



11/30/2007





# Opening Discussion

- Do you have any questions about the quiz?
- Let's look at solutions to the interclass problem.
- Do you have any questions about the project?





- Setting up GUIs is nice, but for some applications you need to draw things that aren't part of the standard GUI library. For that we need to draw our own graphics.
- The classes for drawing graphics are in `java.awt` and the packages under it. We will also use some help from Icon code in `javax.swing`. In particular the `ImageIcon` class.



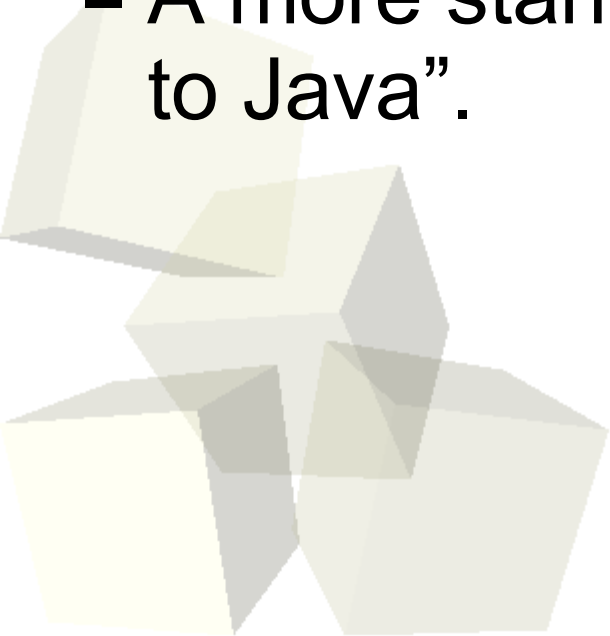


# Graphics and Graphics2D

- The classes we use to actually draw are the `java.awt.Graphics` and `java.awt.Graphics2D` classes.
- `Graphics2D` inherits from `Graphics` so it has all the capabilities of `Graphics`, plus extras.
- For our purposes we only need to worry about `Graphics2D`.
- Let's look at the API for some of the key methods.
  - ◆ `draw(Shape s)`
  - ◆ `fill(Shape s)`
  - ◆ `setPaint(Paint p)`
  - ◆ `setStroke(Stroke s)`
- Shapes are in `java.awt.geom`.



- A simple way to get our graphics onto the screen is to make a `BufferedImage`, use that in an `ImageIcon`, then have a `JLabel` in the GUI that displays the `ImageIcon`.
- The `BufferedImage` has a method called `createGraphics()` that gives us a `Graphics2D` object we can draw to to change the image.
- A more standard method is described in “From C to Java”.





- Let's spend the rest of the time coding. We should write up a little bit of code that makes an image and draws it in a GUI.
- We should also try to attach a listener to something. Making something that draws when we click on the image would be a good application.





- Do you think it would have been helpful to do GUIs and graphics earlier or did they fit well with the flow of topics where we had them? (Note that inheritance is significant for GUIs.)
- Interclass Problem – Write a program that shows three shapes of different colors in a window.

