Hi, I'm Brittany and I'm currently conducting research to help developers understand test failures. I want to thank you for taking time out to help me with this research.

As a reminder, both screen and audio, is being recorded.

Today you will be attempting to understand, explain, and fix real-world defects found by failing tests in the Apache Commons Lang source code. Your fixes will be saved as part of data collection.

Let's start with a training task, which will serve as a tutorial for how the rest of the assignment will go.

On the screen you see some JUnit test code. Are you familiar with JUnit within Eclipse?
To run a JUnit test in Eclipse you right-click in the editor, find 'Run As' and select 'JUnit Test'. Now, for this task, you can just click the green arrow button (*show*) to run the test. For each new task, you will have to initially run the test using the 'Run As' menu option.

Now, if you look in the JUnit view to the left, you'll see to the left some tests ran and this test failed. Your goal is to figure out why it's falling and fix it.

For each defect, you will have a series of questions that need to be answered to satisfy the requirements of this assignment. The questions are inside the IDE (*show*), however you must answer the questions within the actual assignment (*show*). You can answer each question here as you work through the problem or once you have fixed the defect. However, you must complete the questions for each defect before moving on to the next defect.

For all tests, you will have access to the failing test, source code, and the failure trace provided by JUnit. <*Walk through each, what they mean, how to access*>

For some tests, like this one, the test will be annotated with *Fuzzy Driver Capable*. For these tests, you are able to get causal execution information that you may want to use. The causal execution information includes tests similar to each failing test that pass and fail and traces of each execution. Each trace includes method calls, the line on which the method call occurred, and the value (if any) returned from that method call.

Go ahead and see if you can figure this one out.

Sample Response:
Now that you've fixed the defect, typically you will have to formally answer these questions in the browser. But for this one, let's see what we expect responses to look like.

Once you're ready to go on to the next defect, you will go to 'File' → 'Switch Workspace' and select the next workspace in the sequence. For example, the training test is numbered 00, so the next defect will be 01, followed by 02, and so on.

A few things to remember before we get started:

- Once we move past the training I'll be more hands off. I'll be able to assist in certain ways but there are certain questions I won't be able to answer until after you complete each task.
- The passing and failing tests and traces will only be available for some defects. But you are welcome to use them when they are available.
- The problem will always be in the source code and removal of source code is never the right fix.
- All tests must pass; if your fix breaks another test, it is an incomplete fix.

Once you're done with the assignment, there will be a short follow-up questionnaire to fill out.

Are there any questions before we get continue?