

# 24 Lecture - CS506

## Important Subjective

Certainly, here are 10 short subjective questions related to advanced concepts in "More on Multithreading," along with their answers:

**\*\*Question 1: Explain the concept of a semaphore in multithreading.\*\***

**\*\*Answer:\*\* A semaphore is a synchronization primitive that controls access to a shared resource. It allows a specified number of threads to access the resource concurrently, while preventing excessive access that could lead to contention.**

**\*\*Question 2: What is the purpose of a barrier in multithreading?\***

**\*\*Answer:\*\* A barrier is used to synchronize a group of threads, forcing them to wait until all threads have reached the barrier before proceeding. It's particularly useful for scenarios where multiple threads need to complete a specific phase of execution before moving forward.**

**\*\*Question 3: Describe the difference between static and dynamic thread pools.\*\***

**\*\*Answer:\*\* A static thread pool has a fixed number of pre-initialized threads that are reused for executing tasks. A dynamic thread pool adjusts the number of threads based on the workload, creating new threads as needed and removing idle threads.**

**\*\*Question 4: How does parallelism differ from concurrency in multithreading?\***

**\*\*Answer:\*\* Concurrency involves managing multiple tasks simultaneously, often with context switching between tasks. Parallelism is about executing multiple tasks concurrently on separate processing units, achieving true simultaneous execution.**

**\*\*Question 5: What are the potential challenges of data parallelism in multithreading?\***

**\*\*Answer:\*\* Data parallelism involves dividing a task into smaller sub-tasks that are executed in parallel. Challenges include load balancing, ensuring that sub-tasks are evenly distributed across threads or cores, and managing inter-thread communication.**

**\*\*Question 6: How can the `volatile` keyword impact multithreaded programming?\***

**\*\*Answer:\*\*** The `volatile` keyword ensures that a variable's value is always read from and written to the main memory, preventing compiler optimizations that might cause visibility issues between threads.

**\*\*Question 7:** Explain the concept of cache coherency in multithreading.\*\*

**\*\*Answer:\*\*** Cache coherency ensures that multiple threads accessing shared data see consistent values. It involves coordinating the updating of cached values across different CPU cores to avoid reading stale or inconsistent data.

**\*\*Question 8:** Describe how a latch differs from a barrier in multithreading.\*\*

**\*\*Answer:\*\*** A latch is a synchronization mechanism that allows one or more threads to wait until a certain condition is met. A barrier forces a group of threads to wait until all threads have reached the barrier, and then releases them simultaneously.

**\*\*Question 9:** What is thread migration in multithreading?\*

**\*\*Answer:\*\*** Thread migration refers to the movement of a thread from one processor core to another. This can happen for load balancing purposes or to take advantage of available resources.

**\*\*Question 10:** How does multithreaded programming impact memory management?\*

**\*\*Answer:\*\*** Multithreaded programming requires careful memory management to avoid issues like data corruption due to concurrent access. Techniques like thread-local storage and proper synchronization mechanisms are employed to manage memory effectively.