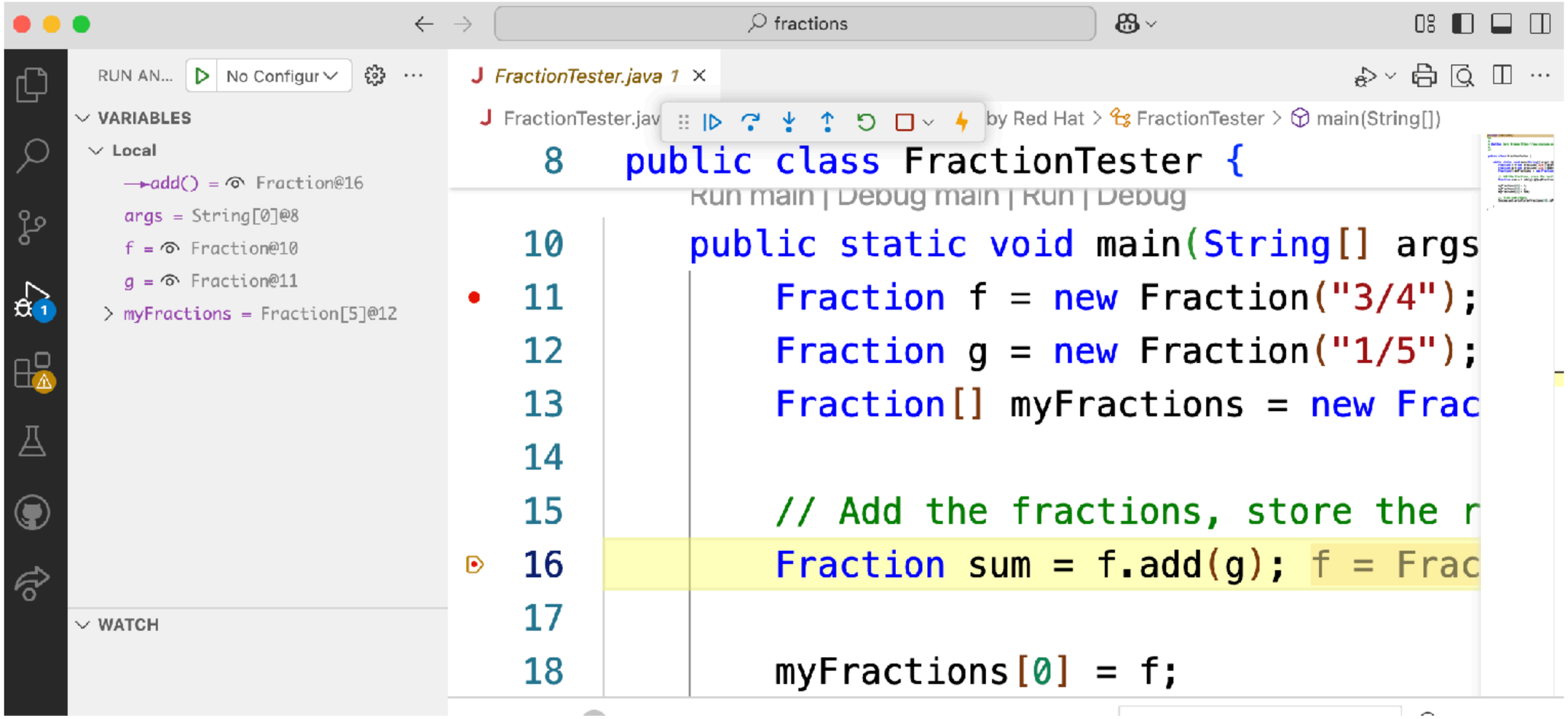
CS62 Lab4: The Debugger

Basic SWE skills



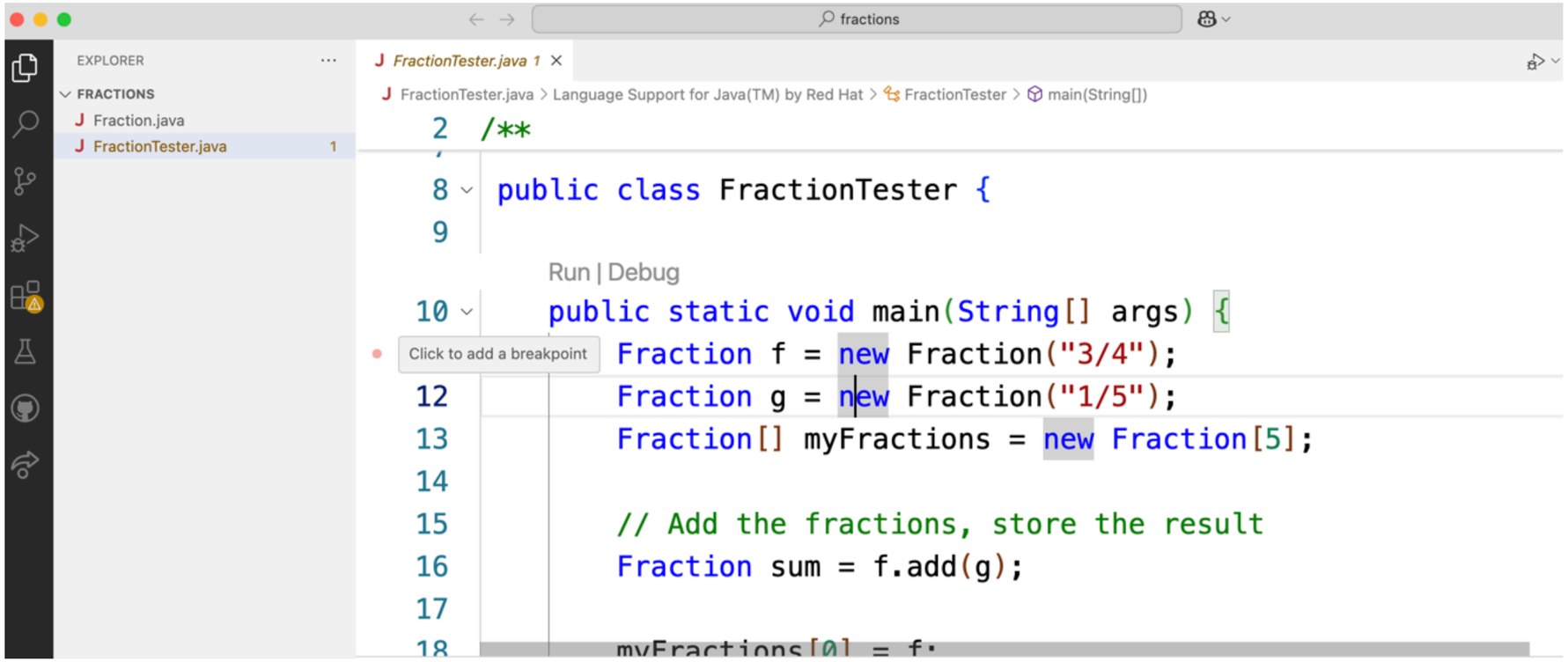
Lab 4 agenda

- Quiz
- Lab

Debugger philosophy

- Software is about managing a lot of *state*
  - What variables are in my current scope? What are the values of my variables? When I run this line of code, where does my code jump? What does my function evaluate to? WHY ISN'T IT WORKING???
- Debuggers let us pause this state (through setting **breakpoints**) and show us all this information at once so we can reason about it
  - Once you avoid compiler errors, a debugger will slowly and meticulously let you **step through** your code, line by line, to catch runtime errors.

