**Day 8: 27-10-2025:**

**Inner thread communication**

To make thread communicate with each other java provided 3 main method

1. wait() : this method is use to make the thread to wait or suspend.
2. notify() : this method is use to resume or callback waited thread. it resume only one thread.
3. notifyAll() : notifyAlll resume more than one thread.

more than one thread part of same memory.

Method must be synchronized.

All these method part of object class.

**Dead lock**

A dead lock happen while writing code using multi threading. When two or more than one thread are waiting for each other forever and none of them can continue. Because all those thread are executing inside a synchronized method and calling other thread synchronized code.

3rd way to create the thread using Callable interface.

Callable interface is a part of concurrent package and it is a sub package of util package.

Runnable Vs Callable

Runnable is an interface Callable is an interface

Part of lang package part concurrent package

run() method call() method

No return type Custom return type with generic and throw exception.

Callback we will execute ExecutorService API.

With help of executor we execute or submit the task.

Using Thread class or implements Runnable we execute or start every thread manually. One thread one task using ExecutorService we can create thread pool and we can do re-usability of thread.

**Concurrency :**

Using the concurrency we can execute multiple task simultaneously or overlapping in time and efficient way to manage the resources using thread as well as thread pool.

**Daemon thread:** Daemon thread is a type thread with low priority mainly help us to do background task. If we make user defined thread as a daemon thread JVM doesn’t wait to finish the task of the Daemon thread.

**Concurrency Collection API**

Collection -🡪 interface

**Set**

HashSet -🡪 Unorder not a thread safe

LinkedHashSet🡪 order

TreeSet 🡪 sorted

All three are not a thread

CopyOnWriteArraySet -🡪 thread safe

ConcurrentSkipListSet 🡪 thread safe, sorted for primitive values.

**List**

ArrayList

LinkedList

Vector (thread safe but legacy )

All these 2 are not thread safe

CopyOnWriteArrayList -🡪 Thread safe List API.

Map :

HashMap unorder

LinkedHashMap order

TreeMap ascending order as key

Hashtable legacy by default synchronized

ConcurrentHashMap thread safe unorder

ConcurrentSkipListMap thread safe ascending order as key

Queue

ArrayBlockingQueue array implementation

LinkedBlockingQueue linked list implementation

PriorityBlockingQueue priority queue implementation

All are thread safe