C++ Training

Duration: 4 full days

Key Takeaway:

Delegates will learn:

1. Will learn how to code from an architecture point of view.
2. Understand function and variable basics from point of c++.
3. Will understand how C++ is different from C.
4. Will learn the meaning and implementation of OOPs.
5. Will understand how to use templates, STL.
6. Will learn how to debug the c++ code and profile the code also.
7. Understand version of c++ and features.

Course Materials:

- Soft and online references

Lab requirements:

Native environment on participant's PC

1. G++ compiler 9.X plus
2. Visual studio code for Linux/Windows
3. Ubuntu /Windows

Day 1

An Overview of C++

- What is Object Oriented Programming?
- Versions of C++ available in the market and compiler support
- Version of c++ and their evolution
- C++11 to c++17,
- C++ Console I/O
- C++ Comments
- Variables and functions in c++
- Namespaces.
- Introducing function overloading
- Using default arguments
- Overloading and ambiguity
- Classes: A First Look, class as a datatype or model
- Variables and different ways of initialization and its implications.
- Some Differences between C and C++
- C++ keywords
- References – Value & R-value
- Returning references
- static cast
- Independent references and restrictions
- assert
- Move operations intro
- Return type optimization. intro

INTRODUCING CLASSES

- This Pointer
- Constructor and Destructor Functions
- Overloading constructor functions
- Constructors that take parameters
- Object Pointers
- Classes, Structures and Unions
- In – Line functions
- Automatic In-lining

Day 2

ARRAYS, POINTERS AND REFERENCES

- Arrays of objects
- Using Pointers to objects
- Using new and delete
- Passing references to objects
- Assert usage in code
- Finding the address of an overloaded function
- Const expr
- static assert

A CLOSER LOOK AT CLASSES

- Creating and using a copy constructor
- Assigning objects
- Passing objects to functions
- Returning objects from functions
- Restructuring.
- An introduction to friend functions
- Static Class members
- Const Member functions and mutable
- A final look at constructors
- rule of 3 and rule of 5 in c++
- Move operations, move constructor. Conclusions
- Return type optimization conclusions.

OPERATOR OVERLOADING AND CONVERSION FUNCTIONS

- The basic operator overloading
- Overloading binary operators
- Overloading the relational and logical operators
- Overloading a unary operator
- Using friend operator functions
- A closer look at the assignment operator
- Overloading the [ ] subscript operator
- Creating a conversion function

Day 3

Creating a new data type based on existing classes.

Dependency

Association

- Composition
- Aggregation.

INHERITANCE

- Base class and derived class
- Derived class object creation and what it can access in base class and derived class.
- Constructors, Destructors during inheritance
- Multiple inheritance and need for Virtual base classes
- Protected scope
- Mode of inheritance

VIRTUAL FUNCTIONS

- Pointers to Derived classes
- Introduction to virtual functions
- More about virtual functions
- Applying polymorphism

RUN-TIME TYPE IDENTIFICATION

- Understanding run-time type identification
- Using dynamic_cast

Abstract class and interface context.

- Pure virtual functions
- Abstract class
- Interface usage

CASTING OPERATORS

- Using const_cast, reinterpret_cast

Exception Handling
Handling exceptions thrown by new

Library making and profiling code basics.

Day 4

TEMPLATES AND EXCEPTION HANDLING

- Generic functions
- Generic classes
- Template specialization.
- Auto and decltype.
- Variadic templates and parameter packs.

STL

- Introduction
- Containers

- Iterator
- Function Pointers
- Need for Lambda.
- Algorithms
- Smart pointers, unique ptr and shared ptr.
- Threads in c++ (std threads)
- Std::optional, Std::variant, std::any, std::string_view
- Introduction to boost library.
- Unit testing in c++ quick intro.