Thread

States of thread

- Sates of thread
  - New (Start)
  - Runnable
  - Running
  - Blocked/suspended
  - Dead
-
-

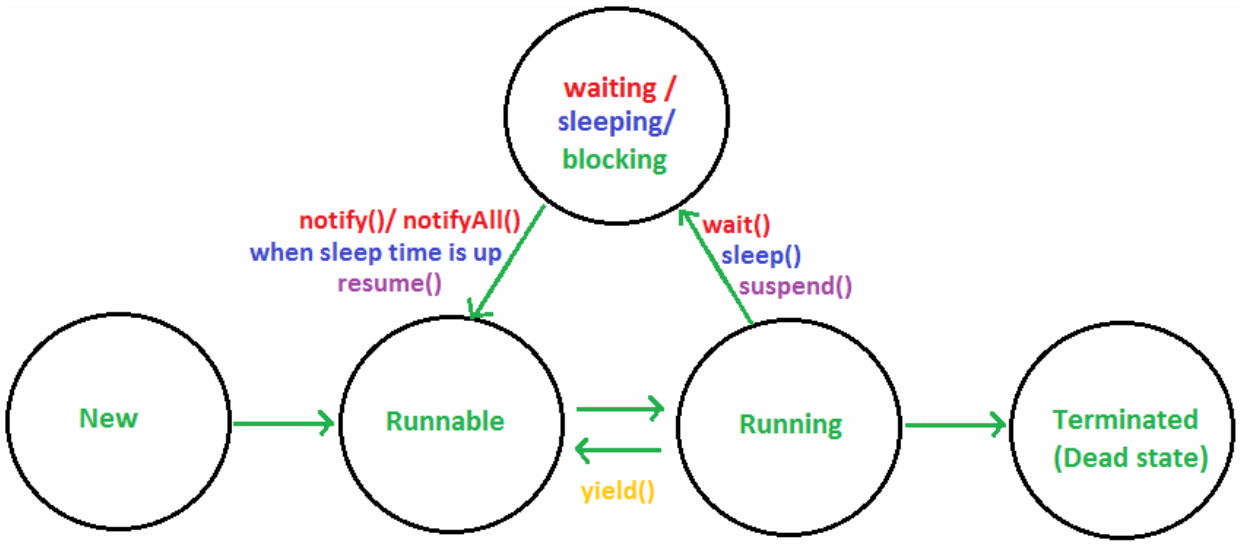

Fig. THREAD STATES

➢ Ways to Create the Thread in Java

- Thread Class
  - Internally uses the Runnable interface method
  - IS-A relationship with Runnable
  - Override run method
- Runnable Interface
  - Preferred way as we are free to extend any other class
  - Override run method

➢ Important methods

- start()
  - Thread class method

        - Used to start the thread
    - run()
        - Runnable interface method
        - Automatically being called by start method internally.
        - Explicit calling is not required.
    - Sleep()
        - Make thread to sleep/hold the execution without releasing the resource.
        - It throws Checked exception
    - isAlive
        - Checks the existence/working of the thread.
    - join()
        - Causes the thread to wait for all the thread to complete the main thread execution.
    -
- Sequence of execution are not same
- Using Runnable interface
    - Override Run method
    - Pass runnable type to Thread class by using construer
        - Thread(Runnable runnable)
    - Call start method on thread object
- Synchronization
    - When two or more thread access the same shared resource, it is required to ensure that only one can access the resource at a time.
    - This can be achieved by using below ways:-
        - Using synchronized block
            - Synchronize the shared resource object
            - Preferred as it only locks the required block of code
        - Using synchronized method
            - Synchronized the method
            - Locks the entire code inside the method
- Inter Thread communication
    - Can only be within synchronized context (either block or method)
    - Wait()
        - It releases the resource and enters into suspended/blocked state.
        -
    - Notify()/notifyAll()
        - Notifies the waiting threads to begin execution.
- Thread priority

- Yield()

- Demon Thread
  - Background Thread
  - Always runs in background.
  - threadObj. setDaemon(true);
  - for Ex: GC, Main Thread
  - priority of 5
- DeadLock
  - When no resource is available.