CS 240: Java Debugging Introduction Transcript

This video shows a split screen of Professor Wilkerson on the right and a PowerPoint screen on the left. Any text displayed or action performed that is not verbalized will be included in italics as visual descriptions.

[00:00:00] **JEROD WILKERSON:** I'm going to talk about debugging now.

[00:00:03] I want to start with an example that you're likely to experience.

[00:00:10] Some of you have probably interviewed for an internship before and you may have been asked to do what's sometimes referred to as a tech assessment or some programming assignment as part of your interview to get that internship.

[00:00:24] If you haven't done that, you probably will do that when you go to get your first job.

[00:00:28] Back when I started in the early to mid 90s, it wasn't that common to go for a job interview and be asked to write code, but now it happens at most places.

[00:00:39] Usually, if you go to do a job interview, you will be asked to take a tech assessment.

[00:00:45] That often is something you do while you're sitting at the company.

[00:00:50] Often, they'll have you do a tech assessment first and then you do interviews after, but sometimes it's more involved than that and you end up doing something ahead of time.

[00:00:58] You write code before your interview and they give you some amount of time to do it and then you show up and they talk about it.

[00:01:05] The reason I'm talking about this is, first of all, it's useful for you to be aware of that, and second of all, I wanted to give a useful example for debugging and the

example that I'm going to give is a tech assessment that we did at the last company that I worked at before I came to BYU.

[00:01:22] We had this project that we would have people write and we would have you simulate a ball clock.

[00:01:30] I want to show you a picture of a ball clock.

Begin visual description. The main parts of a ball clock are the three separate track (one above the other) that designate time for the balls to sit on, a bin for unused balls behind the lower track, and a device that moves balls from the bin to the top/first track. End visual description.

[00:01:35] This is a clock and the way it works is it has a bunch of balls.

[00:01:40] You can't really see them very well, but they're back here.

[00:01:42] It has a bin full of balls.

[00:01:44] It has something that will pick up a ball once every minute and it will put it on this track *(the top/first track)*.

[00:01:49] The ball will come down here *(to the top/first track)* and it will start off in one of these four spots, and this represents minutes.

[00:01:57] That *(the track)* has room for four balls.

[00:01:59] If a fifth ball comes, the track dumps and it dumps all the balls that were already here back into the starting bin.

[00:02:07] It takes the new ball, that fifth ball, and it puts it here *(the second/middle track)*, which is the five-minute track.

[00:02:12] That has room for 11 balls.

[00:02:14] We get a 12th ball, it dumps them all again, *(into the bin)* and the last ball *(the 12th ball)* goes here to the hour track *(the third/bottom track)*.

[00:02:20] With that, you can actually read the time by counting the balls.

[00:02:24] We can see here that this clock is representing 7:32.

[00:02:28] That's basically how a ball clock works.

[00:02:33] What we did for that tech assessment at that company is we had people write a ball clock.

[00:02:37] That's more involved than most tech assessments, but we needed it to be more involved because this company had a need for really efficient high-performance code.

[00:02:48] It was an internet marketing company.

[00:02:51] It would process over a billion transactions a day.

[00:02:54] The code had to run really fast.

[00:02:56] We did this tech assessment and we would give candidates about a week to work on it before their interview, to write the code.

[00:03:03] When they would come for the interview, we would look at the code with them.

[00:03:06] We'd talk about it. We'd ask them questions like, how could you make that faster, and things like that.

[00:03:11] One of the interesting things that we added to the assignment was when you simulated this, you would simulate the balls as not all being the same.

[00:03:21] You could think of them as being different colors of balls or each ball would be numbered.

[00:03:27] They would start out in order and you would run the ball clock.

[00:03:31] If you think about the operations of how I said this worked, these trays dump balls and they fall back into the tray in reverse order of how they were picked up in the first place.

[00:03:41] The balls get mixed up, but it's mixed up in a systematic fashion.

[00:03:45] If you run the ball clock long enough, eventually, all the balls will be in order again.

[00:03:50] Depending on how many balls you started with, that will determine how long it takes all the balls to be in order again, and that's a lot of running of the clock.

[00:04:02] That's a long time. What we would do is we'd have people, they would write their program to work in two different modes.

[00:04:10] One of the modes was, it would start up with a specified number of balls and it would run until the balls were back in the order that they started in.

[00:04:20] We would also have them keep track of what millisecond it was when they started and what millisecond it was when it ended.

[00:04:27] Their ball clock would print out, or their simulation of it would print out, how many milliseconds it took to run, and that allowed us to see how fast of code can they write.

[00:04:37] Then we can talk about their program and talk about how would you make it faster if this isn't fast enough.

[00:04:42] Anyway, that was the tech assessment and I just wanted to introduce that as the assignment that we're going to use for debugging because I wanted it to be somewhat interesting.

[00:04:54] Actually, before we go to the next video, we had a specification for the program.

[00:05:02] You might as well see that.

[00:05:03] Here is what the assignment looked like that the people would receive.

[00:05:08] It describes what a ball clock is, talks about how it works.

[00:05:12] It tells you that it has to support between 27 and 127 balls.

[00:05:17] There were two different modes that it can run in.

[00:05:19] The first mode is the one that I described, where you start with some number of balls.

[00:05:24] If you start with 30 balls, it would run for a while, and then it would say that the ball's cycle, or go back to the way they started, after 15 days of running.

[00:05:34] Then it would print how many milliseconds it took to run.

[00:05:37] The other mode was you give it two parameters.

[00:05:44] You give it a number of balls and tell it how long to run and then it would print out what the tracks look like in a JSON format.

[00:05:52] It would print out where were all the balls.

[00:05:54] That's how this program works.

[00:05:55] In the next video, I will show you how to debug in IntelliJ, and we will be looking specifically at my implementation of that program.