CS 240: IntelliJ Code Coverage Transcript

[00:00:00] **INSTRUCTOR:** IntelliJ has a built-in code coverage tool that you can use pretty easily, so if you have some code and some tests for that code you probably already know how to run your tests by right-clicking. You could right-click a directory and just run tests and that will run all of your tests for you.

Start visual description. Instructor demonstrates how to run tests on Intellij. End visual description.

[00:00:17] But if you want to have code coverage information, you can right-click and go to more, run debug, and then run tests in Java with coverage. So that will invoke the code coverage tool which will recompile your code to instrument it with the code coverage information.

[00:00:34] Then it will run it and then it will report what tests were covered by or what code was covered by tests and what was not. So right here you see a summary window that is showing us all of the classes in the project all of these classes and their line method and class code coverage.

[00:00:53] So that's what we see by default and then in a minute I'll show you how to enable branch coverage as well. So if we look right here we can see that spelling checker has a hundred percent line coverage method coverage in class coverage most of these classes do let's look at one that doesn't have a hundred percent.

[00:01:10] So database manager doesn't have a hundred percent code coverage. And if I bring that up now that I've executed the test with code coverage, I can see color coding or highlighting here that shows me which lines of code were covered and which ones were not.

Start visual description. Instructor shows the code coverage for different files in the project in the Coverage tab on IntelliJ. End visual description.

[00:01:28] So this shows me that my tests hit all of these lines of code that are green and the part that's here in red was not executed by the test so I'm not testing my catch block here. So if we looked at the others the ones that are a hundred percent it would just show everything green.

[00:01:48] So that's what we get by default, but another thing that you can do is you can enable branch coverage and I'm not sure why that's not enabled by default but it's not and so the way to enable that is I can go back here and let me see.

[00:02:09] Okay, so I can go to my run configuration and go to edit configurations, and this could change.

[00:02:17] Sometimes they change where these settings are, so if this is not exactly where to find this setting in your IDE, you can probably just search for it a little bit and find it, but with the current version of IntelliJ, if you go into a run configuration and then select modify options and scroll down to code coverage, you can click that little check box. I'm going to do that again to make sure you could see it.

[00:02:38] So this enable branch coverage and test tracking. So if you enable that, then when we run code coverage, we're going to see branch coverage as well.

Start visual description. Instructor goes to the Run/Debug Configurations and under Code Coverage he selects enable branch coverage and test tracking. End visual description.

[00:02:57] Okay, so now you can see we've got an additional column here for branch coverage, and let's look at one that had 100% code coverage, but not 100% branch coverage. So Spelling Checker is an example of that, so let's go look at that one.

[00:03:11] I'm sorry, it has 100% line coverage, but not 100% branch coverage. Okay, so now we can see green still means we fully covered it, yellow means that that's a branch where we've covered not all the branches.

[00:03:27] So we can tell that if we've covered these lines of code that are in the if statement, and this is yellow, then that means we've covered the condition where if is true because we got inside of it, but we haven't covered the case where if is false.

[00:03:42] Okay, so that's branch coverage, and that is often a useful addition to line coverage.