So in the last lecture, you learned about injection dependencies via method parameters. While this approach works, sometimes, you may run into a couple of issues.

One issue is that you're changing the signature of this method. Imagine we have used this ReadVideoTitle method in ten places in your code, and then all of a sudden you introduce this parameter. Then you have to go and modify those ten places in your code.

If you have used this method in one or two places that is not an issue at all. The other issue you may run into is that some dependency injection frameworks can not inject dependencies via method parameters. There are quite a few dependency injection frameworks out there, every team loves a different framework. So depending on the dependency injection framework you use in your team, this approach may or may not work for you. So I'm going to show you a second way to inject dependency. Instead of injecting the dependency as a method parameter, we can inject it using a property.

So I'm going to declare a property of type IFileReader and call it FileReader.

Now I'm going to remove this parameter, and run our implementation I'm going to use a new property, FileReader.

Now when you run this code, you're going to get a null reference exception.

Because we have not set this FileReader property.

So I'm going to create the constructor.

Now here I'm going to initialize FileReader to a new FileReader. With this approach you will ensure that in your production code you're going to use this real FileReader object, and then when testing this class, just before calling the ReadVideoTitle method, you can replace this real FileReader with a fake FileReader. Let me show you how to do this.

So first I'm going to go to the program

class, as we change the signature of this method, so I'm going to delete this argument. Similarly, let's go to the video service tests, and remove this argument.

Now, before acting, we should replace the real FileReader with a fake one. So, as part of the arrange part. We set service. FileReader. The new FakeFileReader. object. Now, let's run this test.

Okay, still passing beautiful.

In the next lecture I'm going to show you another way to inject dependencies.