests on a slightly larger example of a coffee maker. The coffee maker allows you to set up recipes, add ingredients, and to make one of several beverages, provided that you insert enough money. It is an often-used pedagogical example from our colleague [Laurie Williams](#) at NC State university.

Also, we have made .pdfs of the use cases, the class diagram, and a scenario for the Coffee Maker in the .zip file (Windows) and .tgz file (linux / mac), as well as below:



CoffeeMaker Example.pdf

PDF File



cm_class.pdf

PDF File



cm_seq.pdf

PDF File

Deliverable

Your task is to create a file, CoffeeMakerTest.java, which properly tests the CoffeeMaker class to ensure it is working properly. A slightly buggy version of the coffee maker is available here:



CoffeeMaker_JUnit_Assign

TGZ File



CoffeeMaker_JUnit_Assign

ZIP File

Deliverable

Your task is to create a file, `CoffeeMakerTest.java`, which properly tests the `CoffeeMaker` class to ensure it is working properly. A slightly buggy version of the coffee maker is available here:



CoffeeMaker_JUnit_Assign

TGZ File



CoffeeMaker_JUnit_Assign

ZIP File

And a "golden" version of the same model, in which all tests should pass is available here:



CoffeeMaker_JUnit_Golden

TGZ File



CoffeeMaker_JUnit_Golden

ZIP File

The .zip / .tgz file for the assignment contains an Eclipse project and a gradle script that contains the functional code for the coffee maker and a few unit tests to get you started. For directions in how to install gradle buildship support into Eclipse, see the "Test Doubles: Installing Gradle and Mockito" lecture.

And

Download and expand the .zip / .tgz file into a directory of your choice, then to import the project into Eclipse, follow the directions to import gradle projects into Eclipse:



Loading a Gradle project into Eclipse.pdf

PDF File

If you want to do things from the command line or from another IDE, you can use gradle directly starting from the `build.gradle` and `settings.gradle` file. To install gradle, follow the download and install directions from <https://gradle.org/> that are appropriate for your platform. However, in this case you are on your own.

Inside the project, you will find the functional code, a couple of unit tests to get you started. The goal is to construct a sufficient number of unit tests to find most of bugs in the "buggy" version of the coffee maker that is included. You should be able to detect at least 5 bugs in the code using your unit tests.

How will this be graded?

In order to measure the adequacy of your tests, we will run your tests against each of the bugs in the original model, and a fixed version of the program. The tests should pass on the fixed version and detect the bugs in the buggy version.

You will get half credit for submitting tests which accurately report a working correct implementation. Then, you get incremental credit for each buggy version your tests successfully identify as not correct. There are at least 5 bugs in the "buggy" program; you get full credit for finding 5 bugs.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submissions" tab.

