# Java Thread

Thread → a process in execution.

↓

Allows program to operate more effectively by performing multiple things at a time.

↳ Complicated tasks can be performed in background without interrupting the main program.

↳ Each process can be scheduled and prioritized. – Job scheduling

Ex: Task 1 | P1 | P2 | P3 | P4 |

Scheduling w.r.t time.

- P1 – schedule (100s)
- P2 – Start ( )
- P3 – Schedule (200s)
- P4 – Schedule (300s).

Task 2 | P1 | P2 | P3 | P4 |

"more on these in OS"

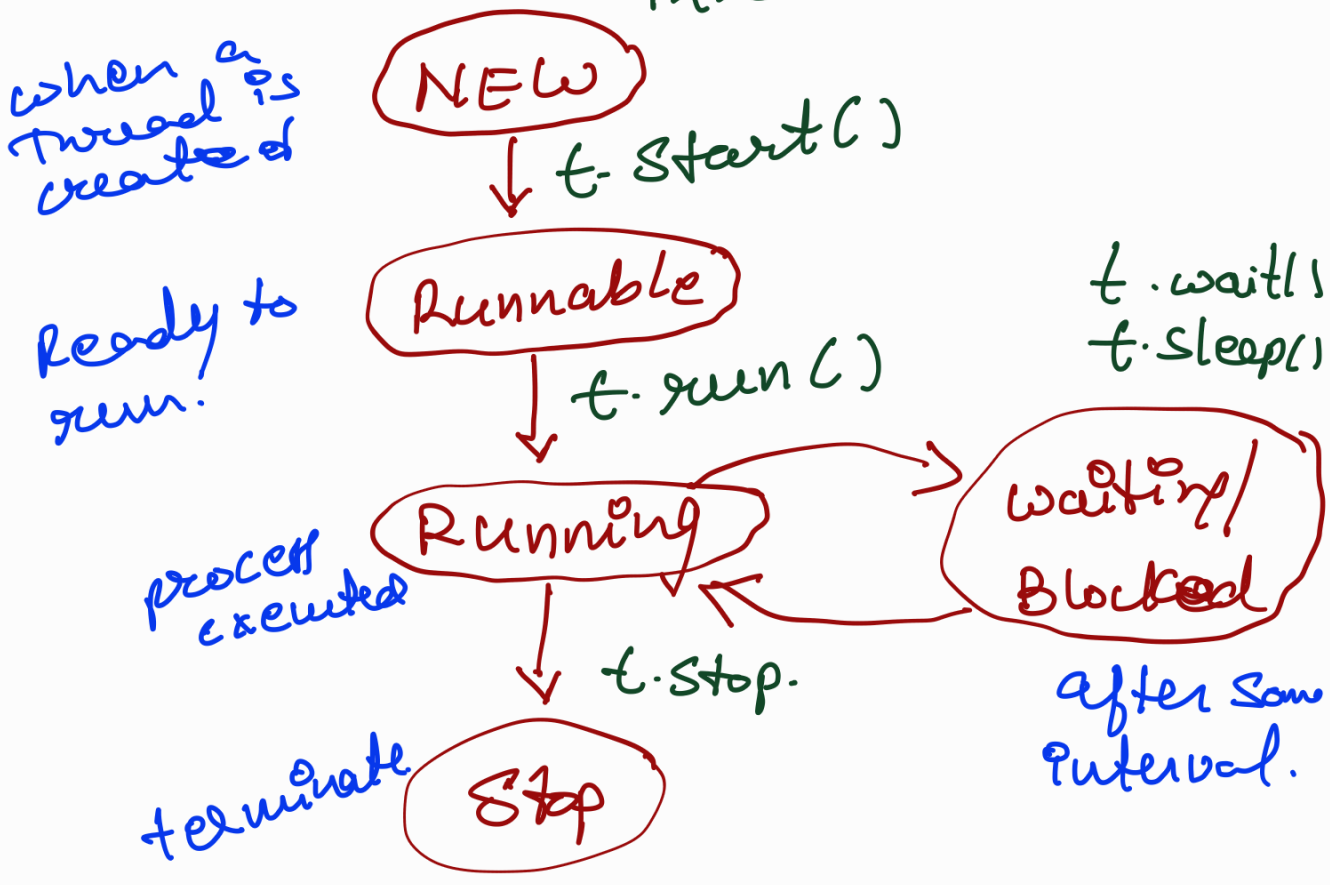
P4 P1 P3 P2

prioritizing.

# Thread Life Cycle

Class : Thread

Thread t = new Thread();




* → run() is overridden in each program.

## Creating Thread

- Inheritance → extends Thread class
- Interface → implements Runnable interface

## Extends Thread

```
public class ST extends Thread {
    public void run() {
        S.o.pln("This is a
                Simple Thread");
    }
p.s.v. main (String[] args) {
    ST myThread = new ST();
    myThread.Start();
    S.o.pln("Thread executed");
    }
}
```

## Implementing runnable interface.

```
public class ST Implements Runnable {
    public void run() {
        S.o.pln("runnable thread");
    }
p.s.v. main(String args[]) {
```

```
ST obj = new ST();
Thread myThread = new Thread(obj);
myThread.start();
S.o.println("Thread executed");
}
}.
```

Note:

(a) If a class extends Thread class, the thread can be run by creating an instance of the class and call its run() method.

(b) If a class implements the Runnable interface, the thread can be run by passing an instance of the class to a Thread object constructor and then calling the threads start() method.