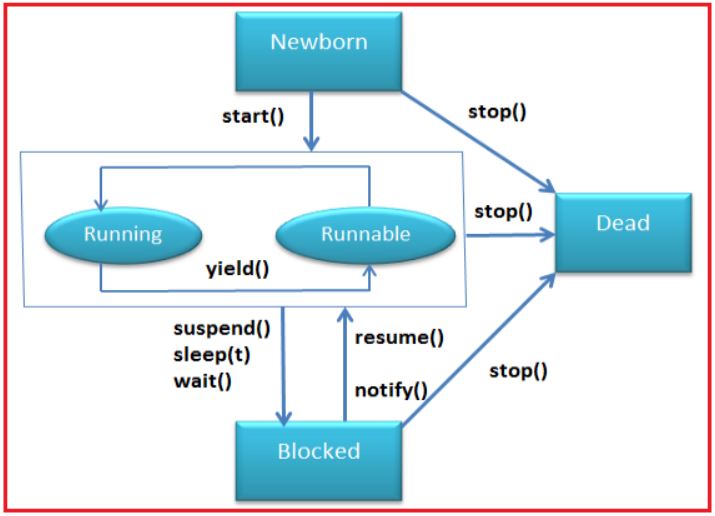
**Thread Life Cycle in Java**

A thread goes through various stages in its life cycle. According to Sun, there are only 4 states in the thread life cycle in java new, runnable, non-runnable, and terminated. There is no running state. But for a better understanding of the threads, we are explaining it in the 5 states. The life cycle of the thread in java is controlled by JVM.

For a better understanding please have a look at the below diagram. The java thread states are as follows:

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



**Following are the stages of the life cycle –**

**New –** A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread. It is also referred to as a born thread. In simple words, a thread has been created, but it has not yet been started. A thread is started by calling its start() method.

**Runnable –** The thread is in the runnable state after the invocation of the start() method, but the thread scheduler has not selected it to be the running thread. A thread starts life in the Ready-to-run state by calling the start method and waiting for its turn. The thread scheduler decides which thread runs and for how long.

**Running –** When the thread starts executing, then the state is changed to a “running” state. The scheduler selects one thread from the thread pool, and it starts executing in the application.

**Dead –**This is the state when the thread is terminated. The thread is in a running state and as soon as it is completed processing it is in a “dead state”. Once a thread is in this state, the thread cannot even run again.

**Blocked (Non-runnable state):**

This is the state when the thread is still alive but is currently not eligible to run. A thread that is blocked waiting for a monitor lock is in this state.

A running thread can transit to one of the non-runnable states depending on the situation. A thread remains in a non-runnable state until a special transition occurs. A thread doesn’t go directly to the running state from a non-runnable state but transits first to the Ready-to-run state.

The non-runnable states can be characterized as follows:

1. **Sleeping**:- The thread sleeps for a specified amount of time.
2. **Blocked for I/O**:- The thread waits for a blocking operation to complete.
3. **Blocked for Join completion**: – The thread awaits completion of another thread.
4. **Waiting for notifications:** – The thread awaits a notification from another thread.
5. **Blocked for lock acquisition**: – The thread waits to acquire the lock of an object.

JVM executes a thread based on its priority and scheduling.