C++ Multithreading Concepts - Simple Definitions & Scenarios

Data Race

A data race happens when two or more threads access the same memory at the same time, and at least one of them writes to it, without proper synchronization. The result is unpredictable.

Common Scenario: Two threads both try to update a shared counter at the same time, leading to random final results.

Mutex (Mutual Exclusion)

A mutex is a lock that allows only one thread at a time to access a piece of code or data.

Common Scenario: Two threads need to update a shared bank account balance, so they take turns using a mutex.

Lock

A lock is the action of acquiring control over a mutex to safely access shared data.

Common Scenario: Before writing to a file, a thread locks the mutex so no other thread can write at the same time.

Deadlock

Deadlock occurs when two or more threads are stuck waiting for each other’s resources, so none of them can continue.

Common Scenario: Thread A holds Lock1 and waits for Lock2, while Thread B holds Lock2 and waits for Lock1.

Race Condition

A race condition is when the program outcome depends on the timing of thread execution. It may or may not fail depending on how threads are scheduled.

Common Scenario: Two threads read-modify-write a shared variable; depending on timing, the result changes.

Condition Variable

A condition variable lets threads wait until a certain condition is met, usually used with a mutex.

Common Scenario: A producer thread waits until there is space in a queue, while a consumer waits until data is available.

Atomic Operation

An atomic operation is performed as a single, uninterruptible step, safe from race conditions.

Common Scenario: Incrementing a shared counter with an atomic variable so multiple threads can update it safely.

Thread Join

Joining a thread means waiting for it to finish execution before continuing.

Common Scenario: The main thread waits for worker threads to finish before printing the final result.

Thread Detach

Detaching a thread means it runs independently and cannot be joined later.

Common Scenario: Launching a background thread that logs events while the main program continues.