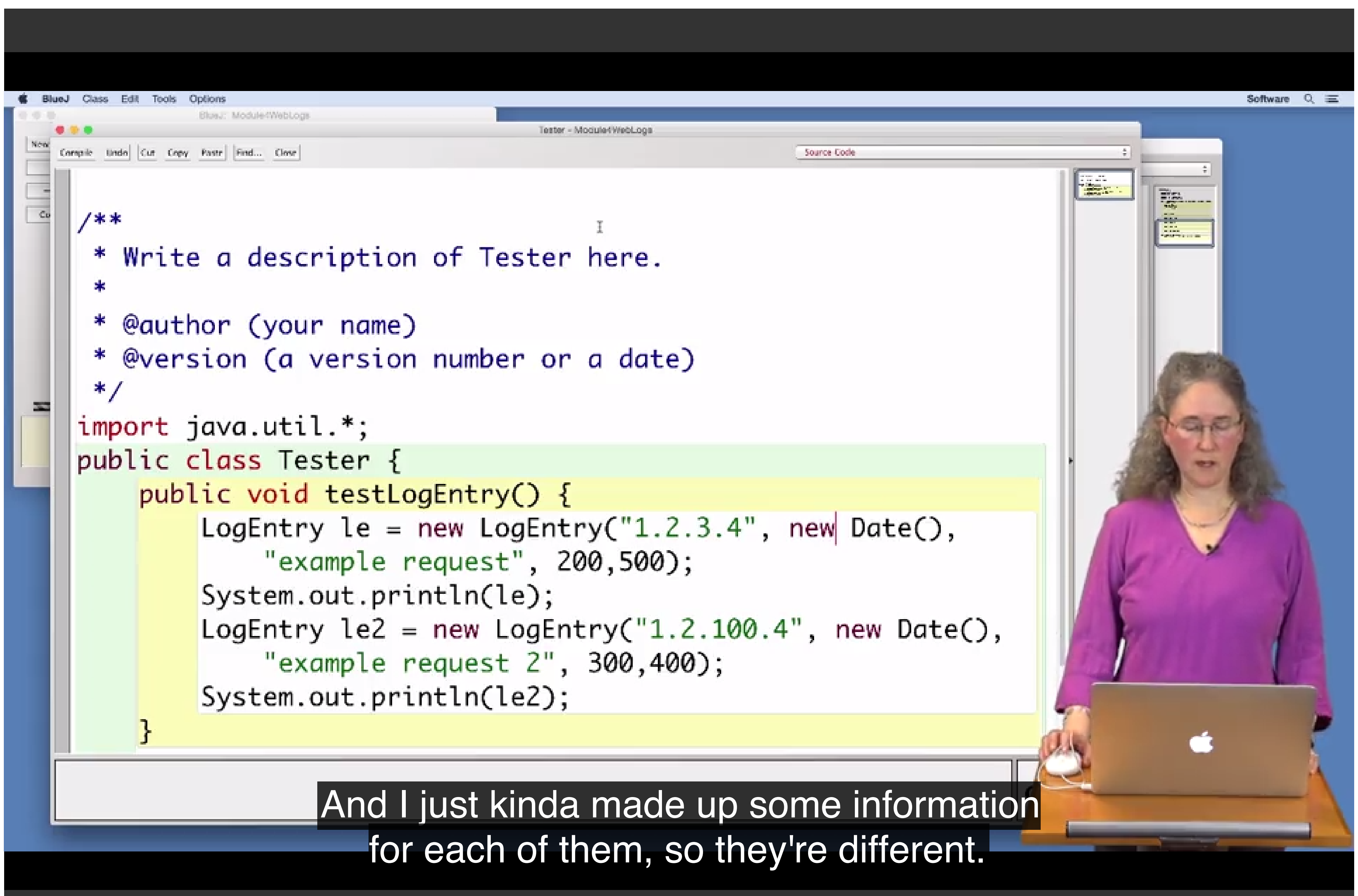
Reading Log Files

- Module Learning Outcomes / Resources 10 min
- Introduction 1 min
- Understanding Log Files 6 min
- LogEntry Class with toString 5 min
- Parsing Log Files 3 min
- Summary 45 sec
- Programming Exercise: Reading Log Files 10 min
- **Practice Quiz:** Reading Log Files 3 questions

