C++ Internals

Duration: 3 days

Course Overview

This workshop is designed to provide participants with concepts, techniques and lessons from experience that will help them be more successful in the C++ design and programming.

Course Highlights

- C++ memory Internals
- Object Construction Internals
- Object Destruction Internals
- C++ Idioms
- Virtual function internals
- Exception internals

Course Coverage and Duration

This is a **three-day** intense course for developers and application architects.

Day 1

- Compile time layout of application
- Runtime memory layout of application
- Function Internals
- Object Model
- Object Initialization & Cleanup

Day 2

- Dynamic Memory Management
- Virtual Functions
- Exception Handling

Day 3

- RTTI
- Advanced Overloading
- Closures
- Multi-Threading

Detailed course contents

Day 1

Compile time layout of application

Source files (.cpp, .h)
Object files (.obj)

Executable File Format
Complie time v/s link time v/s Runtime
Preprocessor Definitions/ Compile time switches
What is a namespace?
What happens to namespace at runtime ?

Runtime memory layout of application

Code Segment
Stack
Heap
Data Segment
CPU Registers
Static link library v/s Dynamic link library

Function Internals

Function Stack Frame
Calling Conventions
Naming Conventions
Parameters, Arguments & return value
Overloading vs Generic Function
Function Template Internals
Library Design Issues with Generic

Object Model

Simple Object Model
Table driven object model
C++ object model
Class internals
Generic class
Class Template Full specialization
Class Template Partial specialization
Template Bloating

Object Initialization & Cleanup

Compiler Synthesized Constructor & Destructor
Constructor Internals
Destructor Internals
Deep copy v/s Shallow copy
Explicit constructor
Copy Constructor v/s Assignment operator
Initialization v/s Assignment
Order of Initialization

Day 2

Dynamic Memory Management

Types of new operator

- Preventing Heap based objects
- Preventing Stack based objects
- Preventing destroying object instance
- Preventing object Instance
- Identifying object is on Heap or Stack
- Reference Counting
- Smart pointers

Constructor and Exception

Deep copy and Shallow Copy

The reference count technique

- Singleton pattern

Virtual Functions

- Internals
- Dual Dispatching
- Casting Internals
- Object Slicing
- Virtual Inheritance

Exception Handling

- Exception Internals
- Resumption v/s Termination
- Exceptions in Constructor
- Exceptions in Destructor
- Object Slicing in Exceptions

Day 3

RTTI and Casting Operators

Runtime Polymorphism & RTTI

typeid & type_info

The Casting operators

const_cast, static cast, reinterpret cast, dynamic cast

The Dual Dispatch technique

Operator Overloading Technique

The Proxy Technique

Lazy Load

Smart Pointer

Overloading member callback operator

Core language usability enhancements

Initializer lists

Uniform initialization

Type inference

Range-based for loop

Lambda functions and expressions

Explicit conversion operators

Alias templates

Variadic templates

Multithreading model

Static assertions

Post Training Reading

1) C++ Primer by Lippman.
2) Inside Objects by Lippman
3) C++ By Bjarne Stroustrup
4) C++ Bible by AL Stevens
5) Complete Reference by Herbert Schildt.
6) Advanced C++ james coplin
7) Effective C++ Scott mayor
8) More Effective C++ Scott mayor
9) Effective STL Scott mayor
10) Design Patterns (GOF)