**Thread Life Cycle in Java (**

**1. New (Born Stage)**

* Imagine a newborn baby. It exists but cannot do anything yet.
* A thread is in the **New** state when it is created but hasn’t started running.
* **How?**

Thread t = new Thread(); // Thread is born

* The thread stays in this state until you call the start() method.

**2. Runnable (Ready to Work)**

* Now the thread is ready to work but is waiting for its turn, like standing in a queue.
* A thread enters this state after calling start(). It’s ready but not yet running because the CPU decides when it will execute.
* **How?**

t.start(); // Thread is ready

**3. Running (Working)**

* The thread is finally working! It gets the CPU and starts executing the task defined in its run() method.
* A thread moves to the **Running** state from **Runnable** when the CPU gives it time to execute.
* **How?**

public void run() {

System.out.println("Thread is working!");

}

**4. Waiting/Blocked (Taking a Break)**

* Sometimes, the thread must pause or wait for something (e.g., a resource or a signal).
* It’s like someone taking a break or waiting for their turn again.
* A thread goes into this state when:
  + It's sleeping (Thread.sleep()).
  + Waiting for another thread to complete (join()).
* **How?**

Thread.sleep(1000); // Thread pauses for 1 second

**5. Terminated (Finished)**

* When the thread finishes its task, it’s done and can’t work anymore.
* This is the **end** of the thread’s life.
* **How?**

System.out.println("Thread finished!");

**Thread Life Cycle Example (Step by Step)**

Here’s an easy example showing the thread life cycle:

class MyThread extends Thread {

public void run() {

System.out.println("Thread is running...");

}

}

public class ThreadLifeCycle {

public static void main(String[] args) {

Thread t = new MyThread(); // New State (Thread is born)

t.start(); // Runnable State (Thread is ready)

}

}

**Key Points**

1. **New**: Thread is created but not running.
2. **Runnable**: Thread is ready but waiting for the CPU to execute it.
3. **Running**: Thread is actively working.
4. **Waiting/Blocked**: Thread is paused or waiting for something.
5. **Terminated**: Thread has finished its work.

Think of a thread as a worker who has:

* A **start** (born and ready to work),
* A time to **work** (running),
* A **break** (waiting),
* And a time to **finish** (terminated).