CSE143X Lecture Questions for Monday, 11/9/20

Time (e.g., 12:45)	Question	Answer
49	Your thoughts on Java's lack of string multiplication? Good or bad? I cry every time I need to write a for loop instead of " " * 4 :(Java doesn't have operator overloading? Bummer. Petition to redesign intro programming courses to use C# or Python pls. Coming from C++, I think Java having some limitations is a good thing sometimes :D Truly ahead of the times. Petition to do 142 in Java and 143 in Javascript just to confuse people. :D What do you think of the newer languages for this then? Like Go or Rust. (or Scratch :p)	Python introduced that syntax to most programmers. It's nice to have available, but Java in general hasn't emphasized that as much. I'd rather have operator overloading the way they did it in C#. In C# you can use array-style bracket notation for maps, which is much more intuitive. The argument against is that operator overloading is a mess in C++ and I tend to agree (too loose). C# has a nice compromise. Hmmm...try to argue that C# could replace Java in intro. Who might have written about that? https://www.researchgate.net/publication/215900276_Can_C_replace_java_in_CS1_and_CS2 Stanford does 142 in Java and Python and 143 in C++. Pretty confusing. I still think Java is the least bad choice overall, but my coauthor Marty Stepp thinks we should teach JavaScript and others I know think we should teach Swift.
46	How careful should we be with preventing infinite recursion? In your file-system example, depending on how Java handles symbolic links, the recursion could be infinite.	It's always good to think about situations that could lead to infinite recursion. We'll be very clear in our assignment specifications about what cases we want you to consider.

30:00	The reason we don't use index++ is because it won't increment until after the recursive call - right?	In general you don't increment variables the way you do in an iterative solution. You instead pass on the updated value. But you're right that it won't work because you'd have to say ++index to get it to work right or do the increment before the method call. But why increment it at all? That particular call has been asked to deal with a specific index. Why change it?
32	How can you use two sum methods? Wouldn't java be confused on which sum method to call? I see thanks.	We've discussed this before. You can overload method names in Java if the different methods have different signatures (signature = name and number and type of parameters). One sum method has a single parameter of type int[]. The other has two parameters of type int[] and int. It is easy to figure out which method to call.
	Why wouldn't writeBinary print the number backwards? Wouldn't the base case be printed first, and then each recursive case be printed in backwards order? Oh I see, I had the wrong understanding of how the binary number would be represented. Thank you	No. Notice how in the recursive case we first recursively print the binary representation of n/2. All of those digits will be written out before it gets to the print command that prints n % 2. Think of the call on writeBinary(6): Calls writeBinary(3, which is 6/2) Calls writeBinary(1, which is 3/2) Prints 1 (base case) Prints 1 (which is 3 % 2) Prints 0 (which is 6 % 2) So it prints 110.