AUBURN UNIVERSITY

Transcript: Random Class

>> Now let's take a look at the Random Class. It's part of the java.util package as was the scanner. But the random class provides us a means to generate pseudo random numbers. It performs calculations based on a seed value, to produce these random values. And, and we're just gonna use the default seed in, in our examples.

So let's, let's take a look at the, RandomNumbersExample1 program. This is in the examples folder, and I've got it open. We'll go to the debug tab here. And just take a, let's just review it just a moment here. Notice we're importing Java.util.Random up at the top. And then we've got the class name.

And in our main method we're gonna create a an instance of Random. So we say random generator.newrandom. We call random is what we imported. That's the class. And then we declare an int and a double and call them random number 1 and random number 2. And then here's where we're actually doing the generating.

So the instance of random we created was called generator. And so, here in line 19, we're calling generator.nextInt. And this generates a full range random integer. It's plus or minus the 32 meg. Or 32 gig I should say. And then we print that out and then in this next statement, we're gonna generate a, we're gonna say generator.nextInt and we're sending it a 5.

We didn't send it any parameter previously. When we send it an integer like this and call nextInt It says that we want a random number back between 0 and up to 5 but not including 5. So this is gonna give us a random integer in the range 0 to 4.

And the next one, notice we're calling random. NextInt, no limit. So we get a full range number. We're taking the absolute value of that so that gives us a positive number. And then we're gonna divide by 5 and get the remainder. And when you divide by 5 your remainder's gonna be 0,1,2,3, or 4.

So this gives us that same range random number. In the next example on line 30 we get 0 to 19 by, by giving it 20. And then we add 5 to it as an offset. So the range would be from 5 to 24. And we do that again in the next statement here at line 33, we give it an 11.

So that's gonna give us something in the range 0 to 10. We add 90 to that and then that would give us a, an, an a random number in the range from 90 to 100. We want a negative number, we could subtract 10 here, and and that would give us something from -10 to -1, since this is gonna give us a 0 up to 9.

And so on. Now here is the first double. We're saying generator.nextDouble. And this gives us a a value by definition between 0 and 1 but not including 1. So essentially to well between 0 and 1 and not including 1. Now if we wanted to get out of that range we could multi call next double and multiply by 10 and since next double is gonna be between 0 and 1, but not quite 1, this would give us something between 0 and almost 10.

And because we're multiplying 10 by something slightly less than 1 there. That would work for us. And then here's a another int here we're doing. And and this is based on the previous random number num 2 up here which was a double. Here we're adding 1 to it.

Recall it's in the range 0 to 9.999, we're adding one to it and then casting that with an int to get back to randomNum1 which was declared as an int, and so that would give us something in the range from from 1 to 10. So let's step through this and let's see what we've got.

I'm gonna set, a break point where we're creating our first one. And then we'll run in debug mode. So we're there, and we're about to generate our first random number. So I'm gonna step, and there we see a random number, it's a full range random number, pretty large.

And if we wanted to see some more of those down here, we could just copy this statement down the interactions. And then I'm going to hit Enter and notice I didn't put semicolon at the end. And there's that full range number. I did the up arrow to, to just execute that statement again.

You're seeing sort of the full range there, negative and, and positive ones, and so on. So now I'm gonna go back and, and step again. And, this next one there, this is the one we said was gonna be in the range 0 to 4 because we're sending it a 5.

So I'm going to step, oops, stepped too far there. I actually stepped all the way to the end. Let's start over there. I'm gonna run in debug mode, I didn't mean to step, to the end there. So that's step, step. Now we're down here, there's our, random number in the range 0 to 4.

And so, again, I'm gonna copy, this guy here. Down to interactions. Notice without the, without the semicolon, this makes it an, a, a expression rather than a statement. And there we get 1. I'm using up arrow there to get the the next one there. We'll just generate a few of these.

And you see they're all in the range 0 to 4 and so on. And that's what a random number generator does for us. This next one does the same thing recall, I'm going to just do option down here to step. It's getting the full range in it, taking the absolute value of it and then dividing by 5 and getting the remainder, so again we're gonna be in that range 0 to 4 and you can see it over here.

There's randomNum1 is now 3. Now this one we're gonna generate 0 to 19 where we're at here on this statement. And then we're gonna add 5 to it so we we end up with something in the range 5 to 24. And there it is we've got 10. And again if you wanna see some more of those you could just again copy this down into interactions.

And hit enter and I'll just generate several of these, and this is all in the range if you will 5 to 24. Okay. And then I'll go back and I'm gonna step again. Go back here and down arrow. And then this is here we're getting something back in the range 0 to 10.

All the way up to 11 but not including 11. And we're adding 90 to it. So again we get something in the range of of of 90 to 100. That was 91 in that case. This will give us a negative value, in the range of -10 to -1.

We see there we got a -2. Then finally, here's next double. So this, this gets a full range, well it's not a full range double. It's between 0 and 1 but not quite 1. And and there it is, we've got 0.5 and so on there. If we wanted to see a bunch of those again, you could just copy this statement down into Interactions and hit Enter, up arrow.

Those are all between a 0 and 1. Sort of full range there, if you will, and so on. Coming back to our statement. Here this is gonna print that out. Now if we wanted this a double but we actually wanted it in the range of 0 to .999 we can just multiply this by 10.

Recall this is gonna be in the range 0 up to but not including 1 so .999 and that's gonna give us one when we multiply by 10 this other range. And when I step there we see that in that case we got a 1.95. Let's generate some more of these for fun.

Again, copying it into interactions and hit Enter. And up arrow gets the next one. But you see we're there in the range 0 to 999, let's see if we can get up, get one that's up near 9 here. There's a 7, there's an 8.8 and change, a 2, oops forgot to do the up arrow.

7.24, there's a 9.6. So you see sort of the the range we've got there. So I'm gonna step one more time here. And this case we're taking that last number we generated which was point this random num2 here, and we're going to convert that to an integer and add 1 so we'll get something in that range of above there.

So that's been a quick review or a quick run through I guess of using the random class. Just keep in mind for integers it's great you can give it an integer and it'll get you 0 up to that integer. If you give it nothing you get the full range there and if you need an offset you can sort of do that.

The doubles that it generates, if you use next double, they're gonna be in the range 0 to 1, but not including 1. And and then you can also give those upset, offsets, as well.