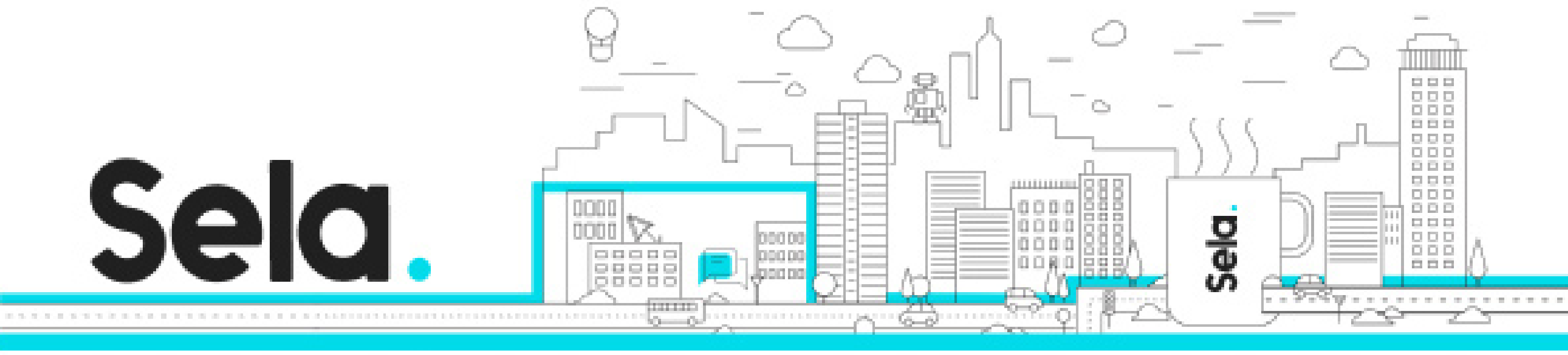
Sela.

ModCPPmulti

Concurrency and Multithreading with Modern C++

college@sela.co.il 03-6176666



Concurrency and Multithreading with Modern C++

ModCPPmulti - Version: 1

2 days Course

Description:

Course will combine theory and hands-on live code demonstration.
Attendees will practice learnt concepts and techniques by solving specially designed exercises.
Hands-on practice is ~30% of each session time.

Intended audience:

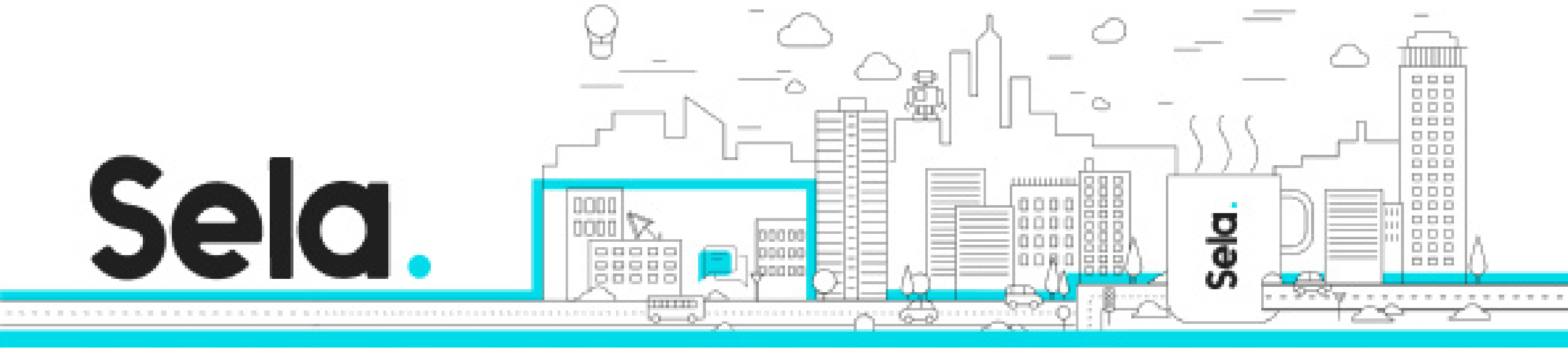
Software engineers with hands-on experience using C++11/14/17. Prior multithreading experience is not required but is recommended.

Prerequisites:

Objectives:

Cover C++ memory model.
Covers multithreading features in C++17 (including C++14/11)
Covers Multithreading and Concurrency features in C++20.
Covers C++17 parallel standard algorithms.
Covers best practices and common pitfalls.

Topics:



Introduction to Multithreading and Concurrency

- What is parallelism?
- What is concurrency?
- Thread vs. Process
- Review of modern CPUs and memory architecture
- Amdhal Law.

std::thread

- What is a thread – OS perspective
- Starting new threads
- Passing arguments to new threads
- Background threads
- Getting results from new threads
- Join and dispatch
- Yield and sleep

Sharing State

- Problems with sharing state
- Threads and global variables
- Threads and thread_local variables
- Atomic operations.
- Once flags
- Race conditions.
- Synchronization with mutex
- Mutex types

Mutex locks

- Working with RAII-based locks
- Unique locks
- Deadlocks
- Deadlocks avoidance

Multiple Readers Single Writer Problem

- Shared Locks
- Why no upgradable locks
- Bias and Fairness

Condition Variables

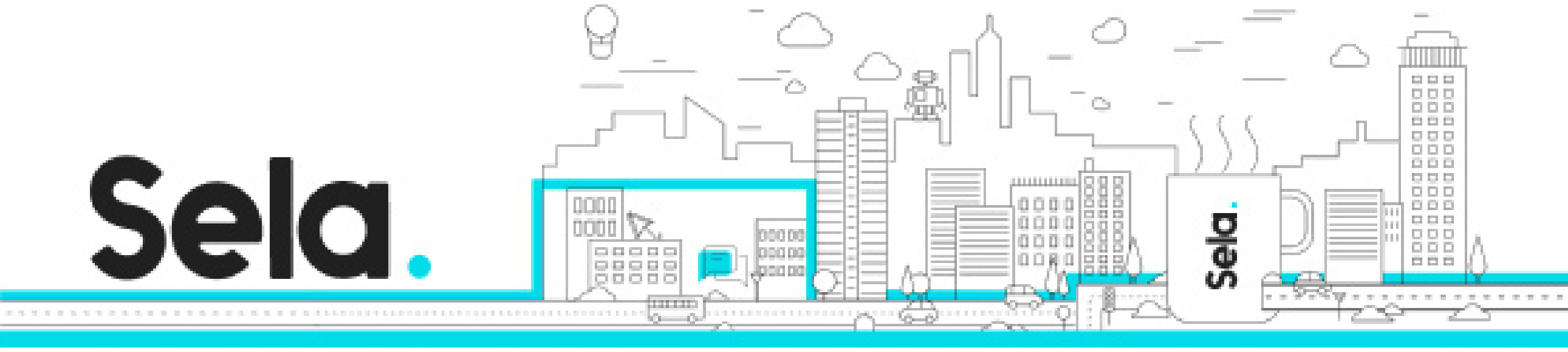
- Condition variables and mutex
- Implementing waitable data structures.
- Spurious wakeup and avoidance

Async Processing

- std::async and std::future
- Packaged tasks

C++20 Joinable Thread

- jthread
- Stop tokens



Additional Synchronization Tools

- Counting semaphores
- Barrier
- Latch

Parallel standard algorithm in C++17

- Execution policies in C++17 and C++20
- What algorithms are available?
- Demo with Intel Thread Building Blocks