Java Concurrency Exercises

1. Write two java thread programs with **Callable** implementation with ***Future*** return types to read and write from common array list of Persons objects.(id,date_of_Birth,List_of_Jobs) .Synchronize the access to *common list* with Lock objects as read and write access.
2. Implement the above with CountDownLatch,Semaphore and mutexes.
3. Implement two Callable Threads to write interleaved messages on sonsole.Provide proper synchronization with *Semaphore* objects.

4. Define two threads with Callable one for writing even numbers and one for writing odd numbers at interval of 2 seconds on console. Using these two threads print numbers till 100 in a sequence. Use *Semaphore* to make it sequential.

5. Define a List containing employee information (name, age, salary, address) and share this across two threads with **Callable** and **Future** as one to read with iterator and other thread to add/remove records by intervals. Provide proper locking and monitor the iterator in first thread for any exceptions.

6. In the above define PooledExecutor to run those two threads in parallel with controlled access.

7. Define a User class with operations for updateUser and readUser methods. Inside updateUser write to user list and in second to wtite to list. Provide console print messages with intervals of 2 sec to display progress. Implement locking with Semaphore and ReadWriteLocks.

8. Implement the above with **CompletableFuture** usage**.**

9. Define two threads with **Callable** and **Future** one to add records in shared list and other to remove duplicates from the shared list. Use proper control with customized **Lock** implementations.

****