Threads in Java

In concurrent programming, there are two basic types of execution: processes and threads.

There are two ways to create and use threads.

1. Declare a class to be a subclass of **Thread**. This subclass should override the run method inherited from the Thread class.
2. Declare a class that implements the **Runnable** interface. This class should implement the run method.

Class extends the Thread class:

```
public class MyThread extends Thread {

    MyThread() {
     // Initialize the instance variables
    }

    public void run() {
        // do something until the thread stops
        // . . .
    }
}

public class ThreadApp {
     MyThread thread = new MyThread();
     thread.start(); // starts the thread’s run method
}
```

Class implements the Runnable interface:

```
class MyApp implements Runnable {
    MyApp() {
        // Initialize the instance variables
    }

    public void run() {
        // do something until the thread stops
         . . .
    }
}
```

Note: Methods (other than constructors) that will be called by a thread should be synchronized. To make a method synchronized, simply add the synchronized keyword to its declaration.

```
public synchronized void push(T value){ … }
```

Some of the methods in the Thread class: http://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html

Thread	
+interrupt() : void	Send an interrupt to the thread
+interrupted() : boolean	Check to see if thread was interrupted
+sleep(long t) : void	Causes the thread to sleep (pause) for a specified number of milliseconds
+isAlive() : boolean	Check to see if thread is active (running)
+join() : void	Waits for the thread to die
+start() : start	Starts the thread – calls the run method
+**run**() : void	The body of the thread (***override***)