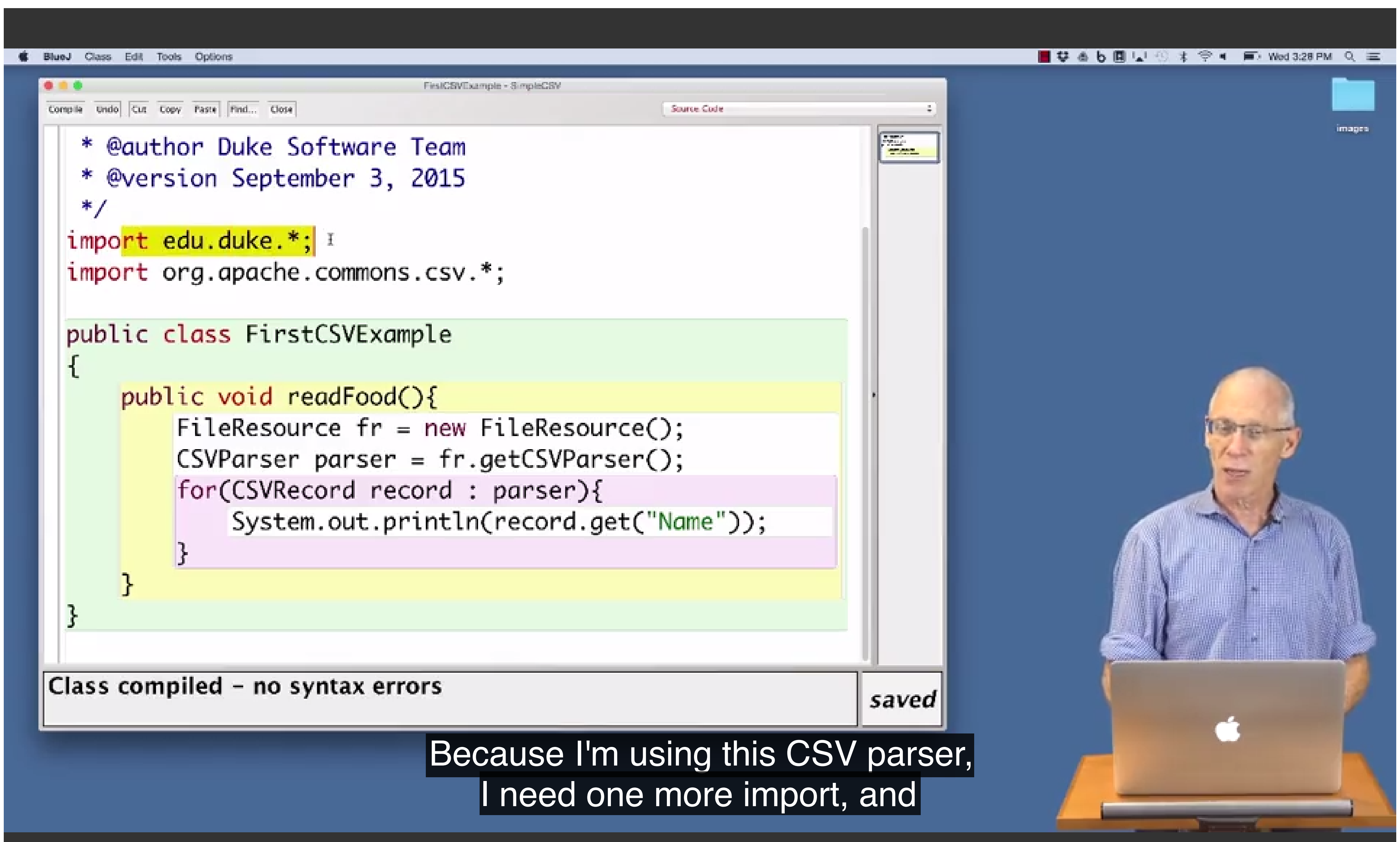
BlueJ Class Edit Tools Options
FirstCSVExample - SimpleCSV
Compile Undo Cut Copy Paste Find... Close
Source Code

```
 * @author Duke Software Team
 * @version September 3, 2015
 */
import edu.duke.*;
import org.apache.commons.csv.*;

public class FirstCSVExample
{
    public void readFood(){
        FileResource fr = new FileResource();
        CSVParser parser = fr.getCSVParser();
        for(CSVRecord record : parser){
            System.out.println(record.get("Name"));
        }
    }
}
```

Class compiled – no syntax errors

saved

Because I'm using this CSV parser, I need one more import, and

Using CSV Libraries

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

Interactive Transcript

Search Transcript English

0:03
Hi. The code you're about to see will show you how to access a comma separated values file, named foods.csv using java. To better understand that program, here are three views of the data in the foods.csv file. The file itself looks like this with the first row having labels used for each column. This is the header row. You can see the labels for each column of data, and you can see the data in each row is separated by columns. Here's the view using the spreadsheet program Microsoft Excel. Some of you may have used this program before.

