**Thread Scheduling**

* Execution of multiple threads on a single CPU, in some order, is called scheduling.
* In general, the runnable thread with the highest priority is active (running)
* Java is priority-preemptive
  + If a high-priority thread wakes up, and a low-priority thread is running
  + Then the high-priority thread gets to run immediately
* Allows on-demand processing
* Efficient use of CPU

**1 Types of scheduling**

* Waiting and Notifying
  + Waiting [wait()] and notifying [notify(), notifyAll()] provides means of communication between threads that synchronize on the same object.
* wait(): when wait() method is invoked on an object, the thread executing that code gives up its lock on the object immediately and moves the thread to the wait state.
* notify(): This wakes up threads that called wait() on the same object and moves the thread to ready state.
* notifyAll(): This wakes up all the threads that called wait() on the same object.
* Running and Yielding
  + Yield() is used to give the other threads of the same priority a chance to execute i.e. causes current running thread to move to runnable state.
* Sleeping and Waking up
  + nSleep() is used to pause a thread for a specified period of time i.e. moves the current running thread to Sleep state for a specified amount of time, before moving it to runnable state. Thread.sleep(no. of milliseconds);

**2 Thread Priority**

* When a Java thread is created, it inherits its priority from the thread that created it.
* You can modify a thread’s priority at any time after its creation using the setPriority method.
* Thread priorities are integers ranging between MIN\_PRIORITY (1) and MAX\_PRIORITY (10) . The higher the integer, the higher the priority.Normally the thread priority will be 5.