

Lab 2: Systematic Testing/Debugging Techniques

Objectives:

- Students will create a robust set of tests for a List data structure
- Students will use the jGRASP debugger to:
 - Visualize the data
 - Step through line by line parts of the program
 - Correct Problems in the code
 - NOTE: Please use jGRASP for the labs, not some other IDE!
- Students will write an easy-to-read report of how they debugged the program
- Students will understand how the array-based list works

PreLab

- Skim the documentation about the jGRASP debugger
http://www.jgrasp.org/tutorials187/06_Debugger.pdf
- Draw pictures to explain all of the list operations in an array based list. Do this from scratch, the exercises will serve you well.

LAB 2A

Part 1: Download IntList.java (stub file)

This is a "bad" implementation. But what each of the methods should do has been documented. Read through the required methods.

Part 2: Create a java class called TestList.java

Be sure assert is enabled in jGRASP. (Build → Enable assertions.)

Inside of TestList create main method that will test all of the requirements of the IntList.

Keep in mind that your testing should aim to FIND problems, not verify that it works.

You will be able to compile your code, but you will not be able to test it locally (i.e., without submitting it to the remote machine on Moodle) as IntList is only full of stubs.

Submit

Upload to Moodle. Keep in mind that you will not think of everything the first time. Read the error messages for clues to find all of the bad examples. You can also use System.out to get more information from Moodle as you solve this problem.

LAB 2B

STOP! Continue after you successfully complete LAB 2A.

Part 3: Fix the Array Based List

Timmy spent all night writing his code for array based list, but now it doesn't work. Use a systematic approach to:

1. Identify that there is an error, by running your tests.
2. Find the smallest example that will generate the error
3. Use the debugger to find the cause of the error
4. Fix the error
5. Document your fix

Oh, and one more thing, Timmy made **more** than one error. Find them all!

(Note: Half of these errors were actually from Dr. B's code when he implemented the array List a couple of years ago. Having bugs in initial code is normal. Finding the bugs before you let others use it is what is important!)

Submit

- 1) Your document of how you found each error. Use complete sentences, code snippets, and screenshots to show each of the things above.
- 2) Submit your corrected IntList code to Moodle, Timmy will thank you.

Extra Challenge

No extra credit, but they seem like awfully good questions to put on a test...

Create methods for `append`, `prepend`, and `orderedInsert`.

Extra Credit (Bonus Points!)

Find a realistic error that will get past the Moodle testing. Contact me with your simplest example code.