

1.) A condition in which the critical section (a part of the program where shared memory is accessed) is concurrently executed by two or more threads. It leads to incorrect behavior of a program.

- a. Livelock
- b. Race Condition
- c. Semaphore
- d. Lock

2.) Which of the following is NOT an advantage of multithreading.

- a. It doesn't block the user because threads are independent and you can perform multiple operations at the same time.
- b. You can perform many operations together, so it saves time.
- c. Threads are independent, so it doesn't affect other threads if an exception occurs in a single thread.
- d. Multithreading introduces complexity in your program making it harder to maintain and understand.

3.) Also known as application threads, these are threads that are explicitly created by the programmer to perform specific tasks.

- a. Multi Threads
- b. User Threads
- c. Daemon Threads
- d. all of the above

4.) Threads that provide support to user threads. They run in the background and are considered "service" threads.

- a. Multi Threads
- b. User Threads
- c. Daemon Threads
- d. all of the above

5.) _____ can also be caused by a problem with resource management where a process runs out of resources in the OS because those resources are being utilised by other processes.

- a. Thread Starvation
- b. Deadlock
- c. Race Condition
- d. A and C

6.) _____ can occur in a situation when a thread is waiting for an object lock, that is acquired by another thread and second thread is waiting for an object lock that is acquired by first thread.

- a. Thread Starvation
- b. Deadlock
- c. Race Condition
- d. A and B

7.) _____ provides constructors and methods to create and perform operations on a thread.

- a. Thread Class
- b. Runnable Interface
- c. User Threads
- d. Threads

8.) It is an interface which is implemented by any class if we want that the instances of that class should be executed by a thread.

- a. Thread Class
- b. Runnable Interface
- c. User Threads
- d. Threads

9.) If two Threads (A & B) are ran at the same time. Thread A obtains a lock on resource R1 while Thread B obtains a lock on resource R2. Now, Thread A now requires access to resource R2 which Thread B has exclusive access to. Thread A now waits for Thread B to finish. At the same time, Thread B now needs access to resource R1 which Thread A has exclusive access. Both Threads are now at a standstill and their task is never finished. What do you call this condition?

- a. Deadlock
- b. Thread Starvation
- c. Resource Thrashing
- d. Reentrant Lock

10.) This keyword can be used to obtain a lock on a method or a block of code making sure that it is only executed by a single thread at a time

- a. static
- b. Runnable
- c. synchronized
- d. Livelock