SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

19CS204 OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

Topic - Multithreading

Multithreading

- Java provides built-in support for multithreaded programming.
- A multithreaded program contains two or more parts that can run concurrently.
- Each part of such a program is called a thread, and each thread defines a separate path of execution.
- Thus, multithreading is a specialized form of multitasking.
- In a thread-based multitasking environment, the thread is the smallest unit of dispatchable code.
- This means that a single program can perform two or more tasks simultaneously

Multithreading

Multitasking

- Executing multiple tasks or programs simultaneously. Each program in execution is called a process.
- Process based multitasking
- Thread based multitasking

Process based multitasking

- process-based multitasking is the feature that allows your computer to run two or more programs concurrently. A process is heavyweight.

Thread based multitasking

- the thread is the smallest unit of dispatchable code. This means that a single program can perform two or more tasks simultaneously.
- A thread is lightweight.

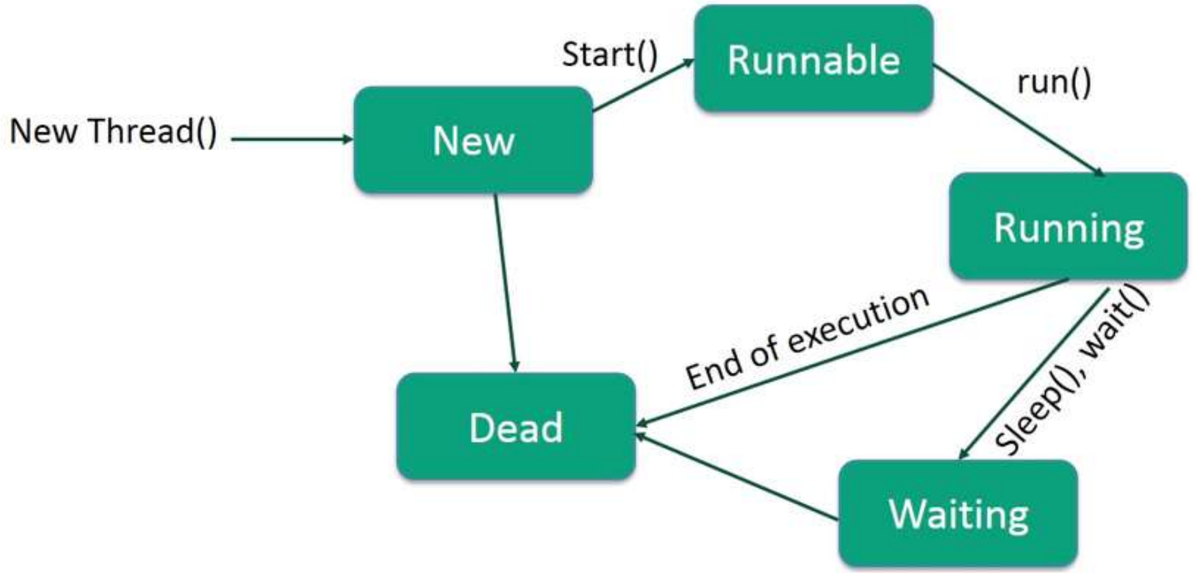
What is thread?

- Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU.
- Each part of such program is called a thread. So, threads are light-weight processes within a process.
- A thread is a lightweight subprocess, the smallest unit of processing.
- It is a separate path of execution.
- Threads are independent. If there occurs exception in one thread, it doesn't affect other threads.
- It uses a shared memory area.

Thread Life-Cycle

- A thread can be in one of the five state.

- The life cycle of the thread in java is controlled by JVM. The java thread states are as follows:
  1. New
  2. Runnable
  3. Running
  4. Non-Runnable (Blocked) or Waiting
  5. Terminated or Dead



Thread Life-Cycle

1. **New** The thread is in new state if you create an instance of Thread class but before the invocation of start() method.

2. **Runnable** The thread is in runnable state after invocation of start() method, but the thread scheduler has not selected it to be the running thread.

3. **Running** The thread is in running state if the thread scheduler has selected it.

4. **Non-Runnable or Wait** This is the state when the thread is still alive, but is currently not eligible to run. This is the state when a thread has to wait. As there multiple threads are running in the application, there is a need for synchronization between threads.

5. **Terminated or Dead** A thread is in terminated or dead state when its run() method exits. Which means it finished processing or execution.

Thread Methods

The Thread class defines several methods that help manage threads**.**

Method Name	Meaning
Start	Starts the execution of the thread and JVM calls the run() method on the thread
sleep	Suspend a thread for a period of time.
Run	Entry point for the thread.
join	Wait for a thread to terminate.
isAlive	Determine if a thread is still running.
getPriority	Obtain a thread’s priority
getName	Obtain a thread’s name.
Yield	It causes current thread on halt and other threads to execute.
getId	It returns the id of the thread.

Other than this currentThread, setPriority, setName, suspend, resume, stop, destroy, interrupt, getState and many more method are avialble in thread class.

THANK YOU