

## Java Thread States

A Java Thread can be found in 6 different states.

1. New
2. Runnable
3. Blocked
4. Waiting
5. Timed Waiting
6. Terminated

**New** – A thread has created. But it still not active.

**Runnable** – This means an active Thread. It can be either running or possibly running thread. It decides by the CPU.

**Blocked** – This not a permanent stop. When the thread is stopped for a span of time, we call it a blocked thread. This happens because of a synchronized lock. (because of another thread)

**Waiting** – This state is also a temporary stop. If the thread is waiting for something to happen it's in waiting state.

**Timed Waiting** – Thread is stopped for a specific given time.

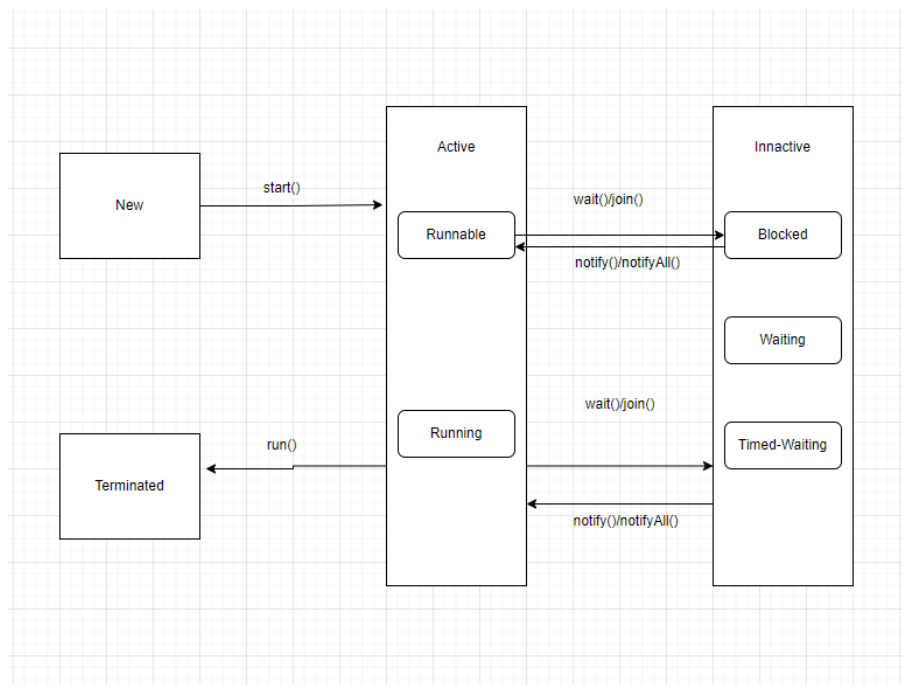
**Terminated** – This is the permanent end of the thread. This can happen after completing the execution or after getting an exception.

Initial State → New

Runnable (Active) → Ready to Run, Running

Non-Runnable (Inactive) → Blocked, Waiting, Timed Waiting

End State → Terminated



## Code Appendix

```

/**
 * Test
 */
class Test implements Runnable {
    public void run()
    {
        // moving thread2 to timed waiting state
        try {
            Thread.sleep(1500);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("4. thread state: " +
MyThreadClass.myThread.getState());
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

```
}  
public class MyThreadClass extends Thread{  
    public static Thread myThread;  
    public static Test obj;  
  
    // initiating the thread and its in NEW state  
    public void run () {  
  
        try {  
            Thread.sleep(1000); // Thread is in Timed Waiting state  
  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        System.out.println("3. thread state: " +  
Thread.currentThread().getState());  
  
        try {  
            Thread.sleep(200);  
        }  
        catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
  
    public static void main(String[] args) throws InterruptedException {  
        myThread = new MyThreadClass();  
        System.out.println("1. thread state: " + myThread.getState()); //Still  
the thread is in NEW state  
  
        myThread.start(); // now the thread is in RUNNABLE state  
        System.out.println("2. thread state: " + myThread.getState());  
  
        myThread.join();  
        System.out.println("4. thread state: " + myThread.getState());  
    }  
}
```

**Output:**

```
1. thread state: NEW  
2. thread state: RUNNABLE  
3. thread state: RUNNABLE  
4. thread state: TERMINATED  
PS D:\UoW\yr4\concurrent> █
```