CMP3001 Operating Systems
Project (Synchronization)

Due: Sunday, January 16th (23:59)

The Reader Writer Problem

In this project you will need to provide a solution to readers-writers problem in which several processes (readers and writers) are trying to access shared variables. Obviously, if two readers access the shared data simultaneously, no adverse effects will result, hence, they are allowed to access. However, if a writer and some other process (either a reader or a writer) access the data simultaneously, chaos may ensue. To ensure that these difficulties do not arise, we require that the writers have exclusive access to the shared data while writing to the data.

The solution must guarantee that:

- If a writer has begun writing process, then
  - No additional writer can perform write function
  - No reader is allowed to read
- If 1 or more readers are reading, then
  - Other readers may read as well
  - No writer may perform write function until all readers have finished reading

You are given Test class written in Java that use ReadWriteLock class and threads for the problem. You are expected to use Semaphore provided in the code.

Two operations on the semaphore is allowed; **acquire()** and **release()** (they correspond wait and signal functions)

To do: Implement methods of **ReadWriteLock** class given.

```
class ReadWriteLock{
	private Semaphore S=new Semaphore(1);
	public void readLock() {
	}
	public void writeLock() {
	}
	public void readUnLock() {
	}
	public void writeUnLock() {
	}

}
```

Submission:

You must still submit your source code and **a word document** explaining details of your solution over **itslearning** before the deadline. No submission will be accepted after deadline.

Grading:

Your grade will be based on manual inspection of the code.