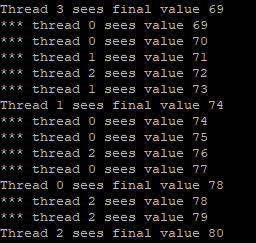
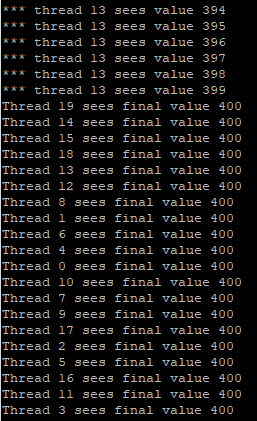
## Part 2: Simple Multithreaded Programming with Proper Synchronization

4 threads without synchronization.



The above output is the result of having 4 threads writing to a shared resource simultaneously without proper synchronization causing a “race condition”. As a result, the final value is non-deterministic and therefore every thread sees a different final value.

20 threads with synchronization.



This output is achieved by using a mutex object to synchronize the critical part of the code in which the shared resource is modified by all threads. To ensure that all threads read the same final value, a semaphore is used to make threads wait until all threads are done modifying the shared resource.