CSCI 3901 Winter 2021

Lab 2: Debugging

Due Friday January 22nd, 08:30 AST on Brightspace

Objective

In this lab, you will become familiar with debuggers.

Working individually you will debug a few programs. While the defects in the programs of this lab may not be complicated, use the time to become familiar with how your debugger works before you are faced with complex code.

Do not resort to debugging in this lab using print statements.

Preparation

- Download `HfxDonairExpress.java`, `Ackermann.java`, and `LinkedListTest.java` from Lab 2 in Brightspace

Resources

- A debugger which allows inspecting variables, breakpoints, stack traces

Procedure

Lab steps

1. Run `HfxDonairExpress` on some inputs. **You do not need to check invalid input**.
2. Examine the output until you find an incorrect value.
3. Set a breakpoint at the start of the program.
4. Select a defect in the program.
5. Run your program in the debugger and step through the program to identify where the defect happens in the code.
6. Determine what could fix the defect.
7. Implement your fix and re-test to ensure that the defect is fixed.
8. Repeat until no defects remain.

Repeat these steps for `Ackermann.java` and `LinkedListTest.java`.

Analysis

What strategy or strategies for debugging are most effective for you?

What makes them effective?

For which conditions of the code will your strategies be effective or ineffective?

How can a debugger support your strategies?

Reporting

1. In one file, list
    - The cause of the defects that you found.
    - The approach that let you locate the defect.
    - The approximate amount of time it took you to locate the defect.
    - The answers from the questions in the Analysis section of the lab.
2. Generate a PDF from the document.
3. Submit the PDF and your Java files in Brightspace in the Lab/Lab 2 section of the course page in Brightspace.

Assessment

The assessment will be on a letter grade and will reflect how well you are demonstrating that you are using and understanding debugging strategies and support tools like the debugger. I am more concerned with your familiarity with the debugging process than on finding and fixing every bug in the lab code.

Letter grade	Defects detected and fixed	Strategies to locating the defects	Analysis thoughts on effective debugging strategies (higher weight)	Analysis on how the debugger supports the debugging strategies
A (Excellent)	All defects fixed with minimal code changes	Different strategies for the types of defects	Demonstrates an understanding of the benefits and limitations of different strategies	Connects the debugger features with the strategies. More than one feature might support one strategy
B (Good)	All defects identified and 75% of them are fixed	Has two strategies for debugging	Demonstrates an appreciation for different debugging strategies in different cases	Knows how a set of debugger features supports one strategy
C (Satisfactory)	All defects identified and a start is made on fixing them	Used the debugger to trace through the code to find bugs	Has at least one strategy other than print statements in mind and knows when to use it or to abandon it	Can see how one debugger feature helps with debugging
D (Marginal Pass)	Some defects identified and fixed	Resorted to some print statements to detect bugs	Repeating what we might find on the web about debugging but not showing that they understand when to use what	Knows how the debugger can at least avoid debugging with a pile of print statements
F (Inadequate)	Little clarity on what's wrong with the programs	Fell to fixing code by perturbing it until it worked	Little understanding of how to approach debugging in the large context	Unsure on how the debugger will help with debugging