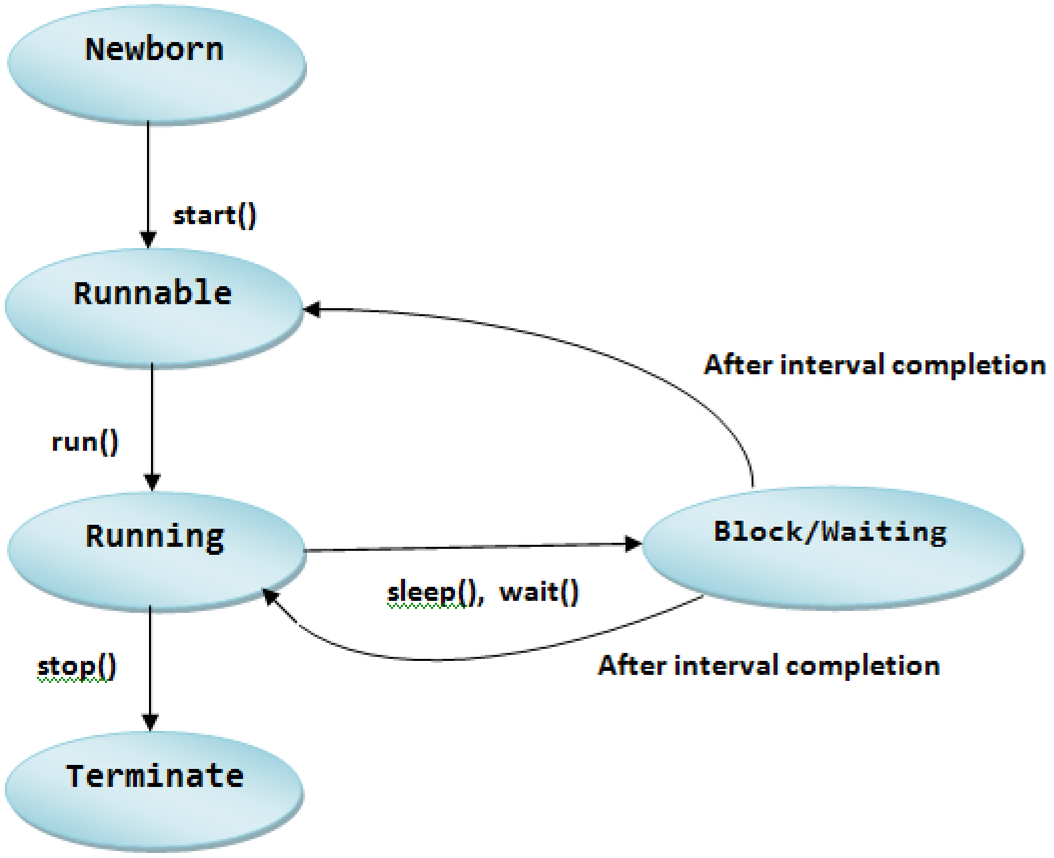
Life cycle of a Thread (Thread States)

A thread can be in one of the five states. The life cycle of the thread in java is controlled by JVM. The java thread states are as follows:

1. Newborn
2. Runnable
3. Running
4. Non-Runnable (Blocked)
5. Terminated



1) Newborn state:

When we create the thread object, the thread is born and is said to be in *newborn* state. The thread is ready to execute not running. At this state we can do only one of the following things with it.

- Ready for running using **start( )** method.
- Kill it using **stop( )** method.

MyThread t= new MyThrtead() → Newborn state

In this state, system resources are not yet allocated to the thread. When a thread is in the newborn state, calling any method other than **start()** method causes an **IllegalThreadStateException**.

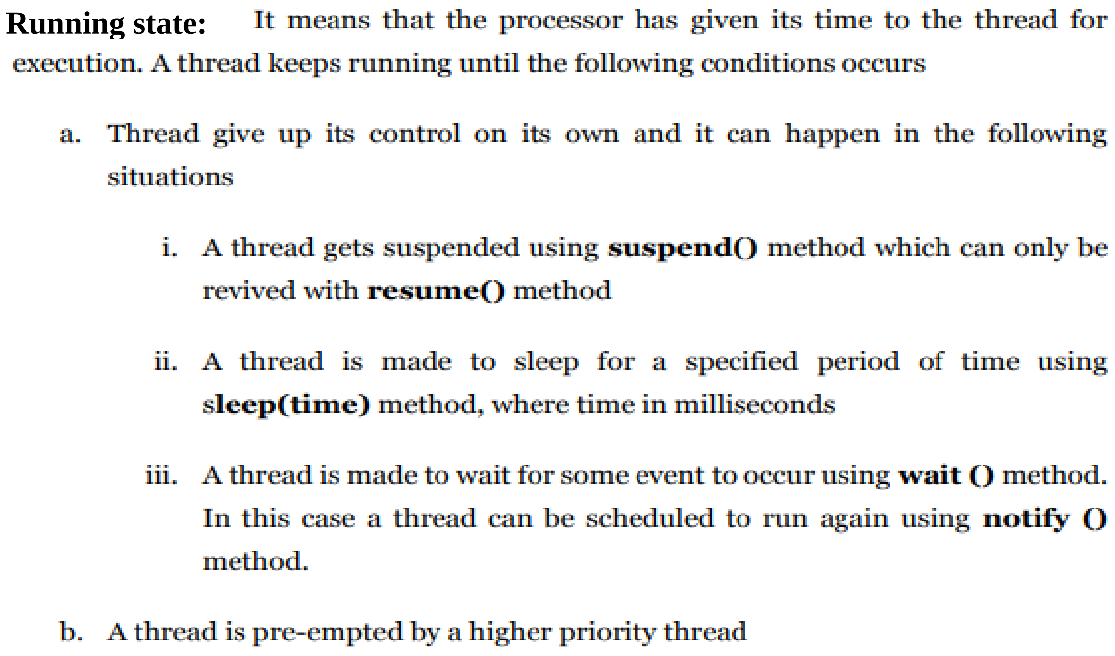
2) Runnable state:

A thread in the runnable state is ready for execution but is not being executed currently. Once a thread is in the runnable state, it gets all the resources of the system (as, for example, access to the CPU) and moves on to the running state.

All runnable threads are in a queue and wait for CPU access. When the **start( )** method is called on the newborn thread, it will be in the runnable state.

3) Running state:

After If thread scheduler allocates CPU for particular thread. Thread goes to running state. The Thread is running state means the **run( )** is executed.



Running state: It means that the processor has given its time to the thread for execution. A thread keeps running until the following conditions occurs

a. Thread give up its control on its own and it can happen in the following situations

   i. A thread gets suspended using **suspend()** method which can only be revived with **resume()** method

   ii. A thread is made to sleep for a specified period of time using **sleep(time)** method, where time in milliseconds

   iii. A thread is made to wait for some event to occur using **wait ()** method. In this case a thread can be scheduled to run again using **notify ()** method.

b. A thread is pre-empted by a higher priority thread

4) Blocked state:

If the running thread got interrupted of goes to sleeping state at that moment it goes to the blocked state.

5) Terminate state:

Every thread has a life cycle. A running thread ends its life when it has completed executing its **run( )** method. It is a natural death. However, we can it by sending the stop message to it at any state this causing a premature death to it. A thread can be killed as soon as it is born, or while it is running or even when it is in blocked state.