

tCSE143X Lecture Questions for Monday, 10/26/20

Time (e.g., 12:45)	Question	Answer
	I think you need to fix your mic	I've reported, but they haven't yet responded.
10	Why does Arrays have a toString method that we have to override to print correctly but for ArrayList it does the right printing?	I think it's basically a flaw of Java that arrays have such a bad toString method. ArrayList came later, so they were able to correct that mistake.
11	Why would you want to use an array over an arraylist? Seems that arraylist is much nicer to work with	ArrayList is nice to work with, but there are many cases where a raw array works just fine. The syntax is nicer for raw arrays.
26	Is there a way to get the index number while using forEach loop? Re: gotcha.	No. You could set up a counter that keeps track of it, but then you might as well use the standard traversal loop.
18:20	What is modCount in an ArrayList? Re: Haha ok, you predicted my next question. I'll look forward to it.	It keeps track of how many modifications have been made to the object. Wait until next week when we discuss iterators with a set and ask again what it is useful for.
	If size is a field in ArrayList, why do we have to access it with a size() method? In arrays we use length without () so why cant we do the same here? Makes sense thanks	You wouldn't want to give a client the ability to reset the value of size. For an array, it's a final field, so the client can't change it. For an ArrayList, it changes in value as you add to or remove from the structure.
	What happens when you do list.set(1,"foo"); list.set(9,"bar"); But then access something that you didn't set like list.get(6); Are they default initialized or return an error? Re: got it, thanks!	You can only call set on an index where a value is already stored. It's used to change a value, not to add something to the structure.
26:43	Screen blacks out for ~2 secs	
25:52	ForEach loop can be used for all objects right?	No. You have to do some special work to allow the foreach loop to work. That has been done for arrays and ArrayList.

10:18	<p>The indexOf method will return the first occurrence of the given value?</p> <p>Ok, thank you</p>	Yes (or -1 if not found).
28:10	<p>Does the iterators .next() return the value as a String or the same data type as the value itself?</p>	It returns one of the values from the list, so that value will be of the type used in defining the list (String for ArrayList<String>, Integer for ArrayList<Integer>, etc).
	<p>Possible to nest ArrayLists? Would the universe collapse into itself?</p> <p>Re: For halloween I'm dressing up as an ArrayList. Pretty Spooky.</p>	Yes, it's possible to do that and the universe seems to continue to exist. You could, for example, have an ArrayList<ArrayList<String>>.
33	<p>What is the difference between int and Integer except that one is primitive and one is an object?</p>	An Integer object has a single field of type int. They are stored differently. Integer is stored as a reference.
45	<p>Does ArrayIntList inherit from ArrayList?</p> <p>Okay, if it hypothetically did inherit from ArrayList and lets say that LinkedList did too, could you pass ArrayList instead of using interfaces?</p> <p>Okay thanks</p>	<p>No. ArrayIntList is a complete class on its own (no inheritance). It's similar to ArrayList<Integer> in what you can do with it.</p> <p>If you had a common class that both ArrayIntList and LinkedIntList inherited from, then you could use that common class as the generic type for both.</p>
39	<p>Do we ever use throws inside a method?</p>	No.
23	<p>Why do you use separate constructors to get the capacity and declare the elementdata/size?</p>	There is one primary constructor that does all of the work like allocating the array. It's the one that takes a capacity as a parameter. I then use the this(...) notation to have the other constructor call that constructor passing it the default capacity as a parameter.
	<p>What is the difference in using an ArrayIntList and an ArrayList<Integer> ?</p>	ArrayList<Integer> can do more. I wrote ArrayIntList as an educational exercise...it's something to study to understand how ArrayList<E> works.
33	<p>Is there a wrapper class Character</p> <p>Re: yeah you say it 20 seconds after. My bad...</p>	yes...I knew you were going to ask :-)

49	<p>When we implement the interface and call the add method for example, how does it automatically know to call the add method in that specific class(say ArrayList)?</p> <p>But after passing the list as a parameter to processList, doesnt it become of type IntList even though it points to the correct object?</p> <p>Got it thank you!</p>	<p>When you call a method like add, you will have a reference to the object you are working with. Each object knows its type, so it will know which add method to use.</p> <p>It doesn't matter what the type is of the variable that has the reference. The key thing is that the object itself knows its type (it stores it as a field).</p> <p>Think of it this way. I might write code for talking to a student. All I'd know is that it's some student. But when I'm talking to a specific student, that actual student knows lots of things about themselves, like whether they are grad or undergrad or what their name is.</p>
----	--	--