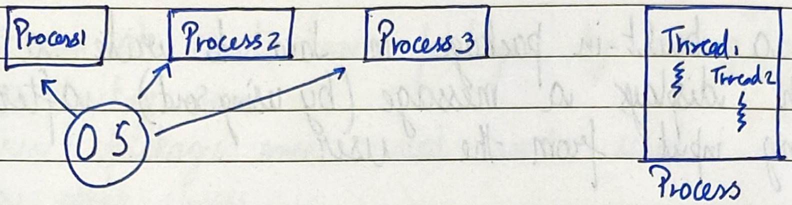
Chapter 13 - Multithreading

Multiprocessing and multithreading both are used to acheive multitasking



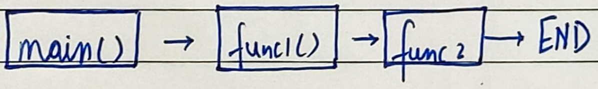
In a nut Shell ...

→ Threads use shared memory area.
→ Threads ⇒ Faster Content switching
→ A Thread is light-weight whereas a process is heavyweight

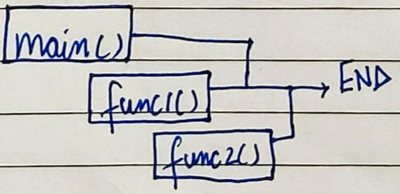
For Example → A word processor can have one thread running in foreground as an editor and another in the background auto saving the document!

Flow of Control in Java

1> Without threading :



2> With threading:

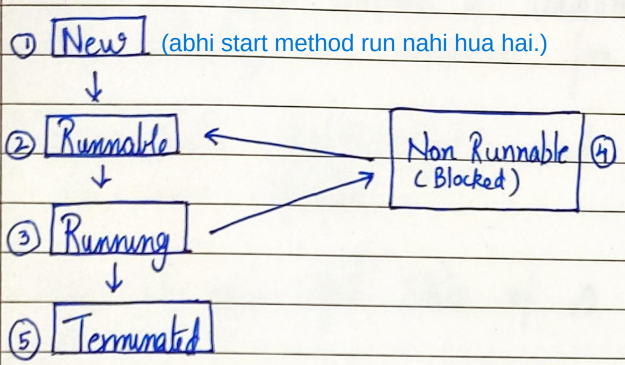


Creating a Thread

There are two ways to create a thread in Java.

1. By extending Thread class and overriding run method.
2. By implementing Runnable interface and overriding run method and make object of class and an object of Thread, putting object of class in Thread object and then calling object of Thread to execute the program using obj.start();

Life cycle of a Thread



(abhi start method run nahi hua hai.)

① New → Instance of thread created which is not yet started by invoking start()

② Runnable → After invocation of start() & before it is selected to be run by the scheduler.

③ Running → After thread scheduler has selected it.

④ Non Runnable → Thread alive, not eligible to run.

⑤ Terminated → run() method has exitted

The Thread class

Below are the commonly used Constructors of Thread class:

① Thread()

② Thread(String name)

③ Thread (Runnable r)

④ Thread(Runnable r, String name)

Methods of Thread class

Thread class offers a lot of methods such as run(), Start(), join(), getPriority(), Set Priority() etc. More can be found on visiting Java docs

.join method

.join() method says that until above thread is not finished don't run the next thread.

Priorities:

Priorities are mainly of three types 1)min Priority - 1
2)norm Priority - 5
3) max Priority - 10
You can set priorities of a thread using obj.setPriority(enter the integer value);

.sleep method

To stop a thread method for some time you can use Thread.sleep(enter to stop time in millis); method . It is used inside the method.

.getState

It gives us the state of a thread - from the life cycle of a thread.