1. What does a threading.Lock do?

a) Pass messages between threads **b) Allow only one thread at a time to access a resource**

c) Wait until a thread is finished d) SHIFT TO ALL CAPS

2. What does the Thread.join() method do?

**a) Waits for the thread to finish**  b) Restricts access to a resource

c) Adds the thread to a pool d) Merges two threads into one

3) Race conditions are …

a) Testing which thread completes first **b) Two threads incorrectly accessing a shared resource**

c) The weather on race day d) Something you should add to your code

4) It sets the lock state to locked. If called on a locked object, it blocks until the resource is free.

a) lock() b)release() **c) acquire()** d)join()

5) You have thread T1, T2, and T3. How will you ensure that thread T2 is run after T1 and thread T3 after T2?

a) Sleep method **b) Join Method** c)Release d)lock method

6) What are the libraries in Python that support threads?

\_threading b) Thread c)None **d)thread**

7) Mention the correct syntax for creating Thread object for calling increment methods

a) t1 = threading.Thread() b) t1 = threading.Thread(target)

**c) t1 = threading.Thread(target=incr)** d)t1 = threading.Thr

8) Which leads to concurrency?  
 a) Serialization **b) Parallelism** c) Serial processing d) Distribution

9) Which is not a method for parallelism?

a) Message Passing b) Shared Memory c) Threads d) **Sockets**

10) A race condition occurs when multiple processes or threads read and write

a). Input b). Information c**). Data Items** d). Programs

11) For a single processor system, implementation of semaphores is possible to inhibited through

a) Deadlock b) **Interrupts c)** Lock Step d) Paging

12) Which method controls the execution of thread in python?

a) Wait b) **Sleep c)** Acquire d) Lock

13) How does run() method is invoked?

a) By Thread.create() **b) By Thread.start()**

c) None d) By Thread.run()

14) What is the difference between *threading.Lock* and *threading.RLock*?

a)  Lock and RLock both primitives are owned by a single thread.

b) **Lock is owned by none while RLock is owned by many.**

**c)** Lock is owned by a thread while RLock is owned by many.

d) Lock and RLock both primitives are owned by many.

15) What is the difference between a s*emaphore* and *bounded semaphore*?

1. Semaphore holds a counter for the number of release() calls minus the number of acquire() calls, plus an initial value but bounded semaphore doesn't.
2. A semaphore makes sure its current value doesn’t exceed its initial value while bounded semaphore doesn't.
3. **A bounded semaphore makes sure its current value doesn’t exceed its initial value while semaphore doesn't.**
4. Bounded semaphore holds a counter for the number of release() calls minus the number of acquire() calls, plus an initial value but semaphore doesn't.

16) What is the method that wakes up all thread waiting for the condition?

a) releaseAll() b) notify() c) **notifyAll() d)** release()

17 How to terminate a blocking thread?

a**). thread.stop() & thread.wait() b)** thread.stop()

c) thread.terminate() d) None

18. Which synchronization method is used to guard the resources with limited capacity, e.g. a database server?

a) Event b) Condition c) Lock d) **Semaphore**

19. How to detect the status of a python thread?

a) isActive() b) isDaemon() c) None d) **isAlive()**

20. Execution of several activities at the same time.

a) processing **b) parallel processing** c) serial processing d) multitasking