Multitasking:

* Executing several task simultaneously is a concept of multitask there are two types of multitasking
* 1 Process based multitasking
  + Executing several task simultaneously where each task is a separate independent program(process)is called process based multitasking

Ex: Listing music download file, surf internet at same time

* + Best suitable at OS level
* 2 Thread based multitasking
  + Executing several task simultaneously where each task is a separate independent part of the same program is called thread based multitasking and each independent part is called a thread
  + Best suitable at program level
  + Whethr is it process based or thread based the main objective of multitasking is to reduce response time of the system And to improve performance
* Thread scheduler
  + It is the part of jvm
  + It is the responsible to schedule threads that is if multiple threads are waiting to get the chance of the exception then in which order threads will be executed it’s decided by thread scheduler
  + We can’t expect exact algorithm followed by scheduler it depend on jvm hence we cannot expect exact output
* Difference between t.start and t.run
  + Start method is responsible to register our thread with thread scheduler and invoke run method
  + If we run direct run method then thread will not be register as thread and will behave as normal class
* Priority
* Every thread has a default priority and programmer can give customize priority
* Valid range of priority 1-10
* 1 is min and 10 is max
* Thread class defines following constants to represent some standard priority
  + Thread.min\_priority --1
  + Thread.norm\_priority--5
  + Thread.max\_priority—10
* Thread scheduler will use priority while allocating processor which thread has having highest priority will get chance first
* If two threads having same priority then we can’t expect exact output it depands on scheduler
* If we provide illegal range priority ex:11 then RE: illegal argument exception
* We can
* Yield()
* Join ()
* Slepp()

Yield method is caused to pause current executing thread to the give the chance to waiting thread on same priority

New thread born ->Start()->ready/runnable->processor allocation->running->complete process->dead

If yield method call thread will switch to runnable state from running

If a thread wants to wait until complete some other threads then we should go for join method

Exception’s:

IlligalMoniterException: if we use wait,notify,notifyall method in not synchronized method or block

notifyAll: to notify all particular object thread

Green model: without taking os support only use jvm