

Advanced C++: STL, Templates & Qt/QML

6-Day Schedule (With Pre & Post Tests)

6 Days

DURATION

11

MODULES

46

TOPICS

8

HANDS-ON ACTIVITIES

COURSE OVERVIEW

This course covers the transition from C to C++ and core object-oriented concepts including classes, constructors, static members and friend functions. It progresses through inheritance, polymorphism, operator overloading, exception handling and RAII, then moves into templates and the Standard Template Library, covering containers, iterators and algorithms. Advanced features such as lambda functions, smart pointers, multithreading, linking and performance profiling are addressed, along with socket-based networking. The course closes with Qt/QML integration for building event-driven GUI and networked applications, reinforced through mini-projects and hands-on assignments.

WHO THIS COURSE IS FOR

C++ Developers

Software Engineers

Application Developers

Systems Programmers

Qt/QML Developers

LEARNING OBJECTIVES

- Explain the evolution from C to C++ and key language differences
- Apply object-oriented concepts such as classes and constructors
- Implement inheritance and polymorphism using virtual functions
- Apply operator and function overloading in practical scenarios
- Implement exception handling and RAII for robust resource management
- Build generic code using function and class templates
- Apply STL containers, iterators and algorithms to manage data
- Develop multithreaded and networked applications using Qt/QML

11 Modules, 46 Topics

00

Introduction & Foundation Enhancer (Add-on)

TOPICS

- Evolution of C to C++
- Procedural vs Object-Oriented Programming Paradigm
- Key Differences Between C and C++ (struct vs class, malloc/free vs new/delete, return codes vs exceptions, etc.)
- Case-study based learning approach (problem-solving examples to learn programming)

01

C++ Basics (as in Original TOC)

Add emphasis: provide sample codes for each concept

TOPICS

- Classes and Objects (with banking system mini-case)
- Access Specifiers (public, private, protected)
- Constructors & Destructors (default, parameterized, copy constructor)
- Static Members (data & methods)
- Friend Functions & Friend Classes

02

Inheritance & Polymorphism

Case Study

TOPICS

Single & Multiple Inheritance · Virtual Functions & Dynamic Binding · Abstract Classes & Pure Virtual Functions · Virtual Destructors & VTable Mechanism

03

Compile-time Polymorphism

TOPICS

- Function Overloading (calculator mini-case)
- Operator Overloading (Complex numbers, Point comparison)

04

Case Study

Exception Handling & Robustness

TOPICS

try, catch, throw · Custom Exception Classes · RAII (Resource Acquisition Is Initialization)

05

Case Study

Templates & Generics

TOPICS

Function Templates · Class Templates

06

Case Study

Standard Template Library (STL)

TOPICS

- **Containers:** vector, list, map, set, queue, stack
- Iterators
- Algorithms (sort, find, count, etc.)

07

Case Study

Advanced C++ Features

TOPICS

- Lambda Functions (C++11+)
- Smart Pointers (unique_ptr, shared_ptr)
- File Handling

MULTITHREADING & CONCURRENCY

- Threads (std::thread)
- **Async** (std::async, future)
- Deadlocks & Synchronization (mutex, lock_guard, condition variables)

STATIC VS DYNAMIC LINKING

- Explanation & Demonstration with sample project

PERFORMANCE CONSIDERATIONS

- Debugging memory leaks (Valgrind/Visual Studio tools)
- Performance profiling (code optimization practices)

08

Case Study

Networking & System Integration (Add-on)

TOPICS

- **TCP vs UDP:** conceptual explanation with examples
- Socket programming basics in C++

09

2 Hands-on Demos

Integration with Qt/QML & Network Programming

Half-day dedicated as requested

TOPICS

Introduction to Qt Framework · Building UI with QML · Event-driven OOP in Qt · Qt Signals and Slots (async-like programming) · Networking with Qt (TCP/UDP sockets in Qt)

10

Wrap-up & Evaluation

TOPICS

- Quizzes & Polls after each module
- Hands-on Coding Assignments

MINI-PROJECTS

- Banking System
- Library Management System
- Shape Drawing Application with Qt GUI
- Performance Debugging Exercise (identify & fix bottlenecks)
- Final Demos & Participant Presentations

HANDS-ON ACTIVITIES

Practise What You Learn

#	MODULE	ACTIVITY
02	Inheritance & Polymorphism	Case Study: Designing a Shape hierarchy and extending it for real-world use cases
04	Exception Handling & Robustness	Case Study: Building robust file reader with exception handling
05	Templates & Generics	Case Study: Generic stack implementation
06	Standard Template Library (STL)	Case Study: Student records search and analysis
07	Advanced C++ Features	Case Study: Producer-Consumer problem
08	Networking & System Integration (Add-on)	Case Study: Client-server chat simulation

#	MODULE	ACTIVITY
09	Integration with Qt/QML & Network Programming	HANDS-ON DEMOS ■ Basic GUI demo (calculator, form app) ■ Networking demo (simple chat app with TCP/UDP in Qt)

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

Classroom Training

Live Online Classes (ILO)

Fly-Me-A-Trainer (FMAT)

Flexi (Self-Paced)

1-on-1 Training

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

Visit koenig-solutions.com

✉ info@koenig-solutions.com