

# Multithreading in Java – Part 2

## Thread - States

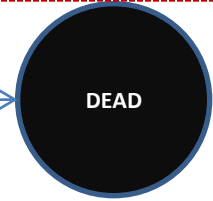
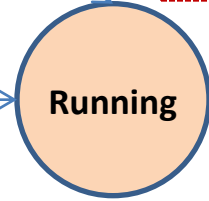
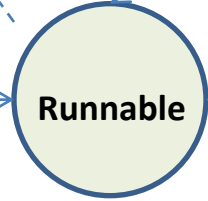
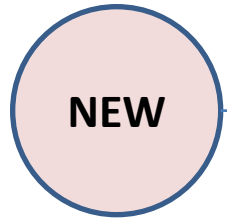


By,  
Srinivas Reddy.S

# Threads – States

When start() method is invoked on thread It is said to be in Runnable state. But it is not actually executing the run method. It is ready to run.

t.start()



Thread t = new Thread()

When a new thread object is created, the Thread is said to be in new state.

In Running state, the thread is actually executing the code.

The thread has finished executing the run method. It moves to Dead state.  
Note: You cannot use t.start() again on this object once the thread is dead.

t.sleep(long x)

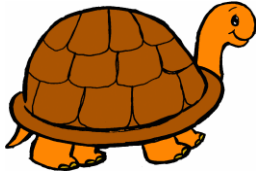
- When the sleep method is invoked, the thread may go to sleep state.
- In sleep, wait or blocked state, it is not running any code. But the thread is alive.
- A thread can move to Wait or blocked state any time.
- A thread in sleep, blocked or wait state can move to Runnable or Running state.
- Invoking sleep method will not guarantee that the thread will go to sleep state immediately

# Thread – sleep()

- A method used to tell the current thread to sleep for certain time.
- Sleep method accepts time in milliseconds
- Can throw InterruptedException.
- Cannot guarantee that a thread goes to sleep for specified time.
- Once the sleep state is complete, the thread can move to the Runnable or Running state.

# Sleep Demo- Demonstration

## The Tortoise and Hare story



- Tortoise and Hare Join the race.
  - **Hare sleeps in the mid of Race** thinking its too faster than tortoise.
  - Tortoise continues to move even slow and wins the race.
- **Slow and Steady wins the race.**

Lets implement this story to demonstrate Creating Threads

# Thread - Priorities

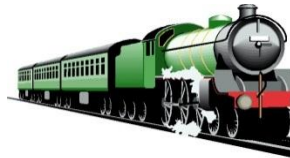
- All the threads created in Java carry normal priority unless specified.
- Priorities can be specified from 1 to 10.
- The thread with highest priority will be given preference in execution. But no guarantee that it will be in the running state the moment it starts.
- The currently executing thread will have the highest priority when compared to the threads that are there in the pool.
- Thread scheduler decides on which threads have to be given chance.
- `t.setPriority()` can be used to set the priorities on a thread object.
- The Priority should be set before the threads `start()` method is invoked.
- `Thread.MIN_PRIORITY`, `Thread.NORM_PRIORITY`, `Thread.MAX_PRIORITY` can be used in the threads `setPriority()` method.

# Thread – yield()

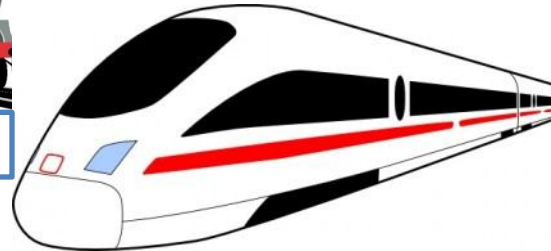
- **yield() method tells the currently running thread to give chance to other threads with equal priority in the thread pool.**
- **There is no guarantee that yield() will make the currently executing thread to go to runnable state.**
- **Remember, yield() will not make the thread to go to wait or blocking state. At the most, it will make a thread to move from running to runnable state.**

# Thread – Join()

- The `join()` method indicates that the currently running thread should run after the thread(on which `Join` is invoked) completes.
- Guaranteed that the currently executing thread will stop its execution and will run after the thread(on which `join` is invoked) is complete.



Just like one train waits for other to cross



# Thread – join() demo



# Deprecated Methods

- `stop()` – It is not possible to stop a thread once started.
- `suspend()`
- `resume()`
- `destroy()`

Instead of the above methods, programmers should try utilizing the following:

- `sleep()`
- `interrupt()` – used to interrupt a thread which is in sleep state



# **\*Test yourself\***

## **Relay Running Race**

- Create an application which simulates the running race of 400 meters.
- Create five thread groups and give names(Country names).
- The number of runners should be ten(two in each group) and give names to each runner thread.
- Each thread should run exactly half the distance – 200 m and the next thread in same group should join the race to complete it.
- Print the winner group name()and all the threads should complete the race.
- Print the time taken by each Group to complete the race and highlight the Winners time.

[www.JAVA9S.com](http://www.JAVA9S.com)

[Srinivas.java9s@gmail.com](mailto:Srinivas.java9s@gmail.com)



[facebook.com/java9s](https://facebook.com/java9s)



[@java9s](https://twitter.com/java9s)



[JAVA9s](https://youtube.com/JAVA9s)