CS 240: Code Coverage Tools Transcript

[00:00:00] **INSTRUCTOR:** There are several different code coverage tools available. Some are proprietary tools that are built into IDEs like IntelliJ. And then others are open source tools. Some are command line tools that aren't integrated into some kind of a GUI, but they're used as part of the build process.

Start visual description. Slide titled Java Code Coverage Tools. Text reads:

- *Proprietary / built-in to IDEs like Intellij*
- *Cobertura: Open-source, command line*
- *JaCoCo: Open-source, command line but has Intellij plugin*
- *Parasoft JTest: Nice but expensive*
- *Many others*
- *Command line tools can be integrated into an automated build/ deployment process*
- *Built-in or IDE plugin tools are useful for individual developers*

End visual description.

[00:00:20] So for example, we have CobraTura and Jococo. Those are both pretty popular open source tools that are used. And these are command line. So CobraTura is a command line tool. Jococo is a command line tool, but there's also an IntelliJ plugin.

[00:00:42] So you can install a plugin and use Jococo to report code coverage within your IDE. There's another really nice tool is ParaSoft JTest, but it's expensive. And the open source tools are really good, so most people just use them.

[00:00:57] There are actually several others. And so what organizations usually do in practice is they will have some kind of an automated build process using a command line code coverage tool. But then individual developers should use one that's built into their IDE.

[00:01:15] So as a developer, you're using the IDE one to make sure that you're writing enough tests and have good code coverage. But then when your code gets deployed, the organization isn't necessarily just going to trust you, and so they're going to run your code through some command line tool that's going to verify that you have enough code coverage.