**Threads, Executors, Runnables, Handlers**

* Thread is a program’s path of execution
* Typically, programs will only be able to do one thing at a time if they run on a single thread, but if you have a multithreaded program, it is able to run multiple options at the same time and run more smoothly

- Classes vs. Interfaces: There are a few differences between a class and an interface. First, an interface can only contain abstract methods and/or static final variables (constants). Classes, on the other hand, can implement methods and contain variables that are not constants. Second, an interface cannot implement any methods. A class that implements an interface must implement all methods defined in that interface. An interface has the ability to extend from other interfaces, and (unlike classes) can extend from multiple interfaces. Furthermore, an interface cannot be instantiated with the new operator; for example, Runnable a=new Runnable(); is not allowed.

* Runnable is an interface of the Thread class and is used in many cases when implementing a thread
* Threading uses different commands, such as start(), stop(), init(), resume(), sleep,() suspend() to control the flow of a thread

EXAMPLE CODE: public class CounterThread2 extends Applet implements Runnable

{

Thread t;

int Count;

boolean suspended;

public boolean mouseDown(Event e,int x, int y)

{

if(suspended)

t.resume();

else

t.suspend();

suspended = !suspended;

return true;

}

...

* }
* Java has a built-in scheduler that monitors threads and determines the order of processes that run
* You can set a priority level to each thread to determine when it should run in the sequence