CSE142 Lecture Questions for Wednesday, 2/10/21, section A

Time (e.g., 12:45)	Question	Answer
	Hello, is there any place to see the answer to the questions in the database of past exam questions? Okay thank you!	We don't provide a key to those, although many of them are in PracticeIt, so you can check an answer there for those problems.
	Are we using java 8? I meant just in general. But ok. Ok! Thank you	You won't be using it for homework or exams. It's such a different style of programming that we don't tend to use it for most of what we do in Java.
	Why use : and not = in your for loop?yeah i'm kind of slow on following the for each loop. Ok sounds good	I think you're referring to the foreach loop. That's just the syntax that Java chose. The syntax of the foreach loop isn't overly important. I'd pay more attention to the later parts of the lecture where I show you what you can do with the functional constructs.
	Wait you can use -> as an arrow and it'll work for the forEach loop? Ok THank you for taking the time to answer questions during this optional lecture!	No, it only works in certain contexts for these functional programming constructs. I used it mostly with streams.
	Why don't you need ; at the end of some of your lines (.sorted() or like .filter())? So it's like punctuation in a sentence. Instead of putting a period, you're.. I guess streaming the program through all the modifiers you're calling continuously. Make sense. Thank you!	With these stream examples, I'm generating a long expression where I am calling a sequence of methods. If you take the example of sorted(), I am taking the output of one function and then calling the sorted() function to have it process the stream. It's a different style of programming more like f(g(h(x)). It's more like the use of prepositions "the square of the filter of the sorted sequence of the distinct values of the array."
	I would have thought that this stream method is frowned upon because i thought programmers want to create programs that aren't so "waterfall" and prefer more agile programs where if one part breaks, it won't screw up everything else. It's the end of class. I'll let you go! Thank you!	I don't think functional programming is any less agile than procedural style or object-oriented. It tends to be easier to prove formal properties of functional code, so that would tend to minimize the probability of errors creeping in.