

CSE143 Lecture Questions for Monday, 4/26/21

Question	Answer
<p>Is it usually clear what situations to use iterative and what situations to use recursive to solve a problem?</p> <p>Hm Interesting, thank you!</p>	<p>I'll mention a few ways to help figure that out. Recursion tends to be most helpful when the underlying structure you're working with is recursive. You also want to be careful with recursion not to let the stack depth get out of hand, so that limits some of the uses we might otherwise choose for recursion.</p>
<p>What dates will the midterm be available? And what topics will be on it? I could not find this information on the syllabus.</p> <p>Thanks</p>	<p>I will be posting information next week.</p>
<p>Is there a way we can keep count of how many times we've recursed? Or is that just iteration?</p>	<p>We'll see an example of that in Wednesday's lecture. Basically you include an extra parameter that keeps track of the count.</p>
<p>When will we have our 1st exam?</p> <p>Thank you</p>	<p>I will be releasing information about that next week.</p>
<p>About the count of how many times we've recursed won't we have to use a field for this? Isn't this bad style?</p>	<p>That would generally be bad style to use a field to keep track of something like that.</p>
<p>In the "this is fun no?" example you did, there was a line of code which said <code>String line = input.nextLine();</code> How does Java store these multiple data fields with the same name throughout the recursive calls?</p> <p>Okay thank you!</p>	<p>We've already seen Java doing something like this. You can have a variable in a method like main and a variable with the same name in another method. Here the only unusual thing is that both methods have the same name. Basically Java has the ability to set aside space for each method call, including multiple calls on a single method, that allows it to store those local variables.</p>