What is Multi tasking programming?

In low label language like C & C++ we can't execute more than one program at a time.

The concept through which we can run or execute several program symonteniously is known as multitasking (In case of windows operating system). But incase of high level language like Java & .Net the above mention concept is known as Multithreading.

What is Multithreading?

To run any long program, deride the program into smaller parts so that the smaller parts are execute symonteniously is the basic concept of Multithreading.

We have an inbuilt java enabled browesers known as HotJava (This browser is only used to execute threading related programming).

How to create Thread programming?

To run any thread programming we have to define run method (run()) in our program. That is to run any java program we have to ~~test~~ weight any thread related program within the scope of - public void run()

```
{

}
```

There are two ways to create a threade programming - (a) extends Thread class and override run method.
(b) implements Runable interface and write down the code within run().

(a) Syntax :-

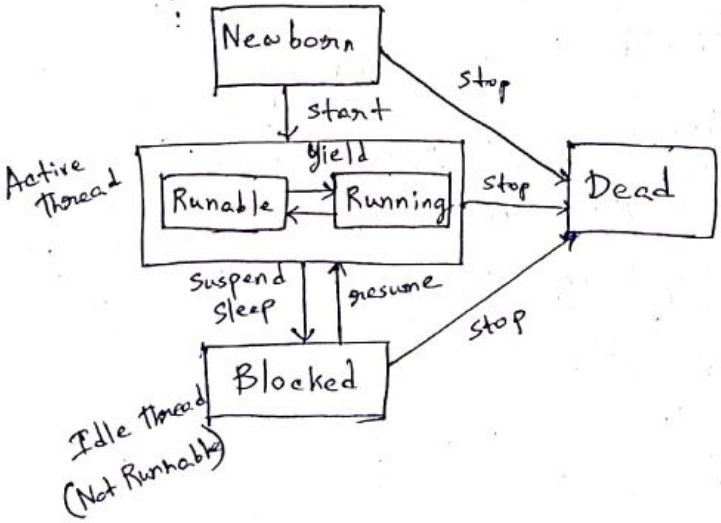
```
class A extends Thread
{

  public void run ()
  {
    = } Source Code
  }
}
```

Life Cycle of Thread -

There are many states to enter into a thread -

1) Newborn State.
2) Runable State.
3) Running state
4) blocked state
5) dead state

(1) execution of the Running
(2) Hold of the Runnable

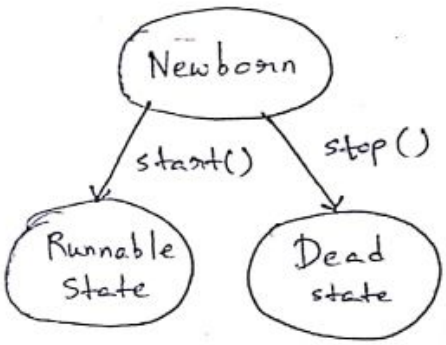


Newborn state- When we are creating an object of thread class the thread is born and is said to be Newborn state. But the thread is not yet to be sheduled for running.

i) start() is used to sheduled the thread program.

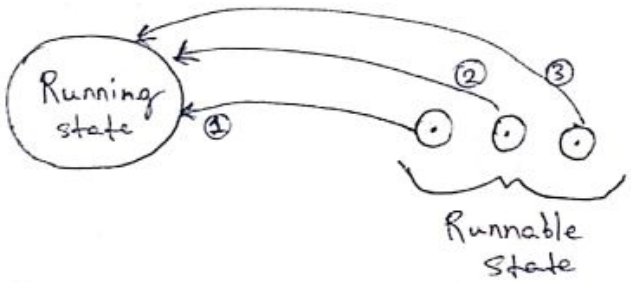
ii) stop() is used to kill the thread program.

```
Thread ob = new Thread();
    ob.start();
    =
    =
    =
ob.stop();
```

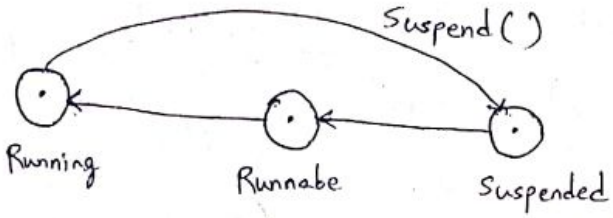


Runnable state - Runnable state means thread is ready for execution and waiting for availability of the processor that is the thread has join the queue of threads that are waiting for the execution,

Note: If the threads have equal priority then there given time slots for execution in round robin fashion that is first come first out method. The thread that relinquish control joins the queue at the end and again waits for its turn. This process asinging time to threads is known as time slicing

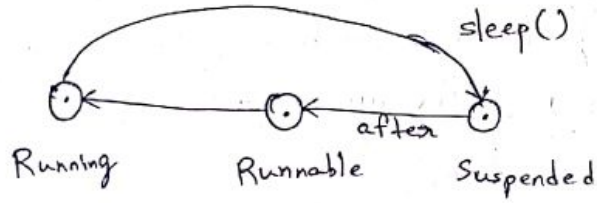


Running state - means that the processes has given its time to the thread for its execution. The thread runs until it relinquishesh control to the other thread.



