Q. What is Synchronization? Give a real-life example for better sense.

In Operating Systems (OS), synchronization refers to the coordination of concurrent processes or threads to ensure that they operate correctly and efficiently when accessing shared resources like memory, files, or devices.

Imagine two people (Alice and Bob) trying to write on the same whiteboard at the same time. If they both try to write simultaneously, their writings could overlap and become unreadable — this is like a **race condition**.

To **synchronize** their access, they decide to use a key to the whiteboard room:

- Only one person can have the key at a time.
- Whoever has the key can enter and write.
- When done, they give the key to the next person.

This key acts like a **mutex** (mutual exclusion), ensuring that only one person accesses the resource (the whiteboard) at a time.

Q. Why Synchronization is Needed?

- **Prevent Race Conditions**: When two or more processes access shared data at the same time, and the outcome depends on the order of execution.
- **Ensure Data Consistency**: Prevent inconsistencies due to simultaneous updates.
- **Avoid Deadlocks**: Carefully control how resources are accessed to avoid situations where processes wait forever for each other.
- **Order of Execution**: Make sure certain tasks happen in a specific sequence.