

Debugging

Due: Day of lab at 11:59PM

1 Debugging Specification

1.1 Available Resources

- Lecture slides
- Other sections of the provided code
- me
- The textbook
- **DO NOT** refer to or use online implementations

1.2 Lab Instructions

- This is an *individual* lab.
- Make sure to read through all of the specifications so your submission is complete.
- Follow all the submission steps in the Setup document by the lab deadline.

1.3 Lab Link

The skeleton code for the lab is available at <https://classroom.github.com/a/D7spNFPJ>.

1.4 Testing

In order to test the existing code in the **LAB3** methods, write JUnit tests to test each method for a range of cases. Read the comments for each method to see the intended behavior. Run the tests to see how the methods perform.

If you implemented your tests correctly, they should all fail because there are bugs in the **LAB3** code! If any test passes, you should be more thorough in your testing.

1.5 Implementation

This lab gives you a chance to practice debugging code that does not behave correctly. The methods are quite simple in their behavior, but their author made some mistakes in implementing them.

In order to get them working correctly again, follow debugging steps to determine where their behavior does not match your expectation. Start by reading through the methods and their comments to understand how they are intended to perform their task. Use debugging techniques from class to determine which lines of code in each method behave incorrectly. Once you have narrowed down which line of code is not behaving correctly, fix that code. Stop and rerun your tests to see if they work, or if they fail in a different way.

In your updates **do not** call the String class implementations. Modify the code minimally so each method is still solving the problem “from scratch”.

After you have completely fixed each method, your tests should all pass.

1.6 JUnit Assertion Syntax

For each of the JUnit `assert*` methods, the first argument is the string that prints out if the test fails. Use any of these four `assert` methods in your tests to confirm the functionality of the `SLList` methods:

- `assertTrue`

- assertEquals
- assertEquals
- assertNull

2 Double Check:

- Have you written three JUnit tests?
- Have you debugged the four LAB3 methods so they pass your tests?
- Have you committed/pushed your code from the two files?

3 Grading

Each of the 7 **TODO** sections is worth $\frac{1}{7}$ of the lab grade.

Grades and any comments for the lab will be posted to your project on github. Grades will also be posted to Brightspace, eventually.