What is Thread?

- A thread is the smallest unit of process
- It can be executed independently
- It is lightweight sub-process
- Each thread has its own path of execution
- It means it can perform tasks concurrently with other threads

Difference Between Process and Thread

- A process is an independent program which runs on its own memory space. Example: Browser – it runs as a separate process
- Is a part of process. A single process can have multiple threads running within it, sharing the same memory space.
- Memory Usage- threads share memory with other threads in the same process, while processes have separate memory spaces.
- Threads can communicate with each other as they share the same memory space.

Life Cycle of a thread:

New : The thread is created but not yet started.

Runnable: The thread is ready to run , waiting for CPU to execute it.

Blocked/Waiting: The thread is temporarily inactive, waiting for a resource or another thread to complete.

Timed waiting: The thread is waiting for a specific amount of time.(e.g, using sleep())

Terminated: The thread has completed its execution and is no longer running.

- Creating a simple thread

```java
package thread;

public class MyThread extends Thread{

    public void run(){
        System.out.println("Thread is running....");
    }

    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        t1.start();
    }
}
```

- Way to create thread
  1. By extending the thread class
  2. By implementing the Runnable interface.

Thread Methods:

- Start(): It begins the execution of the thread
- Run(): contains the code that the thread executes
- Sleep(milliseconds): pauses the thread for a specific amount of time
- Join() : waits for a thread to complete its execution
- isAlive() : checks if the thread is still running.
- setName() & getName() : Used to set and get the name of thread.

1. New State:

```
package thread;

public class MyThread extends Thread{

    public void run(){
        System.out.println("Thread is running....");
    }

    public static void main(String[] args) {
        MyThread t1 = new MyThread(); // Thread is in
new state
        System.out.println("Thread State: "+
t1.getState());
    }
}
```

O/P:

Thread State: NEW

2. Runnable State

A thread enters the Runnable state when the start() method is called.

```
package thread;

public class MyThread extends Thread{

    public void run(){
        System.out.println("Thread is running....");
    }

    public static void main(String[] args) {
        MyThread t1 = new MyThread(); // Thread is in new
state
        t1.start();     // Thread moves to runnable state
        System.out.println("Thread State: "+ t1.getState());
    }
}
```

O/P:

Thread State: RUNNABLE
Thread is running....

3.Blocked/ Waiting State
Loose Coupling
Why Runnable Interface?
Why Thread Class?

Thread (Java SE 21 & JDK 21)