

CSE143X Lecture Questions for Monday, 10/5/20

Time (e.g., 12:45)	Question	Answer
3:15	Would 'final' be equivalent to 'const' in other languages? Re:thanks!	Yes, but const has many other meanings in C++. The sense in Java is very simple: you can't alter the value of this variable.
49:00	The graphics produced by g.fillOval() is centered around the x coordinate, but aligns to the left coordinate, is that right?	No. The (x, y) coordinates are for the upper-left corner of a rectangle (the containing rectangle). It draws the biggest oval that fits inside that rectangle.
13:20	Is it common to use local constants or is that usually not done?	Local constants are not commonly used. It's almost always class constants.
	What is the creative aspect of this project (Cafe Wall) that you mentioned in the lecture?	Look at the assignment writeup; doodle (part A). Available from the homework tab.
	What is import java.awt.* doing when you run it? (literally)	It instructs the compiler to look in the java.awt package for any classes that otherwise are not found. So when you refer to Color, Graphics, Font, etc, it finds them in the awt package because they aren't defined locally.
	So we import packages when we run our program but not when we submit hw? In that case do we have to run it, edit out the import statement and save it again?	You always keep the import declaration. Otherwise your program won't compile on our server. It is a way to have one class have access to classes defined in a library like the awt package.
	Is there a time limit for how long Doodle.java can take to produce one frame or a full animation?	Nope. Whatever you want.
	Does the graphics class' functions appear the same way when called without the drawing panel? Do the drawings appear on the console?	Graphics is a standard Java class that is well documented. It always behaves the same way. But there isn't a way to tie a Graphics object to the console because the console is text only.
	So in that case where will the objects that we draw appear?	You can only get a Graphics object that is associated with a graphical environment. It just won't work to draw to the console.

14:22	<p>Why wouldn't you want the int to be specifically tied to the row variable? Re: ok thank you!</p>	<p>If you have the method take the row value, then it is useful just for this specific task. A more generic writeSpaces method could be used in many programs, including programs that don't have a row variable. We like to write methods that can be used in more applications. No, not to reduce redundancy. To increase the ability to reuse the method in other programs.</p>
39:00	<p>Why do we need to put in DrawingPanel before panel rather than just do panel = new DrawingPanel. Re: ahhh! Python spoiled me.</p>	<p>You are declaring a variable called "panel" and you have to tell Java what type of data that variable will store. Yes, Python is much more relaxed.</p>
39:19	<p>Does the DrawingPanel create a new Graphics object every time we call panel.getGraphics()?</p>	<p>No. The DrawingPanel creates a Graphics object when it is constructed and gives you a reference to that object every time you call getGraphics.</p>
41:29	<p>Can we just store the Graphics object in a variable instead of calling panel.getGraphics every time? Got it thanks</p>	<p>Yup...wait for it...</p>
	<p>Suppose we're halfway through drawing our car and we create another Graphics object, say g1, calling the rest of the methods with g1 would not make any difference since it would be storing the same reference as g right?</p>	<p>If you have two variables that have been set by calling getGraphics on the same DrawingPanel, then yes, that's just two references to the same object and it won't matter which one you use. But if, for example, you had two DrawingPanel objects and you called getGraphics on each, then those would be different objects drawing to the different panels.</p>
	<p>This is just a style question, but is it okay to leave blank spaces separating for loops and if statements from the rest of the program? Just to separate out some of the logic</p>	<p>It's a matter of personal choice. I tend to include blank lines if I think there is a new subtask being performed.</p>
44:21	<p>How come when we called panel.setBackground, we didn't have to call getGraphics()? Makes sense Thank you</p>	<p>The panel has certain methods that you can call including setBackground. It does certain things. Other things are done by the Graphics object and it has its own methods for those. You don't need a Graphics object to set the background of a panel. That's just how it works.</p>

	<p>Why do we have to clear the panel and how does 10 milliseconds come into play? The car moves for a longer</p>	<p>An animation involves showing a series of frames that are slightly different as time goes on. You have to call clear to erase the previous frame before you draw the new one. If you don't sleep, then the drawing tends to happen so fast that the human eye can't perceive it. You have to pause to give the human time to catch up.</p>
	<p>I'm using IntelliJ to follow along and after introducing Graphics g = panel.getGraphics, it gives me an error saying "java: incompatible types: java.awt.Graphics2D cannot be converted to Graphics"</p> <p>Is this a problem on my end? The code is the same as the lecture.</p> <p>Re: Yes is that why.</p> <p>Ok thank you - I will try that</p>	<p>I know what the problem is. Did you name your class Graphics?</p> <p>If you have a class called Graphics, then Java won't use java.awt.Graphics. You have to call it something else and delete or rename Graphics.java and Graphics.class.</p>