Enterprise C++

6-Day Schedule (With Pre & Post Tests)

Module 0: Introduction & Foundation Enhancer (Add-on)

- 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)

Module 1: C++ Basics (as in original TOC)

(Add emphasis: provide sample codes for each concept)

- 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

Module 2: Inheritance & Polymorphism

- Single & Multiple Inheritance
- Virtual Functions & Dynamic Binding
- Abstract Classes & Pure Virtual Functions
- Virtual Destructors & VTable Mechanism
- **Case Study:** Designing a Shape hierarchy and extending it for real-world use cases

---

Module 3: Compile-time Polymorphism

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

Module 4: Exception Handling & Robustness

- try, catch, throw
- Custom Exception Classes
- RAII (Resource Acquisition Is Initialization)
- **Case Study:** Building robust file reader with exception handling

---

Module 5: Templates & Generics

- Function Templates
- Class Templates
- **Case Study:** Generic stack implementation

Module 6: Standard Template Library (STL)

- Containers: vector, list, map, set, queue, stack
- Iterators
- Algorithms (sort, find, count, etc.)
- **Case Study:** Student records search and analysis

Module 7: Advanced C++ Features

- 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)

    - **Case Study:** Producer-Consumer problem
- **Static vs Dynamic Linking**
    - Explanation & Demonstration with sample project
- **Performance Considerations**
    - Debugging memory leaks (Valgrind/Visual Studio tools)
    - Performance profiling (code optimization practices)

Module 8: Networking & System Integration (Add-on)

- TCP vs UDP → conceptual explanation with examples
- Socket programming basics in C++
- **Case Study:** Client-server chat simulation

Module 9: Integration with Qt/QML & Network Programming

(Half-day dedicated as requested)

- 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)
- **Hands-on Demos:**
    - Basic GUI demo (calculator, form app)
    - Networking demo (simple chat app with TCP/UDP in Qt)

Module 10: Wrap-up & Evaluation

- 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