**MULTITHREADING CONCEPTS**

**Thread Class Methods**

| **Method** | **Description** |
| --- | --- |
| start() | Starts a new thread by calling the run() method. |
| run() | Contains the code that is executed by the thread. |
| sleep(ms) | Pauses the current thread for a specified time (in milliseconds). |
| yield() | Suggests that the current thread is willing to yield its current use of the CPU. |
| join() | Waits for another thread to finish execution. |
| interrupt() | Interrupts a thread that is sleeping or waiting. |
| isAlive() | Checks if a thread is still running. |
| setPriority(int) | Sets the priority of a thread (1 to 10). |
| getPriority() | Gets the priority of a thread. |
| setName(String) | Sets the name of a thread. |
| getName() | Gets the name of a thread. |

**Locking and Synchronization**

| **Class/Method** | **Description** |
| --- | --- |
| synchronized | Keyword to lock methods or blocks for thread safety. |
| ReentrantLock | A flexible lock with advanced features like fairness and try-lock. |
| lock() / unlock() | Used with ReentrantLock to acquire and release locks. |
| tryLock() | Attempts to acquire the lock without blocking. |
| lockInterruptibly() | Acquires the lock unless the thread is interrupted. |
| newCondition() | Creates condition variables for advanced thread coordination. |

**🧵 Thread Coordination**

| **Method/Class** | **Description** |
| --- | --- |
| wait() | Causes the current thread to wait until another thread calls notify() or notifyAll(). |
| notify() | Wakes up a single thread waiting on the object's monitor. |
| notifyAll() | Wakes up all threads waiting on the object's monitor. |
| CountDownLatch | Allows one or more threads to wait until a set of operations are completed. |
| CyclicBarrier | Allows a set of threads to wait for each other to reach a common barrier point. |
| Semaphore | Controls access to a resource through permits. |
| Exchanger | Allows two threads to exchange data. |

**Advanced Concurrency Utilities**

| **Class** | **Description** |
| --- | --- |
| ExecutorService | Manages a pool of threads for executing tasks asynchronously. |
| Future | Represents the result of an asynchronous computation. |
| Callable | Similar to Runnable, but returns a result and can throw exceptions. |
| ForkJoinPool | A special thread pool for parallelism using divide-and-conquer (used with ForkJoinTask). |
| ForkJoinTask | Base class for tasks that run in a ForkJoinPool. |
| RecursiveTask / RecursiveAction | Subclasses of ForkJoinTask for tasks that return results or not. |