

CSE143X Lecture Questions for Friday, 12/11/20

Time (e.g., 12:45)	Question	Answer
	<p>any chance we get to come back for your famous cookies in a future quarter</p> <p>We had a discussion in XA about the pizza party too, cookies would be a nice addition :D</p> <p>Thanks for a great quarter. +1</p>	<p>That's an interesting suggestion. 143x reunion when we're allowed to be on campus again. I'll think about it.</p> <p>Thanks for the kind words.</p>
	<p>Why did the lambda function not need to override the existing string comparison?</p> <p>I sort of understand...Arrays.sort has another parameter to fit a custom comparator function, is that it?</p> <p>I understand now, thank you.</p>	<p>That doesn't have to do with the lambda. That has to do with the call on Arrays.sort. In the first case, we call it passing just the array. Then it uses the existing comparison function. The second call passes a second parameter known as a "Comparator". We could have gotten the same effect without the lambda by saying: Arrays.sort(words, new LengthComparator());</p> <p>That's assuming you have that class called LengthComparator that is described in chapter 13 and that I was showing. This provides a custom comparison object that the Arrays.sort method uses instead of using the built-in comparison.</p> <p>Yes, you've got it. The Arrays.sort method is overridden with different versions. The version that takes one parameter uses the built-in comparison. There is another one that takes two parameters including an object that it uses to do the comparisons instead of using the built-in comparison.</p>

	<p>Is there a specific reason Java implements hashCode() but forces us to implement Comparable? Couldn't a generic comparable method be derived from comparing hashcodes?</p> <p>I think it would be useful for deterministically ordering items. (unless default hash codes are implementation specific)</p>	<p>Not all data can be compared. What if you had employee objects. How would you compare them? I guess they also felt that hashing was more fundamental than sorting.</p> <p>Hmmm...Comparison of hash codes. That would work in the sense that it would give you an answer, but I'm not sure it makes much sense.</p> <p>I think this is something where programmers are likely to have different opinions. For example, Python is willing to sort a list that has different kinds of data in it (numbers, strings, etc). I think that's weird and I've never liked it. But other people probably like it. So maybe this is something where it's a matter of personal choice.</p>
51:40	<p>I seem to remember you saying something about an optional lecture on some advanced topic; does that exist and if so, when?</p> <p>Yes, I see it there for today. For some reason, my browser's search feature didn't find it when searching the string "optional"</p>	<p>I mentioned that I had given optional lectures when I last taught cse143 in a fall quarter. The lectures were on Java 8 and quicksort. I've included a link on the calendar entry for today to that page (search for "optional").</p>