Using the Debugger



1. Click on the red circle on the side to add/remove a breakpoint

2. “Debug Java” to start debugging

Using the Debugger

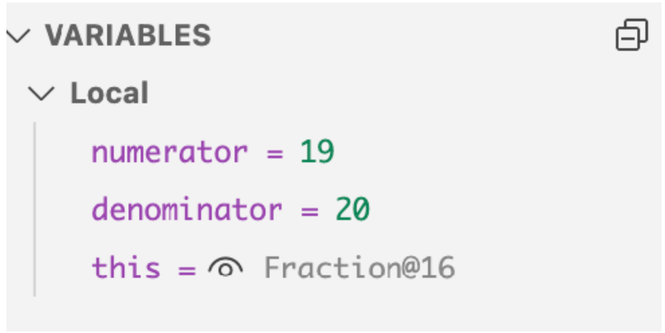
Continue until next breakpoint

Restart Stop Replace code

Step over (if there is a function call, don't go into the function)

Step into (if there is a function call, do go into the function)

Step out of (if you are in a function, exit the call stack)



Sidebar lists all variables in the current scope

Lab structure

- Use the VSCode debugger to debug 3 matrix classes with different errors
- Checkoff for lab today (no style points, etc) - 3 points for finding the errors, pushing to Github, and submitting to Gradescope