Finding Unique IP Addresses

Counting Website Visits

Review


BlueJ Class Edit Tools Options
BlueJ: Module4WebLogs
Tester - Module4WebLogs
Compile Undo Cut Copy Paste Find... Close
Source Code
Software

```
/**
 * Write a description of Tester here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
import java.util.*;
public class Tester {
    public void testLogEntry() {
        LogEntry le = new LogEntry("1.2.3.4", new Date(),
            "example request", 200,500);
        System.out.println(le);
        LogEntry le2 = new LogEntry("1.2.100.4", new Date(),
            "example request 2", 300,400);
        System.out.println(le2);
    }
```

And I just kinda made up some information
for each of them, so they're different.

LogEntry Class with toString

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

Interactive Transcript

Search Transcript English

0:03
Hi. So today what we're gonna do is we have this LogEntry class, and I want to show you toString. So the LogEntry class, notice it has these five fields here, ipAddress and so on. We've got our constructor and we've all these methods. And I've written toString down here. And you can see that it returns those five fields as one big long string and what I've done is I've got a tester over here, that is creating two log entries le and le2. And I just kinda made up some information for each of them, so they're different. And then, it prints out the actual object. It prints out le and it prints out le2. So, let's run it and see what happens when we do this.

0:48
So I'm gonna come over here. Notice I need to compile both of these. I'm gonna compile a tester. And it actually compiles both of them for me. I'm gonna now run tester.

1:05
Test log entry, and there you can see it printed out, the five pieces of information for each log entry, very nice.

1:17
So, now what I'm gonna do is, I'm gonna come back to my class and instead of calling it toString, I'm gonna change the name, I'm gonna call it, [NOISE] getLogInfo. [NOISE] And compile it again and run it. [NOISE] Now let's see what happens.

1:48
So I've got my object, and now I'm running test LogEntry. Oh, so I ran it, but it just printed out something else. Let's see what it printed. It actually printed out the memory location of each of those objects. So let's go back to test our tester. Here, and you can see when I print out an object le. It doesn't know how to print it out.

2:15
So, it just prints out its address location.

2:19
It's not calling the method we wrote, which is called, let's see here, right here. We called it, get logged info, it's not calling that because we didn't specify it. So let's come back over here and we'll specify to print out a log entry with that method, we need to actually write, LogEntry.getLogInfo, and I'll just do that for the first object, and I'll leave the other one le2 like that, and let's see what happens. We'll compile it, and we'll come over here and run tester,

3:04
and so, you can see, for the first object I called get log info, and it prints the five pieces of information. But, for the second one, I just said print the object, and so it just shows the memory location of it. So what's going on here?

3:21
It turns out that every class has a two string method by default. But it only knows to print out the memory address of an object, unless you actually specify a two string class. So I'm gonna change this name back to toString.

3:47
And I'll come back over here in our tester, and get rid of this. Cuz notice, I didn't actually call toString here. I just said, print out the object. And again, if I show you what happens here.

4:05
We don't need the. Okay, there we go. So, we're just gonna print out the object, we're not going to say how or anything, but it just knows go look in my class and if there's the two string method, that's how I'd specify how I want to print it out. So again, we call it here.

4:34
And even though I didn't call two string, it knows, it says, you have a two string method, I'll use yours and it prints out the five pieces of information the way i specified I wanted the object printed out. I want to show you one more thing, I'm gonna come over here, that name two string is very important. It has to be spelled exactly two string with a capital S because I'm gonna show you, if you change it to lowercase s, so it says two string, then when I come over here and I run it.

5:13
Let's see what happens. It goes back and prints the memory address, because it says, if there's a two string in there, spelled with a capital S, it's gonna use that. But that's the only thing it looks for, since I don't have it spelled correctly, it doesn't find it. And so it says, I'll just print out the memory address location.

5:32
So just remember that all objects have a default two string method and if you don't like it, it's just gonna print out the object address, then you can write your own. And so, when it runs, it'll see if there is a two string method in there, then it will call it. That's it. That's for two string, 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?