

Generating Random Text

Module Learning Outcomes / Resources	10 min
Introduction	5 min
Order-Zero, Order-One	6 min
Finding Follow Set	7 min
Implementing Order-Two	9 min
Testing and Debugging	7 min
Programming Exercise: Generating Random Text	10 min
Practice Quiz: Generating Random Text	7 questions
Interfaces and Abstract Classes	9 min
Summary	2 min
Programming Exercise: Interface and Abstract Class	10 min
Practice Quiz: Interface and Abstract Class	4 questions

Word N-Grams

Review

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



Interactive Transcript

Search Transcript

English

0:03

Wait a minute. I don't think his output is right.

0:08

It's kind of hard to tell when it's random like that, but when he did Markov three and it still looked like gibberish, I don't think that's right. But if you think about it, we're using Markov three. We've got three characters that should be together and we're looking at a character that follows it. So we should have in our output every group of four characters, every substring of four characters, should be four characters from the actual training text, but that still looked a lot like gibberish. So I'm going to try and see if I can debug his program and see if there's a bug in it. So here's is his get random text message. We're going to look at Markov Two, which is where he was, and it's doing one thing or at least I'm looking at my code and [I think it's going to do one thing, but it actually doing something else](#). So how do I figure that out? I'm going to have to print out some information.

1:03

Now I don't want to work with a big file, so I'm going to come back to Markov Runner and the first thing I'm going to do is just put in a small string, so I'll just set st equal to.

1:17

This is a test, yes a test. So something really small. And that will just block the file even though we're going to load a file, we're just going to replace it immediately with a small stream for testing. And then in my markov two class, what we're going to do is we need to print out the key and the follows that we're getting back to see if those are correct. So I'm going to add a print statement here.

1:57

Let's see, key

2:08

And then we'll just put the, so we're printing the key, and we're going to print also beside it the follows array. If we can spell it right.

2:21

Okay, so again, we have a small amount of data we're working with now. So let me go back over here to Markov Runner and let's see, We'll make this smaller too.

2:42

And let's compile it.

2:47

And let's see what happens now.

2:53

This is always helpful to figure out what your program's doing, because you think it's doing one thing, but it might be doing something else.

3:00

Now we're going to load a data file, but then we're going to ignore it, because we have that strings so I have to load Romeo. Here we go.

3:09

So I would did this so we could look at the follow set, but I'm looking at this and oh my gosh, we're supposed to have a key that's two characters long and you can see that our key is only one character long. The first key is two characters, but all the rest are just one character. That is a problem. Our characters should be, our key should always be two characters. So that the way Markov works is you take your old key and then with Markov two we want to start with, we want to shift a little bit and use, could take off the first character and add the new follow character we use and use that to predict the next character. So let's look at our code.

3:54

We just get random text, so you can see here we set the key in this line here up here we set the key, and it is two characters, that's the first key. And where do we change the key? We change it down here, and we're just setting it to that next character, so that's wrong. What we want to do is we want it to be the old key minus the first character so we'll just start it at sub string one. And then we'll tack on at the end, the new character that we just found. We'll just shift it a little bit each time. So let's try running that. We'll compile it.

4:46

Okay, so we're going to run it, now again we're running it on our test data to see if that works.

4:55

Here, now, oh yes and that looks much better. And you can see we have, all our keys are two digits, some of them have a space in there so they look funny, and then we have text. So now let's run it on a file. Let's run it on Romeo now. So we need to comment out what we just put in markov two.

5:13

We're going to comment out the print statement here.

5:18

That's the extra print statement. We don't need that now.

5:22

Okay, that's good. And then we'll come back over to Markov Runner and we'll ignore this extra small string to put in there, so we can now run it on Romeo. And we'll also change this back to 500. We can look at a big thing now.

5:41

And let's compile it and run it.

5:53

So we're going to select Romeo to run it on. And there, that looks a little bit better, more like Romeo.

6:01

But let's run it on Markov Three now, so this is what he had done before. If we come back over here at Markov Two, he said there was three changes we'd need to make. And now we can make those same three changes. Here we go.

6:16

We're going to change the key length to three. Let's see.

6:24

Everywhere we see a two, we change it to a three, and this should give us even better, when we run it, even a better random text generation that looks more like Romeo. So those are the three things he changed before. Our key down here will still be the same, it'll be of link three every time. Let's try that.