0:39
The Google Sheets program that runs in a web browser is free software that also allows you to manipulate spreadsheets. Here's the view of foods.csv from that program. You can see the first column has the label name, the second column has the label favorite food and the third column has the label favorite color. Let's get coding.

1:00
I'm gonna walk through a simple example of using the csv libraries that we have in our course. So that you can understand how to create a csv parser, and how to use it in the most basic ways. More complex ways will be something that you learn in later parts of this course and that you can read about when you study the API. So I have a simple first CSV example. I'm going to open up the code so that we can see a few things. But, rather than studying it in detail right now, I'm gonna run it very quickly so that we can understand how it works. And then, we're gonna walk through each piece and make one small modification. So, my class is already compiled. I can see that because there's nothing shaded here. I'm going to right click and create a new object on the object workbench. I already have one. Now I have two. I'm going to read the food from that.

1:52
That pops up a file dialog. This is an example of our directory resource, and I choose foods.csv, a csv file and the program comes in and reads, and prints Drew, Owen, Susan, and Robert. So I'd like to try to understand why it's printing those names, and then see if we can find some more information in this CSV file other than just our names. Looking at the source code again, I'm going to notice a few things. First, I've imported the edu.duke libraries which is common in many of the examples that we do. Because I'm using this CSV parser, I need one more import, and that's a very complicated one, but kind of something that you'll be able to just copy and paste over a while. org.apache.com.csv, we're using an open source library for our CSV parser, and we've made it a little more convenient to use in a way that I'll explain. I have one method, read food. I've created a file resource object that's using our standard library. Because it has no parameters, the file resource object will pop up a dialog and allow me to navigate to the file I want to use. I just showed you using foods.CSV. Then, I asked the file resource object, F-R, to give me the parser. Get CSV parser. This is the new class, the CSV parser class, that's part of my Apache library that you can see highlighted on the screen.

3:17
I now loop over the iterable, that is the parser, getting a CSVRecord each time. So I have two new classes here. The CSVParser class and the CSVRecord class. The CSVRecord class has one method that I'm using, get, and that allows me to get one of the records on the line of that CSV file. As you may remember, CSV files consist of several elements of data separated by commas. One of the elements is named Name in this case. Since I've studied the CSV file, I know that one of the other elements is named favoriteFood So, if I ask the record to get me the field, the element that's favorite food, that will do it. I'm going to print this one with just a space after it. So, notice I've changed the print lin to just print. That stays on the same line, Println will finish the line. I'll show you how that output works. My class is compiled, no errors. I'm going to create a new object on the workbench by right clicking which is what I normally do here. So I'll select, right click, create a new one, and then, on the object workbench I'm going to read Navigate to the food.csv file. And notice, now I have Drew's favorite food is chocolate, Owen, his favorite food is pineapple. Susan really likes cake and Robert likes pizza. One more example, it turns out in addition to favorite foods

4:55
In this CSV file, there's also a favorite color. So, I'm going to print the favorite color. We'll look at those briefly, and then we'll review finally. So, favorite color and I should put a space at the end of that line so it's easier to read. I'll compile that when I create the new object it appears on my object work bench. I'll run that, navigate to the foods.csv file, and low and behold, Drew's favorite color is green, surprisingly Robert's favorite color is green too. If you're a very astute observer of the courses that we've done so far, that might make sense to you. Susan loves purple, and I like blue. That is running through, let me review one more time. In this CSV file that this program read, there are three fields, name, favorite color, and favorite food. If you tried to get another field, so for example, I decided to say that my favorite number was get favorite number. So, I'll say get favorite number. That will compile, and when I try to run this example by making a new object and running, by right clicking.

6:24
Sometimes I can't right click too well. Open foods.csv, and I've got all kinds of illegal argument exceptor. Favorite number not found. My CSV file does not have a favorite number field, and so I could not open that. When we study this in more detail, and when you read the API, you'll see that there are ways that you can avoid trying to access CSV elements that don't exist. But for right now we've seen use a library appropriately, that's org.apache.commons.csv. Get the CSV parser from a file resource object, and then loop over the parser, which is an iterable, to get records one at a time.

7:06
Have fun with CSV and finding information from data.

Downloads

- **Lecture Video** mp4
- **Subtitles (English)** WebVTT
- **Transcript (English)** txt
- **Lecture Slides** pdf

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