1. Create and Run a Thread using Runnable Interface and Thread class.

Ans == <https://github.com/pranav0297/Thread/blob/main/create_thread_1.java>

1. Use sleep and join methods with thread.

Ans == <https://github.com/pranav0297/Thread/blob/main/sleepJoin_2.java>

1. Use a singleThreadExecutor to submit multiple threads.

Ans == <https://github.com/pranav0297/Thread/blob/main/ExecutorService_3.java>

1. Try shutdown() and shutdownNow() and observe the difference.

Ans == <https://github.com/pranav0297/Thread/blob/main/shutdown.java>

1. Use isShutDown() and isTerminated() with ExecutorService.

Ans == <https://github.com/pranav0297/Thread/blob/main/isshutdown_5.java>

1. Return a Future from ExecutorService by using callable and use get(), isDone(),

Ans == <https://github.com/pranav0297/Thread/blob/main/Future_6.java>

1. Submit List of tasks to ExecutorService and wait for the completion of all the tasks.

Ans == <https://github.com/pranav0297/Thread/blob/main/java_7.java>

1. Schedule task using schedule(), scheduleAtFixedRate() and scheduleAtFixedDelay()

Ans == <https://github.com/pranav0297/Thread/blob/main/Schedule_8_1.java>

<https://github.com/pranav0297/Thread/blob/main/Schedule_8_2.java>

<https://github.com/pranav0297/Thread/blob/main/Schedule_8_3.java>

1. Increase concurrency with Thread pools using newCachedThreadPool() and newFixedThreadPool().

Ans == <https://github.com/pranav0297/Thread/blob/main/java_9.java>

1. Use Synchronize method to enable synchronization between multiple threads trying to access method at same time.

Ans == <https://github.com/pranav0297/Thread/blob/main/SynchronizedMethod_10.java>

1. Use Synchronize block to enable synchronization between multiple threads trying to access method at same time.

Ans == <https://github.com/pranav0297/Thread/blob/main/SynchronizedBlock_11.java>

1. Use Atomic Classes instead of Synchronize method and blocks.

Ans == <https://github.com/pranav0297/Thread/blob/main/Atomic_12.java>

1. Coordinate 2 threads using wait() and notify().

Ans == <https://github.com/pranav0297/Thread/blob/main/waitnotify_13.java>

1. Coordinate mulitple threads using wait() and notifyAll()

Ans == <https://github.com/pranav0297/Thread/blob/main/waitnotifymultiple_14.java>

1. Use Reentract lock for coordinating 2 threads with signal(), signalAll() and wait().

Ans == <https://github.com/pranav0297/Thread/blob/main/java_15.java>

1. Create a deadlock and Resolve it using tryLock().

Ans == <https://github.com/pranav0297/Thread/blob/main/ReentrantLock_16.java>