

CSE143X Lecture Questions for Monday, 11/23/20

Time (e.g., 12:45)	Question	Answer
	Stuart stopped saying “ookay let’s go ahead and get started”	Could fix that starting in the next lecture. :-)
	<p>It’s a good philosophical question though, “where is outside truly?” All relative.</p> <p>It’s always “where is outside” but never “how is outside?” :(</p>	
5:00	<p>Would those empty trees simply be empty references? Why would we call them trees, if the value is null?</p> <p>Okay - I suppose I assumed that an empty tree might just be a “zero-equivalent” tree, like a node with value 0, and two null subtrees.</p>	<p>Remember that an empty tree is a tree. It makes it easier to write compact recursive code to interpret the tree this way.</p> <p>Some people don’t like using “null” as the representation of an empty tree and they introduce objects that are empty trees. I prefer to keep it simple by using null, but it can be confusing, as you suggest.</p>
	<p>I appreciate the effort put into the scooch. Oooh.</p> <p>That was pretty epic +1</p>	Thanks.
15:00	Is it generally good practice to use the same method name when making a private helper method with a different # of params?	It’s really a matter of personal choice. Some people like different names. I prefer using the same name. Either is fine.
	<p>If there’s a public/private pair with the same parameter list is that considered the same signature?</p> <p>I see, so the classifier doesn’t affect it at all? Thanks!</p>	The signatures have to be different, so if the number and types of parameters are the same, then they can’t have the same name. The public/private is not part of the signature.

	<p>If a binary tree node is technically a tree, why do we need two classes: XXXTree and XXXTreeNode? Is it just for aesthetic reasons?</p> <p>But those statistics would be equally valid for any of the child nodes.</p> <p>Okay :)</p>	<p>There might be “tree level” statistics that we want to keep track of in the IntTree class. For example, keeping track of the size of the tree. So it makes sense to have tree-level and node-level.</p> <p>If you have a tree with 1000 nodes, you don’t want 1000 copies of that value. Having one tree object that keeps track of one occurrence of 1000 makes more sense.</p>
	<p>Last bit got cut off.</p>	<p>Just mentioned that we’ll be looking at concepts that will be part of the final exam in Wednesday’s lecture.</p>
	<p>“Line”-us Torvalds Ah yes inventor of Linux</p>	<p>Yes.</p>
<p>30:11</p>	<p>If there allows duplicates in binary search tree. A</p> <pre> / B \ A </pre> <p>The adding order is right? Why are the two As not adjacent? How about BCAB?</p> <p>If two Bs are interrupted by A, how can the treeSet realize sorted?</p> <p>Oh yes, inorder can do that. Thx!</p>	<p>If you add A, B, A, you get:</p> <pre> A / \ A B </pre> <p>If you add B, C, A, B, you get:</p> <pre> B / \ A C \ B </pre> <p>The tree that you get from this process would not have an A in between two Bs. An inorder traversal of the tree above would give you A, B, B, C</p>