Multitasking :-

1. Process-based Multitasking

In this type of multitasking, two or more processes can be run concurrently. The smallest unit of execution is a process.

- Inter-process communication is costly and inefficient.
- Context-switching time is more in the case of processes.
- It is unable to gain access over CPU idle time.
- Every program has its own address space.
- It requires more overhead than thread-based multitasking.

Example:
A browser and a video player are two processes that can be running at the same time on a modern computer.

2. Thread-based Multitasking

In this type of multitasking, two or more threads can be run concurrently. The smallest unit of execution is a thread.

- Inter-thread communication is inexpensive and efficient.
- Context-switching time is less in the case of threads.
- It can gain access over CPU idle time.
- Threads share the same address space.
- It requires less overhead than process-based multitasking.

Example:
A word processor can simultaneously have a thread for spelling and grammar checks and another thread for responding to keystrokes from the user.

Multithreading :-

What is multithreading?

Multithreading is a process to execute multiple threads at the same without dependency of other threads.

What is threads?

Thread is a pre-defined class which is available in java.lang package. Unit of CPU and it is well known for independent execution.

We can create threads by :-

- Extending thread classes.
- Implementing Runnable interface.

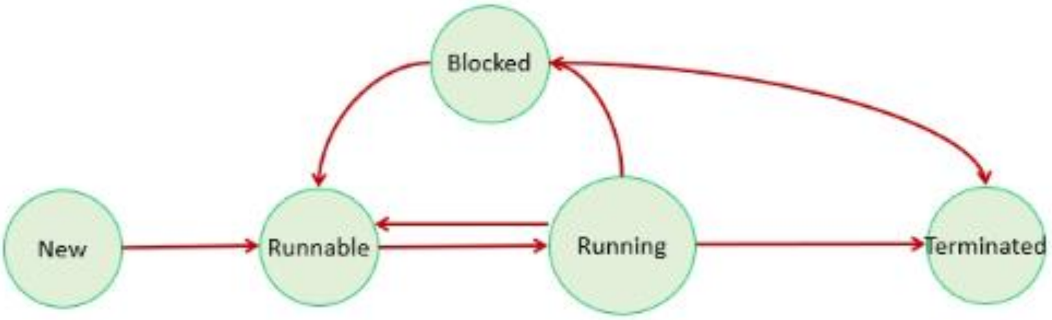
What is thread scheduler?

It is the part of JVM which executes multiple threads on a single processor randomly.

What is thread life cycle?

As we know a thread is well known for independent execution. During the life cycle a thread can move from different state.

- New State(Born)
- Runnable State(Ready)
- Running State(Execution)
- Waiting State(Blocked)
- Dead State(Exit)



Methods :-

What is Sleep() method?

It is a static method of thread class which throws checked exceptions (interruptedException). The main purpose of sleep method is to put a thread into temporary waiting state.

```
Thread.sleep(1000);

Thread t = new Thread();

t.sleep();
```

What is join() method?

If we want to execute complete any particular thread among all the threads available in the thread pool. It also throws interruptedException.

What is suspend() and resume() method?

The main purpose of suspend method is to put the thread from running state to waiting state.

The main purpose of resume method is to resume a suspended thread from waiting state to runnable state.

What is yield() method?

It is a method of thread class that allow us to run another thread which has same priority by pause it's current thread.

What is stop() method?

It is method of thread class which is used to terminate a thread permanently.

What is isAlive() method?

It is a pre-defined method of thread class through which we can verify weather a thread is alive or not.

- If thread is alive then it will return true otherwise false.
- If we use isAlive before the start method then it will print false but after the start method it will print true.

What is interrupt() method?

It is a pre-defined method of thread class, that is used to interrupt a thread.

- If any thread is sleeping or blocked state then we can easily interrupt the execution of thread by throwing InterruptedException.
- If thread not in sleeping or waiting state then thread executes normally.

What is thread priority?

In java it is possible to assign the priority of thread. To set this priority java thread class has provided two pre-defined methods.

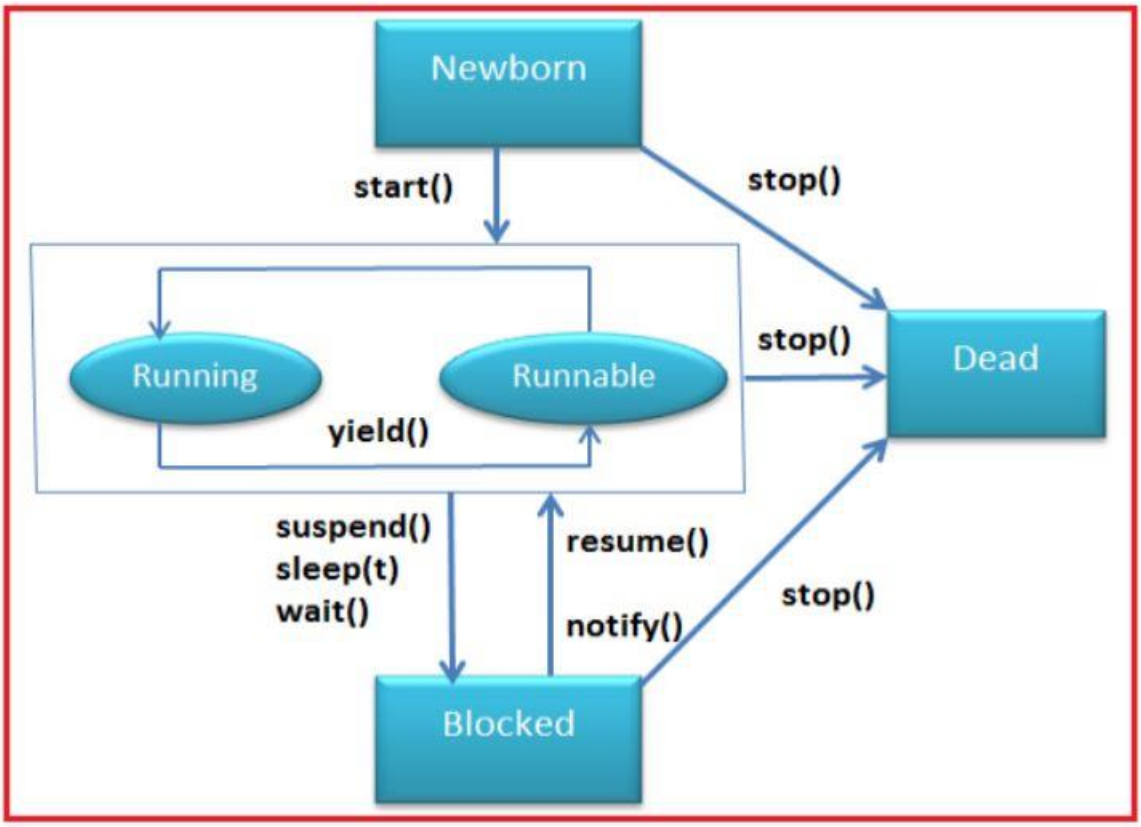
- setPriority();
- getPriority();

The thread class has also provided three pre-defined final static variables and it's value lies between 1 to 10.

Thread.MIN_PRIORITY =>1

Thread.MIN_PRIORITY =>5 //default

Thread.MAX_PRIORITY =>10


Newborn
start()
stop()
Running
Runnable
yield()
stop()
Dead
suspend()
sleep(t)
wait()
resume()
notify()
stop()
Blocked