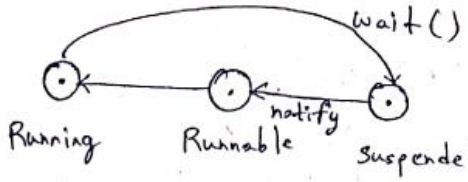
It has been suspended using suspend(). A suspended thread can be revived by using resum method.

Note: This approach is useful when we want to suspend a thread for sometime due to certain reason but do not want to kill it.



It has been made to sleep we can put a thread to sleep for a specific time period using the method sleep (time) where time is in mili seconds.

That means thread is out of the queue during that time. The thread re-enters the runnable state as soon as the time period is ellapsed.



It has been told to wait until some event occures this is done using wait(). The thread can be scheduled to run again using notify method.

Blocked State - A thread is used to be blocked when it is prevented from entering into the ~~running~~ runnable state and as well as into the running state. This happens when the thread is suspended, sleeping or waiting in order to satisfy certain requirements.

Note: A blocked thread is considered as "NOT Runnable" but not dead and therefore totally qualified to run again.

Dead state - Every thread has a life cycle when the run methods execution has been over it is naturally death. How ever we can forcefully kill any thread by using stop function.

Note: When the thread had been dead while using stop(). It can't return back to the runnable state.

Thread Exception :- When we call ~~thread~~ sleep method we have to write down the sleep method inside try & catch block because sleep method will generate an exception known as Illigal Thread State Exception.
class

When ever we attempt to envoke a method that a thread can't handle in the given state.

Example - a sleeping thread can't deal with the resum method because a sleeping thread can't receive any instruction. Like wise suspend method will also do the same.

Thread Priority :- With in a thread class three priority is there. ① Minimum Priority is set to one. 1
② Normal Priority is set to Five. 5
③ Maximum Priority is set to ~~ten~~. 10.

Definition of Thread

Thread is the smallest unit of a process which can run independently main program is also is a thread program.

Process :- Programming in execution state is called process.

Utility of Thread :- We can maximise the utilization of CPU by ~~thre~~ using threading.

Priorities :— i) Every thread has a unique name

ii) We can change the name of the thread.

iii) Every thread have a priority, in Java the priority is lying between 0-10, by default priority is 5, 0 is called the lowest priority or the minimum priority, 5 is called the default priority or normal priority and 10 is called the maximum priority or heighest priority.

We can also change the priority of a thread as per our requirement.

How to create a Thread by using Runnable Interface?

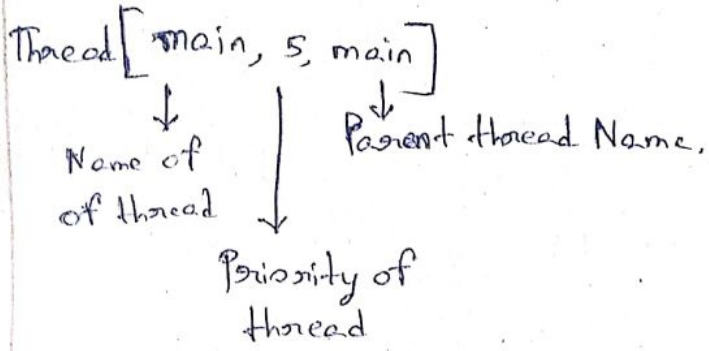
Step 1 :- Create a class which is will implement the runnable interface.

Step 2 :- Create a thread by using thread constructor.

Step 3 :- Call the method start.

Step 4 :- It is automatically called the run funation. Inside the run ( ) we will write all the functionality of ~~the~~ a thread.

System.out.println(t.currentThread());



Thread Synchronization:- We have seen that threads that use their own data and methods provided inside their run methods. What happen when they try to use data and method outside themselves? On such occasions they may compite for the same resources and may lead to serious problems. For example- one thread try to read a record from a file while another is still writing to the same file. Depending on the situation we may get strange results. Java enables us to overcome to this problem using a technique known as synchronization.

In case of Java the keyword synchronized helps to solve such problem by keeping a watch on such locations.

Declaration of any synchronized function :-

~~Synchro:~~

```
synchronised (lock ob)
{
  =  ⟶ code here
}
```