ERIK DEMAINE: I developed this class on algorithmic lower bounds because I felt like there was no class like it. And I really wanted to share with the world how to show that problems are hard. And in theoretical computer science, usually there's two kinds of classes out there. There's more algorithmic classes, which are about showing how to solve problems and give positive results. Yes, this can be solved.

And there's another world of complexity theory, which is about building up a mathematical infrastructure to show that problems are hard. But usually, it's much more common for the algorithms classes to be tuned to particular problems that people might care about. And complexity theory is usually more about theory building.

But there's this other world, the hardness proof world, which is much less common and isn't usually represented by a class, where it's all about showing that particular problems you care about are hard using the complexity theory infrastructure that's been built up. So this class is very much not a complexity theory class. The goal is to be-- because there are lots of classes like that.

I wanted to use the complexity theory infrastructure. Just tell people what they needed to know, but not show how it works inside the mathematical machinery. Because it's quite complicated. And instead, I wanted to build on top of that and show how to prove equivalences between different problems and use that to show that your problem is as hard as an other known hard problem.

And that turns out to be a lot of fun and also very accessible. And so my goal was to-- but unless you're familiar with it, it's quite difficult. My goal was to make it easy. The tagline is "hardness made easy." I wanted to bring students to a point where they're not scared to prove problems are impossible. It's actually not so difficult. And it can be fun.