CPSC/ECE 3220 – Summer 2018 – Exam 2 Name: ______________________

No Electronics.

Write one of the words or terms from the following list into the blank appearing to the left of the appropriate definition. Note that there are more words and terms than definitions. (2 pts. each)

affinity scheduling	busy waiting	compare-and-swap	concurrency	CPU burst
critical section	deadline	false sharing	fine-grain locking	gang scheduling
lock-free data structure	mutual exclusion	oblivious scheduling	preemption	semaphore
synchronization barrier	test-and-set	time quantum	workload	work-conserving

1. ________________________ A sequence of code that operates on shared state.

2. ________________________ Multiple activities that can happen at the same time.

3. ________________________ An efficient way to check that N threads have finished their work.

4. ________________________ A type of synchronization variable with only two atomic operations, P() and V().

5. ________________________ When a scheduler takes the processor away from one task and gives it to another.

6. ________________________ When a thread uses a lock to prevent concurrent access to a shared data structure.

7. ________________________ A thread spins in a loop waiting for a concurrent event to occur, consuming CPU cycles while it is waiting.

8. ________________________ A scheduling policy where the OS assigns threads to processors without knowledge of the intent of the parallel application.

9. ________________________ A way to increase concurrency by partitioning an object's state into different subsets each protected by a different lock.

10. ________________________ A scheduling policy for multiprocessors that performs all the runnable tasks for a particular process at the same time.

11. ________________________ An atomic read-modify-write instruction that reads a value from memory to a register and writes the value 1 to that memory location.

12. ________________________ An atomic read-modify-write instruction that first tests the value of a memory location, and if the value has not been changed, sets it to a new value.

13. ________________________ A scheduling policy where tasks are preferentially scheduled onto the same processor that they had previously been assigned, to improve cache reuse.

14. ________________________ Extra inter-processor communication required because a single cache entry contains portions of two different data structures with different sharing patterns.

15. Here is proposed code for software synchronization that uses a global “available” flag. (5 pts.)

```
                        boolean available = true; // initialization

T1S1   if( available ){                          T2S1   if( available ){
T1S2       available = false;                    T2S2       available = false;
       }else{                                           }else{
T1S3       while( !available ) /*empty*/;        T2S3       while( !available ) /*empty*/;
T1S4       available = false;                    T2S4       available = false;
       }                                                }
T1S5   // critical section 1                     T2S5   // critical section 2
T1S6   available = true;                         T2S6   available = true;
```

Show a scenario in which this approach fails to provide mutual exclusion. Give the exact sequence using the T<i>S<j> thread and statement identifiers so that it is clear the order in which the statements are executed.

16. This is a buggy version of the BBQ::insert() method from the blocking bounded queue code in the textbook:

```
void BBQ::insert(int item) {

        while ((nextEmpty - front) == MAX) {
                ItemRemoved.signal(&lock);
        }

        lock.acquire();

        items[nextEmpty % MAX] = item;
        nextEmpty++;

        ItemAdded.signal();

        lock.release();
}
```

Identify and correct the two bugs. (2 pts.)

17. With regard to signals and waits, what do we mean when we say that a condition variable is memoryless? (2 pts.)

18. What is the difference between a CV::signal() operation and a CV::broadcast() operation? (2 pts.)

19. What is the only assumption you should make on return from a CV::wait() operation? (2 pts.)

20. Beside spurious wakeups, why must the CV::wait() operation be called from within a loop? (2 pts.)

21. Under what two conditions does the textbook say it is appropriate to replace a mutual exclusion lock with a RWLock? (2 pts.)

<u>Compare and Swap / Mellor-Crummey Scott lock / Read-Copy-Update.</u> Circle **only one** of CAS, MCS, or RCS. (3 pts. each)

22. CAS / MCS / RCU Has a “grace period”.

23. CAS / MCS / RCU Is an efficient form of readers/writer locking.

24. CAS / MCS / RCU An efficient spinlock implementation where each waiting thread spins on a separate memory location.

25. What is the key idea in hardware transactional memory? (2 pts.)

26. Give the four necessary conditions for deadlock. (8 pts.)

27. Identify at least one deadlock recovery approach. (2 pts.)

True/False. Circle **only one** of T or F. (2 pts. each)

Use this table for questions 19-22 for deadlock avoidance questions based on the Banker's algorithm. Treat each question independently, starting each time from the values shown in the table.

process	max need	allocated	remaining need
A	10	4	6
B	9	2	7
C	8	3	5

Unused units = 6

28. T / F Can safely grant request of process A for 3 units.

29. T / F Can safely grant request of process B for 3 units.

30. T / F Can safely grant request of process C for 3 units.

FIFO/RR/MFQ/SJF-preemptive. Circle **one or more** of F, R, M, S, as applies. (2 pts. each)

31. F / R / M / S Is preemptive.
32. F / R / M / S Uses time quantums.
33. F / R / M / S Has minimum average response time.

True/False. Circle **only one** of T or F. (2 pts. each)

34. T / F Round-robin scheduling allows starvation of long-running tasks.
35. T / F A hardware timer allows round-robin scheduling to be implemented.
36. T / F Priority boosting when signaling a waiting task on I/O completion can help a system increase utilization of I/O devices.

37. Given the following list of tasks, arrival times, and service times, calculate the completion (i.e., departure) time and response (i.e., turnaround) time of each task for FIFO scheduling and again for SJF-preemptive scheduling. (8 pts.)

			------------ FIFO ------------		------ SJF-preemptive ------	
Task	Arrival	Service	Completion	Response	Completion	Response
	Time	Time	Time	Time	Time	Time
A	0	10	_________	_________	_________	_________
B	2	4	_________	_________	_________	_________
C	4	1	_________	_________	_________	_________

38. When can RR result in pessimal (i.e., worst case scheduling)? (2 pts.)

39. Briefly explain priority donation in the context of priority inversion. (4 pts.)

40. If additional resources are not available, what are two approaches identified in the textbook to manage overload? (2 pts.)