Graphics
Computer Graphics vs. Graphic Design

• Computer Graphics is not using Photoshop- it’s learning how to MAKE Photoshop.

• Within CS, computer graphics is the study of how to make a computer render images.

• We usually focus on 3D images, but it also encompasses 2D image processing.

• It involves a lot of code, and a lot of math (particularly geometry and matrix algebra), but the rewards are worth it.
Some applications of graphics

- Cinematic CG
- Animation & Scientific Modeling
- Video games
- Photorealistic image rendering
- Image editing tools
- Java Applets
Cinematic CG
Video Games
Animation and Scientific Modeling

http://www.youtube.com/watch?v=E37Ss9Tm36c
Photorealistic Images
Image editing tools
Most of the really cool stuff you can do with Graphics requires a very solid background.

One of the more basic forms of graphics that you can do is in Java applets and applications.

Many times when writing an applet, you’ll want to include graphics or animations that the user can interact with (such as an interactive chess game).
If you’ve never programmed in Java before, don’t worry about it! We’ve provided and explained all the commands that you will need to know for this project.

Ignore most of the code- All of the code that you add to draw the graphics will be placed in the paint method.
Your objective
The coordinate system

- When you specify where to draw an element, you are telling the computer the pixel locations of where you want to put the shape.

- The upper left hand corner is the origin, (0,0).

- The x axis increases as you go across

- The y axis increases as you go down.
For your convenience, we included the startX and startY variables to serve as an easier reference point for you to use.

So instead of going through the trouble to specify the exact pixel coordinates, you can just say startX+___ and startY+____

Both have been initialized to 250. Feel free to change these or not use them at all.
g.setColor(Color.BLACK);

- Use this command whenever you want to change colors. The next items you draw will be in that color until you change the color again.

- If you wanted to change the color to blue, you would type `g.setColor(Color.BLUE);`

- Or for red, `g.setColor(Color.RED);`

- And so on. Note that the name of the color must be in all capitals
g.fillOval

- Draws an oval (or circle) and colors it in.

- FORMAT: \( g.fillOval(x_{position}, y_{position}, width, height) \). All in units of pixels.

- A circle is just an oval where \( width = height \)

- Sample: \( g.fillOval(startX+5, startY+5, 30, 30); \)
g.fillArc

- Draws and colors in an arc. Think of a pie chart
- FORMAT:
  - g.fillArc(xposition, yposition, width, height, startAngle, arcAngle)
  - startAngle = the beginning angle
  - arcAngle = the extent of the arc, relative to the beginning angle.
If you finish early…

Feel free to add to your picture or even make a new one!

There are many more Java graphics tools you can use.

This site lists and explains all of them.

http://download.oracle.com/javase/1.4.2/docs/api/java/awt/Graphics.html