CS112 – Spring 2024

Lab14

Instructor: Paul Haskell

FINAL EXAM TIMES

In GitHub's CourseInfo repo, you will find a file called **finaltimes.txt** . Please make a **Lab14** directory in your **MyWork** directory, copy the file there, and read it. Fill out the requested information and push the file to your GitHub repo.

This part of the lab is due Friday March 22nd at 11:59pm.

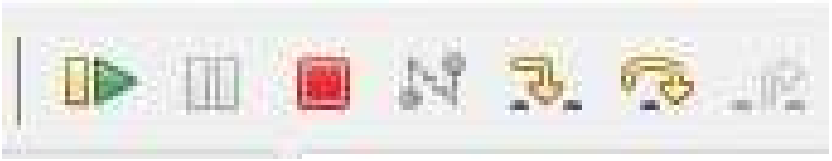
INTRODUCTION

This week you will learn about Eclipse's debugging capabilities. Then you will use Eclipse to debug a buggy program that I will give you. You must turn in the program by the end of the class period. The good news—no homework today!

1. Please start by doing a "git pull" in **CourseInfo**, so you have the latest updates.
2. Now start Eclipse and create a new Project in your **MyWork** repo called **Lab14**.
3. In **Lab14**, create a new Java file called **ReverseString.java**.
4. In a text editor, open the **ReverseString.java** file in **CourseInfo/Lab14**, and copy the text into your version of the file in Eclipse.

Debugging with Eclipse

On the left side of Eclipse's source code window there are line numbers. Right-click a line number that has an actual statement on it and select "Toggle Breakpoint". Then from the Run menu, select **Debug**. The program should run up to your breakpoint and then pause. Just below the menu-bar near the top of the screen, some controls appear that look like:



These controls let you:

- Continue from a breakpoint
- Stop the current program execution
- "step into" – if the current statement is a function call, step into the function and pause
- "step through" – if the current statement is a function call, execute the function entirely and pause at the next statement in the source code
- "step return" – complete running the rest of the current function and return to the calling function

You can create more than one breakpoint. Give it a try!

If the program is stopped at a breakpoint, you can hover the mouse cursor over a variable, and Eclipse will pop up a window showing you the value of the variable. Wow! You can imagine that a graphical debugger like Eclipse will make it a lot easier to figure out what is happening as a program executes.

Eclipse Debugging "View"

Do you see the two icons for the "Java view" and "GitHub view" in the upper right corner of your Eclipse window? Now you can open up a view focused on debugging.

- On the top-level "Window" menu, select "Show View" and then "Other…". Select the "Debug" entry (not "Debug Shell") and push the "Open" button. You now should have a new "View button" at the top-right of your Eclipse window.

The most useful thing in this view is the "Variables" subwindow, which shows the value of all in-scope variables whenever your program hits a breakpoint.

ReverseString

Ok, now onto the actual programming part of Lab14. The **ReverseString.java** I gave you is buggy.

Your mission? Debug the program, of course! The program is supposed to read a string from the command-line's args[0], reverse the string, and print the result.

There are both syntax bugs and logic bugs. Eclipse should automatically warn you about the syntax bugs. After you fix those, you should try to Eclipse's debugging capabilities to find and fix the logic bugs. And then you can test inside Eclipse.

But how do we set command-line arguments in Eclipse?

- go to the Run menu and select "Run Configurations…"
- On the left, click the name of the program you are working on
- Click the "Arguments" tab
- Enter what you want into the "Program Arguments" subwindow and click "Apply". You can click 'Run' right here if you want to

Reminder

When you're done testing, you must push your code to GitHub – you can do this inside Eclipse, but Eclipse does not do that for you automatically.

ReverseString.java must be turned in by the end of class.

Conclusion

This lab hopefully gives you some experience debugging with Eclipse.

Grading Rubric

finalexam.txt is worth 10 points if filled out and submitted on time.

ReverseString.java is worth 15 points: 3 points for each of 5 test cases.