CSE143 Lecture Questions for Wednesday, 1/5/21

Question	Answer
Not sure if this is the right place to ask but for assignment 1 letter inventory can I assume O(n) is a good time complexity and O(n) is a good space complexity currently where we are required to use some data structure which may take up space. Using 2 nested for loops also caused O(n2) would we be marked down for such inefficiency	If you introduce an unnecessary inefficiency that was highly expensive, then you could lose points. The assignment specification also mentions specific things you have to do to guarantee efficiency.
For the letter inventory also we are called to construct and object inside an object class we are currently writing.	You are responsible for writing the LetterInventory class. That will involve deciding what fields are appropriate and then you will write a constructor that does appropriate initialization to make the object function properly. We don't tell you what to do…that's one of the things you're figuring out.
Is it safe to assume in general(after first week) hw will be assigned fri and be due next thurs	Most weeks it will be released on Friday and due the next Thursday.
Just to clarify more why are we using implementation rather than List<Integer> list = new ArrayList<>	In this homework you are exploring how such a structure is implemented. That's an important coding skill to master.
Is it possible to save these questions so we can look at them later for review?	I am doing that. Under the calendar tab you will find a "Q&A" link for Monday's lecture that has all questions asked in both lectures. I'll do the same today and on Friday.
We are required to group the letter inventories in alphabetical order when they are printed can we assume that doing binary search would be efficient or must I implement the comparable interface thanks. **Strings by default implement the compare To method does this mean I don't need to define my own such method**	You have to decide how to implement the behavior we have specified. It won't involve implementing the Comparable interface. You shouldn't be using a compareTo method for homework 1.
Will we be using public static void for the upcoming homework assignment? Or public void?	We have sort of switched gears from writing static methods to writing what are known as "instance" methods. All of the methods I have asked you to write for homework 1 are instance methods, so you won't use static. The only thing we are using static for is for constants.

What do you mean by a precondition in a method?	A precondition is something that must be true in order for the method to do its job. A classic example from math would be that if you want me to find the square root of a number, then it can't be negative. So a precondition for Math.sqrt would be that the value is greater than or equal to 0.
Why did you change the capitalization of toString to ToString? In the notes you mentioned that you did so, thus I got confused: "I changed the name of the method to ToString (different capitalization--this is actually what C# uses)." Got it! Thank you.	Where do you see ToString? I see what you're talking about. I didn't do that in today's lecture, but I've done it before. The point there is to show that if you use a nonstandard name like ToString, then you would have to make explicit calls on the method to make it work. If you use the standard toString, then Java will also make those implicit calls for you (e.g., when you call println and when you do string concatenation).
For next week, will the lectures also be uploaded to the site? If so, then how long after the lecture time the recording will be posted? Thanks!	Yes. I generally record the second lecture, so they would tend to be available around 4 pm MWF.

In the lecture you mentioned that we could have multiple constructors. Is that the same for the other public classes? If so, how does Java know which one you are referring to?
 	
If for example I was to write:
 	
Public void add(int value){...} and public void add(int value, int index){...}
 	
They would be different?
	The idea here is something called overloading. Java allows you to overload constructors and methods. That means having more than one. It's okay to do that as long as they have different signatures. The signature is the name of the method or constructor plus the number and type of its parameters. So if you're going to have multiple constructors, they need to have different signatures. For ArrayIntList, we have one that takes no parameters "()" and one that takes a single int "(int)".
	
	Yes, that's two different signatures that will be obvious when you call them, as in:
	   list1.add(18); // first of your methods
	   list1.add(3, 24); // second one

Can we write additional helper methods for the Homework assignment? Like a boolean method to check if a char is a letter, or an increment method? Or can we only write the specified methods in the assignment?	As I mentioned at the end of the lecture, you are allowed to introduce whatever methods you want as long as you make them private. We've indicated which public methods we want and you shouldn't add extra public methods. But add as many private methods as you like. Methods are good.

