

Assignment #6

Practice with JUnit

CS 4354 Summer II 2016

Instructor: Jill Seaman

Due: before class **Monday, 8/8/2016** (upload electronic copy by 9:30am).

Write some tests using JUnit for the Library Management System. A preliminary (incomplete) version of the Library Management System is available on TRACS. Download and install the Library Management System into an eclipse project (or use another IDE, or the command line).

Include test cases for the following:

1. Test the **method Resource.calculateFine(Calendar)**. You should have at least 4 test cases, including the “exceptional cases”. `java.util.Calendar` is described below.
2. Test the **Collection class** (this is the list of Resources owned by the Library). For this assignment you need test only the `addResource(Resource)` method. You should include a successful and an unsuccessful add operation. Note that you are testing the collaboration of the `addResource` and `findResource` methods.
3. Test the **collaborating classes** `Member` and `Resource` as they carry out the **Resource.calculateNewDueDate()** operation. You should have 4 tests for this, one for each combination of a `Member` subclass and a `Resource` subclass.
4. (optional, will not be graded, but answer will be included in the solution)
Test the **Controller.checkout** operation. Be sure to test a successful checkout operation. Note: you do not need to test the `Calendar` date returned (this value is tested in question 3). Instead you should verify that the `Member` and `Resources` were changed appropriately.

java.util.Calendar:

The Library Management System uses this type to record the due date of the items, and the due date is calculated from a given start date. It also uses this type to calculate the value of the fines on overdue resources. You are encouraged to look it up in the online Java API, but this should be enough information to do the testing you need to do:

```
Calendar date = Calendar.getInstance(); // constructs a value equal to the current date
date.add(Calendar.DATE, 10);
// Adds (or subtracts, if negative) the specified number of days to the date object. It is a
// mutator. This example advances date to 10 days ahead.
```

NOTES:

- I updated the lms.zip file with a new version of the lms code. Please download the newer version for this assignment!! Resource.calculateNewDueDate() now returns a Calendar object (and does not modify the parameter).
- This assignment is to be done with your partner (in groups of 2).
- Use the package "assign6" for your classes and put your files in the appropriate directory structure. Submit ONLY your test case java files!! (your files may need to import lms.*;).
- Each test case must be in its own method. More specifically, for problems 1-3, **each @Test method may contain only ONE assert... method call!!**
- Make sure your test will fail if the code is broken. Just because your test passes does not mean it is a good test. After you have written your test, consider changing the code to introduce a bug that you think your code should catch.
- Follow the style guidelines from the class website. Javadoc comments are not required, but good comments describing each test method are required.

Submit:

Please combine your *.java files into a single zip file (assign6_XXXXXX_YYYYYY.zip). The XXXXXX and YYYYYY are your TX State NetIDs (mine is js236, you have two, one for each partner). Submit an **electronic copy**, using the Assignments tool on the TRACS website for this class.