

CS 455 Lab 5: PA2 milestone

Fall 2021 [Bono]

Goals and Background

There are two goals of this lab: (1) to help you manage your time in completing PA2 by completing a milestone this week and (2) to encourage you to use unit testing and incremental development in solving the problem.

Please use your PA2 workspace to complete this milestone. The Lab 5 workspace is only for DEN students to submit the lab: more information for DEN students at the end of this document.

Note: it is possible to complete your assignment without doing this milestone. However, this lab concerns a specific test program; *no* lab credit will be given for a completed PA2.

Because this involves code you will be turning in for a programming assignment **this lab must be completed individually** (i.e., not as part of a partnership) and because of the size of the problem you will ideally spend significant effort on it before your lab section (See Advanced Preparation section below for more details.)

The first lab exercise here is related to the assignment, but is not part of the milestone.

Advanced preparation

You will need to read over the whole programming assignment before starting the work for the milestone.

This milestone is likely too large to do start to finish during one lab period. You should have the milestone code pretty much written and tested before the lab period to get all the lab points: your lab period serves as a deadline for having this code working.

However, even if you don't have much chance to work on the assignment before the lab, you can use the lab session as a pre-scheduled 2-hour block of time to work on your assignment with TA help available. If you are behind, you should still be able to get Exercise 1 completed during the lab period.

Reading and reference material

- [Programming assignment 2](#) and these sections in particular:
 - "BookshelfKeeper class" (exercise 1 only)
 - "BookshelfKeeperProg program" (exercise 1 only)
 - "Bookshelf class"
 - "Representation/Implementation"
 - "Representation invariants" (exercise 5 only)
- Horstmann, Special Topic 11.6 covers assert statements
- CS 455 relevant lecture topics this week: preconditions, assert statements and invariants

Exercise 1 (1 checkoff point)

[Note: This is a pencil-and-paper exercise: DEN students should put their answers in the file `README.lab5` in the Lab 5 workspace. On-campus students should just write the answers in their lab notebook.]

For each of the following example of books on a shelf managed by a BookshelfKeeper, write down the minimum number of necessary calls to bookshelf mutator methods (addFront, addLast, removeFront, or removeLast) on that bookshelf to do the operation shown, and also show the updated bookshelf:

1. bookshelf: [1, 3, 4, 8, 11, 14, 15, 18]
operation: putHeight(13)
2. bookshelf: [1, 3, 4, 8, 11, 14, 15, 18]
operation: pickPos(2)

General instructions

To help avoid version control problems, there are no separate starter code files for this lab: **complete this lab in your Vocareum PA2 workspace**, unless you are using some outside IDE. There is a Lab 5 workspace, but it is solely for DEN students to submit their lab. This means DEN students will need to do a little extra at the end to submit it (details at the [end](#) of the lab).

For the rest of the lab exercises described below you'll be writing a unit test and implementation of your Bookshelf class. No lab credit will be awarded for showing us code for a Bookshelf class without the accompanying unit tester, BookshelfTester.java, that demonstrates that your code works.

This unit tester should have the structure of others we have seen and written for this course: it should use hard-coded data (no user input), should show expected as well as actual results, test various combinations of calling the methods, and test for general and edge cases.

Exercise 2 (1 checkoff point)

Write a BookshelfTester.java program that uses hard-coded data to test the following subset of your Bookshelf class: the two constructors plus the toString method. Demonstrate to the TA that these methods work, including explaining your tests.

Exercise 3 (1 checkoff point)

First make a backup of your Ex 2 version of the tester. Add test cases to BookshelfTester to test four more of your public Bookshelf methods. (You can choose which ones.) Add the implementations of these methods to the Bookshelf class. Demonstrate to the TA that these additional methods work, including explaining your tests.

Exercise 4 (1 checkoff point)

First make a backup of your Ex 3 version of the tester. Add test cases to BookshelfTester to test the remaining three of your public Bookshelf methods. Add the implementations of these methods to the Bookshelf class. Demonstrate to the TA that these additional methods work, including explaining your tests.

Exercise 5 (1 checkoff point)

Part 1. For the Bookshelf class write (a) your representation invariant comment, (b) your private isValidBookshelf method, and (c) the calls to isValidBookshelf from your two constructors. Now write a small program called TestAssert.java that calls the one-argument Bookshelf constructor two times: one time such that it satisfies its precondition, and one time such that it doesn't. Run TestAssert with and without the -ea flag. Your program should fail for the bad call when assertions are turned on.

Part 2. Add the additional assert statements to the other public Bookshelf methods you wrote, and retest this updated version of Bookshelf on your BookshelfTester.java. (Make sure you run the program with the -ea flag

set.)

For checkoff show the TA your representation invariant, your `isValidBookshelf` method, the two runs of `TestAssert`, and the source code and run of the code you wrote for part2.

Checkoff for DEN students

If you did your work using the Vocareum IDE you should have completed your work in the PA2 workspace. When you are ready to submit your lab, you will need to copy this work to the Lab 5 workspace. You can do this as follows:

1. Select the work folder in the file list in PA2 on Vocareum and choose Copy (menu above that list). Or if you did your lab work in some subdirectory of the home dir on PA2 workspace, just select and copy that dir instead.
2. Then go to the Lab 5 workspace, and select your work directory on the file list. (You do not go through d2l to get to lab 5, but directly on the vocareum website. If there is no list of assignments to choose from at that point, use the URL labs.vocareum.com to connect to see the list of your assignments, then choose Lab 5.)
3. Now click Paste (same menu). The Paste will create another *sub*directory called `work` with the new files and sub-directories.
4. You can then use Linux to move the lab-related files into your home directory (that's where the Submit script will expect them).

If you completed your work outside of the Vocareum IDE, you should instead upload your Lab files into the Lab 5 workspace, compile and test it there, and then submit it.

When you click the Submit button in the Lab 5 workspace, it will be looking for and compiling the files `BookshelfTester.java`, `Bookshelf.java`, `TestAssert.java` and `README.lab5`. Make sure you put your name and loginid in all the files you submit.