Normally, I use intelliJ. Is there a problem if I upload intelliJ files onto Ed instead of jGrasp?	That should be fine.
The learn remove method why do we start at index rather than the back of the list to the index	Chapter 7 discusses this. If you're shifting to the left in an array, then you want to go forwards. If you're shifting right, you want to go backwards.
Are there any Indentation tools you recommend for jGrasp?	There is an indentor tool that you can access from the homework tab
How would I upload a file into Ed from jGrasp?	Copy and paste works
Is there a list of exceptions that we should know to list in the precondition in case our method might throw that exception?	Preconditions and exceptions are not quite the same thing. Preconditions are just conditions that you tend to describe in English or with math (as in, index $>= 0$). An exception is a particular class of object that is thrown under particular circumstances (often when a precondition is violated). We will tell you exactly what type of exception to throw for each homework assignment.
Is there a particular reason why you don't want us to use try/catch? Even though we're writing methods that can throw Exceptions?	What a program should do to recover from an error is never simple. We just don't end up doing anything this quarter where try/catch would be helpful.
In 142 we put the exception in the method header when using scanners. What is the difference in putting it inside the method?	I understand the confusion, but these are two different things. In lecture today I talked about how you construct and throw an exception. That's using the "throw" command in Java. What you're remembering is that in 142 when we processed files, we had to include a special clause in the header using a "throws" notation (throws, not throw). We had to do that because we were dealing with what is known as a checked exception (FileNotFoundException). One is an action (throw) and one is a description (throws).

I took AP Computer Science instead of CSE 142. In that class, "break;" was a central part of writing while() loops. "while(true) {... if(x) {break;} }" was our default while() loop. Is there a reason why we aren't using "break;" in this class?	Some people like the programming style you have described, although it's somewhat controversial. I used to encourage that kind of loop myself, but most of my textbook reviewers hated it. But the answer is simpler. It's just a matter of picking what constructs I want students to use. These other features aren't necessarily bad, they're just not the constructs I have chosen to teach. I don't want you to have a different set of tools for writing loops than other students, so I'm asking you not to use break.
Is the name elementData for the int[] object in the ArrayIntList only a reasonable name for the ArrayIntList class or is it reasonable to name the int[] object in my LetterInventory class the same thing.	I used that name to match what the built-in ArrayList<E> class uses. It's up to you to decide what names to use for your homework.
For add(int index, int value) instance method, when we are running the for loop, since we initialized i = size in the beginning, wouldn't elementData[i] report an error because it is out of bound (or similar idea)? Is it ok, for an easier understanding, to increment the size before the for loop and write a slightly different for loop accordingly?	That involves adding a new value to the list, so you're going to have to shift everything right to make room for it. So elementData[size] isn't being used yet, but now we need to use it. In the final version I posted on the calendar, it checks to make sure that there is enough capacity to add a value and it throws an exception if not. It's a matter of personal choice whether you'd update size before or after the loop.
Do you know what happened to the syllabus? It says it was removed.	There is an issue that needs to be resolved. I'm working on it.
Is every homework assignment released after all of the content needed to complete it has been covered?	Yes, that is something that I consider very important. I will always have covered what you need for a homework when I release it.
Is there an easier way to generate test data for ArrayintList? In class, are we going to implement the static initialization constructor?	You can look over the testing programs that we are providing. You can use an array initializer if you want in testing code you'd write for yourself.
Should we use Array.copyOf to resize the array and copy every element over? When we want to resize the array when we run out of room in our original array?	I will show you a version of ArrayIntList later that does something along those lines. You should not use it for your own homework solution.

Are we implementing an iterator for arrayIntList similar to javas array list?	We will discuss that in a later lecture.
Do you recommend commenting like this /* @throws- @Paramter- */	You're talking about JavaDoc comments. You can use that form if you want to. I find that they aren't easy to read in text form, but they make nice html files. We don't have the time to devote to JavaDoc, but they are obviously useful.
Also last question does java support shorthand circuit analysis where if it knows the answer it does not evaluate what's left in the if condition.	The && and || operators use short circuit evaluation.