**Q0**) What are the possible state transitions of a process?

- The process can be in any one of the following three possible states.

1) Running (actually using the CPU at that time and running).

2) Ready (runnable; temporarily stopped to allow another process run).

3) Blocked (unable to run until some external event happens).

**Q1**) What are the differences between a thread and a process?

* Process means a program is in execution, whereas thread means a segment of a process.
* A Process is not Lightweight, whereas Threads are Lightweight.
* A Process takes more time to terminate, and the thread takes less time to terminate.
* Process takes more time for creation, whereas Thread takes less time for creation.
* Process likely takes more time for context switching whereas as Threads takes less time for context switching.
* A Process is mostly isolated, whereas Threads share memory.
* Process does not share data, and Threads share data with each other.

**Q2**) What is a race condition?

A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly.

Race conditions are most commonly associated with computer science and programming. They occur when two computer program processes, or threads, attempt to access the same resource at the same time and cause problems in the system.

Race conditions are considered a common issue for multithreaded applications.

**Q3**) Five jobs are waiting to be run. Their expected run times are 9, 6, 3, 5, and *X*. In what order should they be run to minimize average response time? Given X = 10 and X = 1

-

**Q4**) Five batch jobs *A* through *E*, arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time.

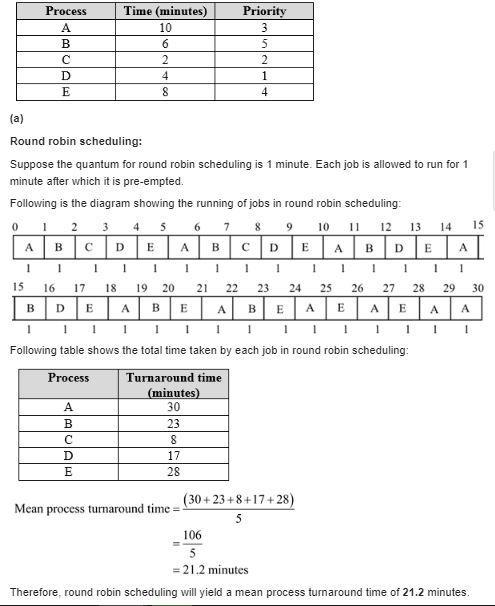
(a) Round robin (RR=4).

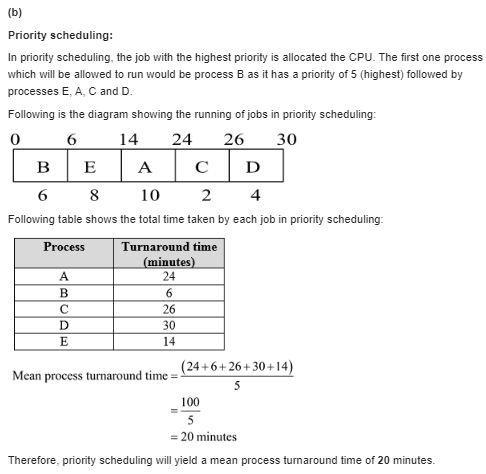
(b) Priority scheduling.

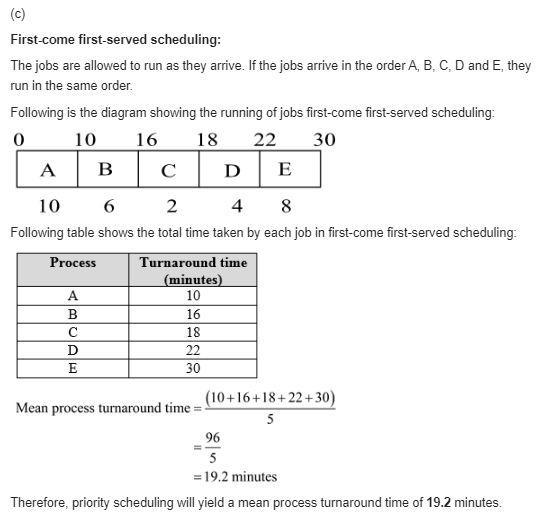
(c) First-come, first-served (run in order 10, 6, 2, 4, 8).

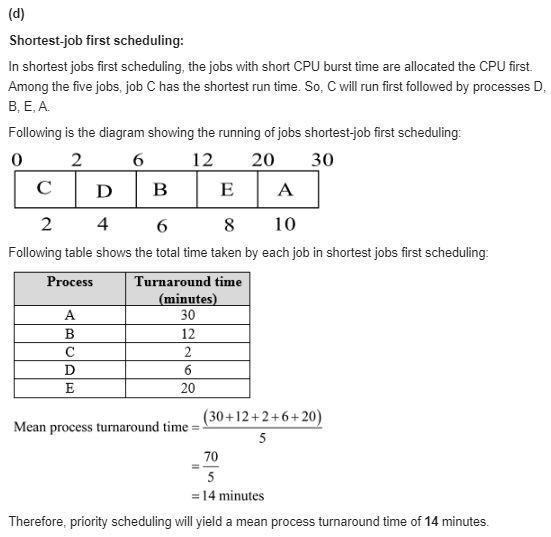
(d) Shortest job first.

For (a), assume that the system is multiprogrammed, and that each job gets its fair share of the CPU. For (b) through (d) assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.









**Q5)** What is the difference between preemption and non-preemption in the context of process scheduling.

**Key Differences Between Preemptive and Non-Preemptive Scheduling:**

1. In preemptive scheduling, the CPU is allocated to the processes for a limited time whereas, in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to the waiting state.
2. The executing process in preemptive scheduling is interrupted in the middle of execution when higher priority one comes whereas, the executing process in non-preemptive scheduling is not interrupted in the middle of execution and waits till its execution.
3. In Preemptive Scheduling, there is the overhead of switching the process from the ready state to running state, vise-verse and maintaining the ready queue. Whereas in the case of non-preemptive scheduling has no overhead of switching the process from running state to ready state.
4. In preemptive scheduling, if a high-priority process frequently arrives in the ready queue then the process with low priority has to wait for a long, and it may have to starve. , in the non-preemptive scheduling, if CPU is allocated to the process having a larger burst time then the processes with small burst time may have to starve.
5. Preemptive scheduling attains flexibility by allowing the critical processes to access the CPU as they arrive into the ready queue, no matter what process is executing currently. Non-preemptive scheduling is called rigid as even if a critical process enters the ready queue the process running CPU is not disturbed.
6. Preemptive Scheduling has to maintain the integrity of shared data that’s why it is cost associative which is not the case with Non-preemptive Scheduling.