Thread States

* A java thread is always in one of several states which could be running, sleeping, dead, etc
* States:
* New thread
* Runnable
* Not Runnable
* Dead

**Now Threads:**

* A java thread is always in one of several states which could be Running, sleeping, dead, etc
* A thread states life in the Ready to run state
* You can call only the start() or stop() method when the threads is in this states
* Calling any method besides start() or stop() causes an IllegelThreadException (A descendant class or RuntimeException )

**Runnable:**

* When the start() method is invoked on a new Thread() it gets to the runnable state or Running state by calling the run() method
* A Runnable thread may actually be running or may awaiting its turn to run

**Not runnable:**

* A thread becomes not runnable when one of the following four events occurs
* When sleep() method is invoked and it sleeps for a specified amount of time
* When the suspend() method is invoked // not use
* When the wait() method is invoked and the thread waits for notification of a free resource or its for the completion of another thread or waits to acquire a lock of an object
* The thread is blocking on I/O and waits for its completion

**Switching from not runnable to runnable:**

* If a thread has been put to sleep, then the specified number of milliseconds must elapse ( or it must be interrupted )
* If thread has been suspended, then its resume() method must be invoked
* If a thread is waiting on a condition variable, whatever object owns the variable must relinquish it by calling either notity() or notifyAll()
* If a thread is blocked on I/O, then the I/O must complete.

**Dead state:**

**A** thread enters this states when the run() method has finished executing or when the stop() method is invoked. Once in this state, the thread can’t ever run again.