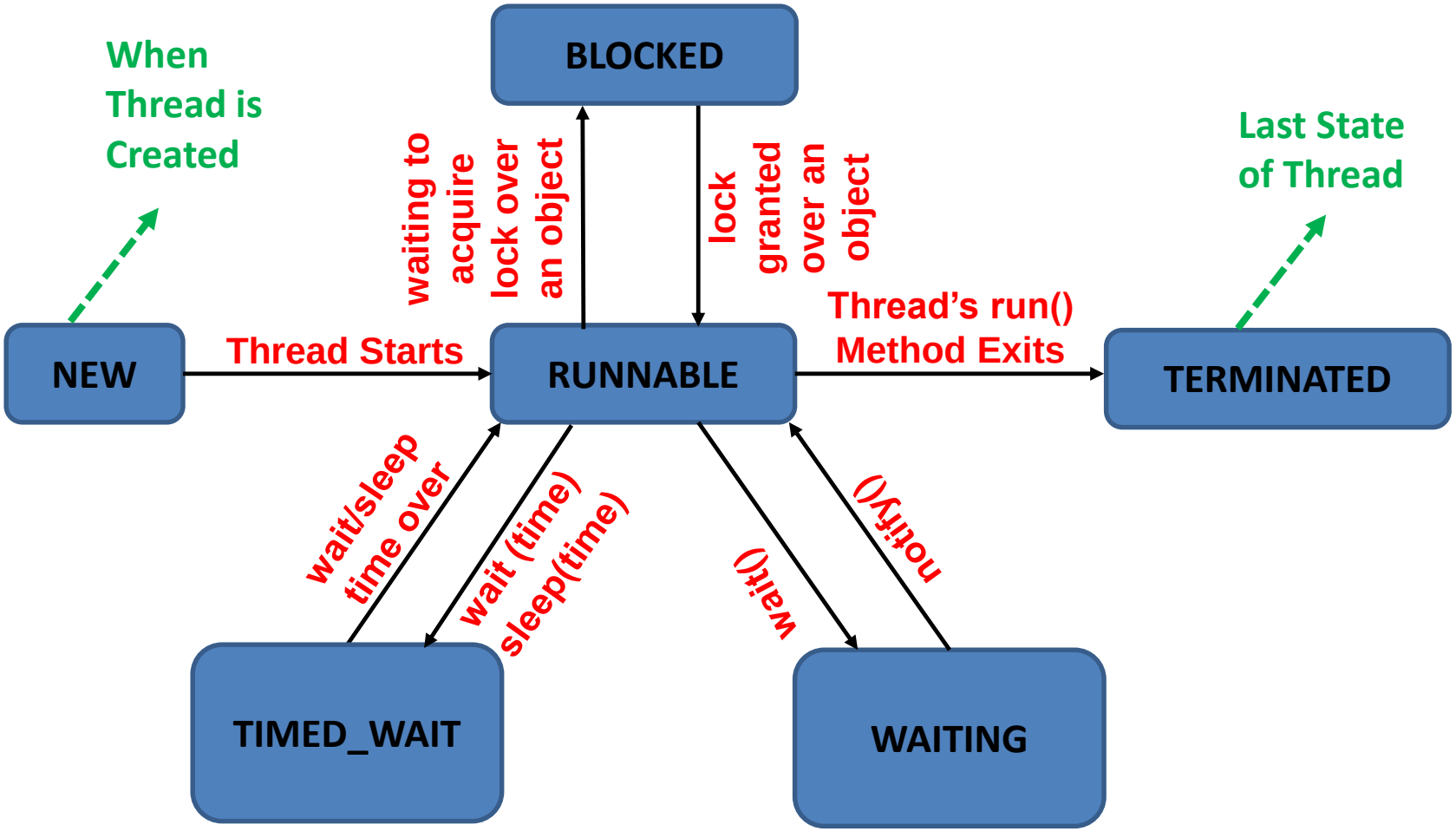
Topics

- Threads in Java

Threads in Java (Basics)

- Java Supports Multithreaded based Multitasking
- Java Virtual Machine (JVM) Executes Each Thread for Short Period of Time (Known as Time Slice)
- Thread Scheduler Activates, Deactivates Threads
- Each Thread is Assigned a Priority Number in Range 1 (MIN_PRIORITY) to 10 (MAX_PRIORITY)
- If No Priority is Assigned to a Particular Thread Then it is Assigned Priority Number as 5 (NORM_PRIORITY)
- Thread Priorities are used by the Thread Scheduler to Decide when Each Thread Should be Allowed to Run. Higher Priority Threads gets more CPU Time than the Lower Priority Threads

Thread States



Thread Scheduling

innovate achieve lead

Thread Scheduler Activates a new Thread if

- ❑ A Thread has Completed its Allotted Time Slice
- ❑ A Thread has Blocked itself
- ❑ A Thread with Higher Priority has become Runnable

Scheduler Determines new Thread to Run

- ❑ Looks only at Runnable Threads
- ❑ Picks the Thread with Maximum Priority

run() Method of a Thread

- Every Thread Performs a Well Defined Task During its Allotted Time Slice
- Task of a Thread is Described by the instructions in its run() Method
- A Thread is Alive as long as its run() method Executes
- When run() method of a Thread Exits, the Thread reaches its Final State (Terminated)
- Life Time of a Thread = Execution Time of Thread's run() Method

The Main Thread

- During the beginning of execution any program, Java Creates a Special Thread known as Main Thread
- It is the Main Thread from which other child threads are created and spawned
- Often it must be the last [But not Necessary and Neither Compulsory] to finish Execution
- Main Thread Performs various Shutdown Actions [That's why it is recommended that Main Thread Should Finish Execution in the Last]
- Reference to the currently running thread can be obtained by using method "static Thread currentThread()" of Thread class.

Thank You