6:50

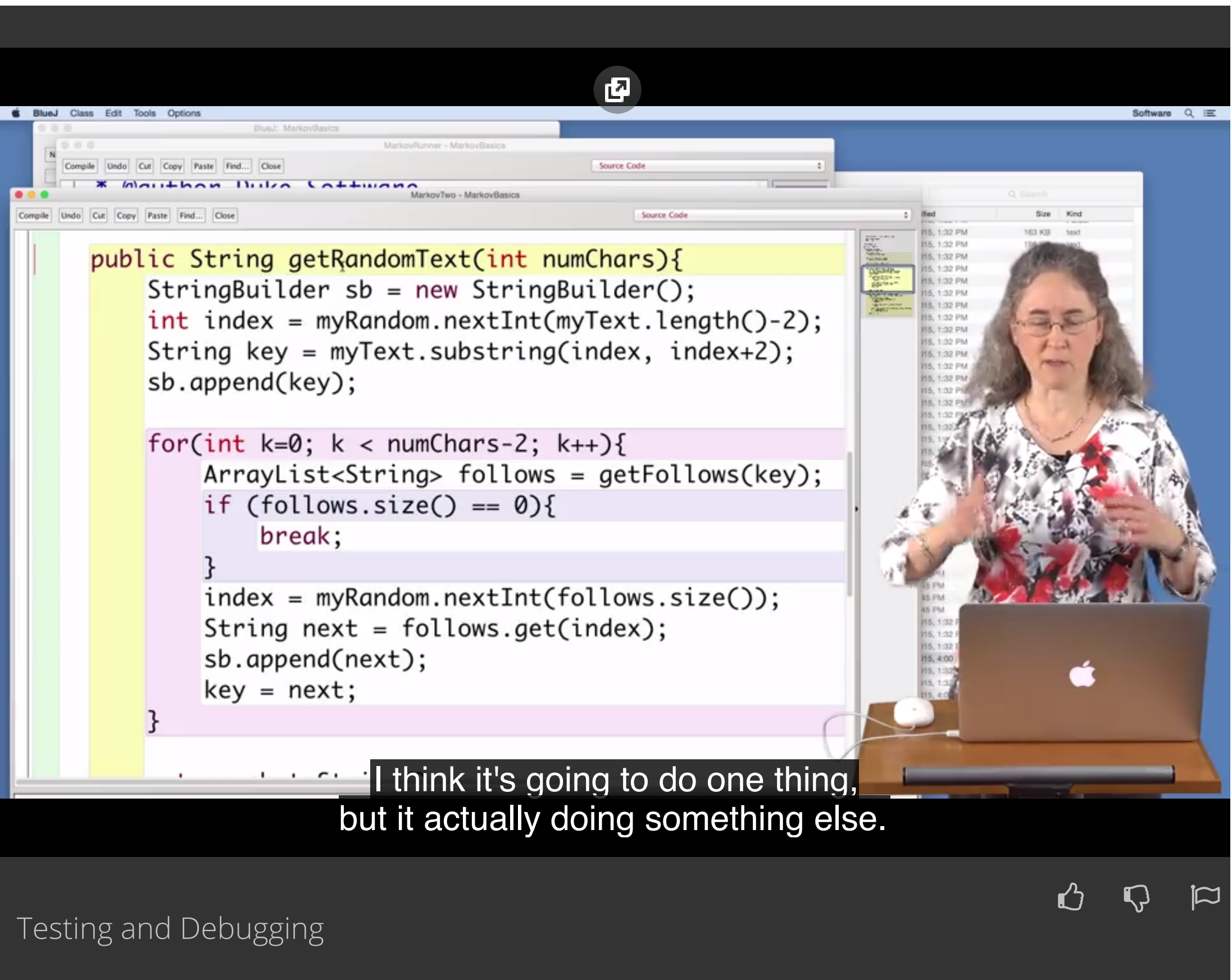
Compile, markov runner. And so it says markov two, but we're really running markov three.

7:01

And we're going to run it on Romeo.

7:05

Now let's see what we get. Oh yeah there we go. Nurse, good hite me, my go all! Those look like words, so that looks pretty good. Now, love, ear, eyes, for the, so we're getting almost real words now for most of these. That looks much better, so, sometimes when your program doesn't look quite right, you're going to have to think small, test cases, and add some print statements to print out to figure out what's going on, because you may think one thing and something else is going on. Anyway this looks a lot better, thank you.



Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?