CSE 551
Design Pattern for Concurrent Programming

1. All shared data has a lock

2. Lexically pair lock with unlock – no complex control flow (e.g., do not throw a lock across a fork)

3. Always lock before use – no performance optimization

4. Reestablish invariant before release

5. Always retest condition after wakeup

6. If you must acquire multiple locks, use ordering

7. Always assume you are multithreaded

8. Use objects: do not modify shared variables or static storage in a procedure

9. Each call should be synchronous (asynchrony via thread fork/join)

10. Don't use naked notify (unless you are an interrupt handler, and even then, use semaphores instead)