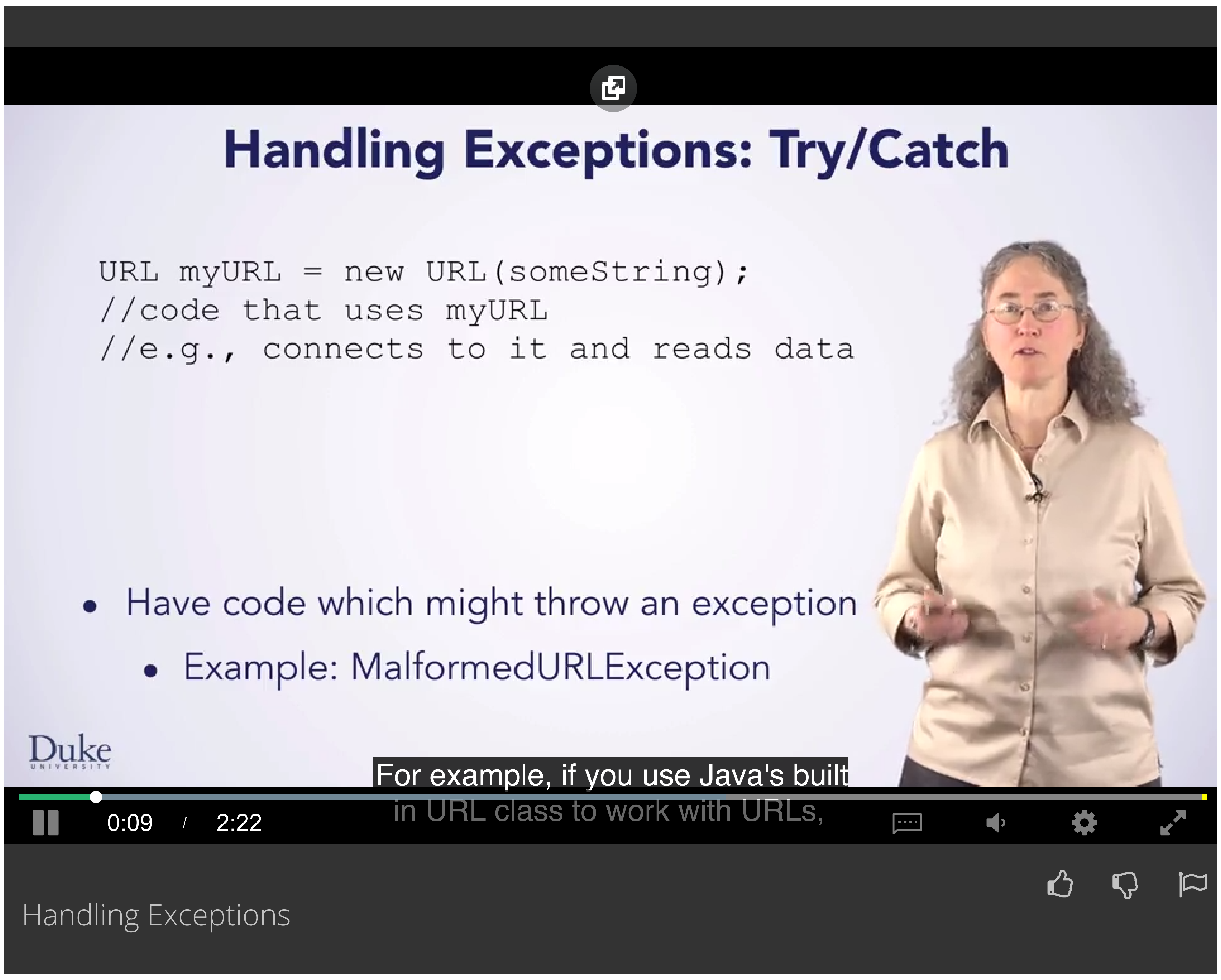
Using Java Beyond BlueJ

Mainstream Java

- Introduction 1 min
- Understanding Exceptions 4 min
- Handling Exceptions 2 min
- Declaring Exceptions 1 min
- Throwing Exceptions 1 min
- Reading Files with Java.nio 6 min
- Hello Around the World with Eclipse 4 min
- Summary 45 sec
- Where To Go From Here 10 min

Review



Handling Exceptions: Try/Catch

```
URL myURL = new URL(someString);
//code that uses myURL
//e.g., connects to it and reads data
```

- Have code which might throw an exception
  - Example: MalformedURLException

Duke University
For example, if you use Java's built
in URL class to work with URLs,
0:09 / 2:22

Handling Exceptions

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript English

0:02
Suppose you have code which might experience a problem. For example, if you use Java's built in URL class to work with URLs, its constructor can throw a MalformedURLException, if you pass in a string which is not a valid URL. Maybe you read this string from the user and do not want your program to crash if the user made a mistake. How would you deal with this exception?

0:25
The first thing you do is put the code that might throw an exception into a try-block. A try-block starts with the keyword try. And then wraps the code that might have problems inside of curly braces.

0:40
Immediately after the try-block, you write a catch-block which states what type of exception you want to handle. In this case, MalformedURLException. And then has the code you want to do whenever this problem occurs. You name the exception much like declaring a parameter, which gives you access to the exception object. You can call methods on it. And if you need to get more information out of it, the code in the catch block can do whatever you want. For example, telling the user that their URL was not valid and asking them to enter a new one.

1:12
When the string is a valid URL, no exception will be thrown. In such a case, Java will execute the code inside the try block as normal. When it reaches the end of the try block it will skip over the catch block and continue to execute the code after it. However, if the string is not a valid URL, then when somewhere inside of this constructor a MalformedURLException will be thrown, Java will then go inside the catch block and begin executing whatever code you wrote there to handle the error.

1:48
After finishing the catch block, Java will continue to execute the code immediately after it. You can also use a finally block, which is always executed whether or not an exception was thrown. Finally blocks are often used to clean up resources which were allocated, and need to be freed regardless of what happened. Java 7 also introduces a try with resource construct to simplify releasing certain types of resources, but we aren't going into that here. Thanks.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?