Bosch
Global
Software
Technologies
alt_future

Thub

Advance C++ with features of 11,14,17 & 20

Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
V 0.3	05-Jun-2024	Ganesh A M P	Ganesh Babu	Sandhana Mohan	ToC to be used only for SAT Open Calendar which is customized

Course Title: "Advance C++ with features of 11,14,17 & 20"

1. **Course Summary:** Modern C++ 11/14/17/20 a practical approach

2. Pre-Requisite: Exposure to Traditional C++ programming and the following topics and OO concepts is a necessity:

- Virtual functions and run-time binding
- C++ object model & this pointer.
- const correctness, const methods & const objects
- static data and static methods
- mutable data member
- Operator overloading
- Functors and Function objects
- constructor/destructor behaviour with regard to inheritance
- At Least a few SOLID principles.
- Familiarity with certain OO design methodologies:
  - PIMPL model
  - RAII technique and the rule of three.
  - Smartpointers

3. Target Audience

- Audience with at least 3 years experienced in C++ project development.
- Target Batch Size: 20 participants

4. Hardware & Network Requirements

System with either windows O/S or Linux with access to internet.

5. Software Requirements

- Visual Studio Code C++ or related C++ compiler installation

6. Learning Objectives:

At the end of the training, the participants will be able to:

- Understanding and implementing modern C++ features and syntax across C++11, C++14, C++17, C++20.
- Mastering memory management and threading using modern C++ libraries and features.

7. Course Content with Lab/Hands-on Activity (day wise break-up):

Day 1:

- Introduction to Templates and Template Argument Deduction & Instantiation
- Explicit Specialization and Non-type Template Arguments
- STL - (std::vector, std::pair, std::map, std::unordered_map,)
- Tuple, std::tie
- Type inference mechanism (C++11)
- Constant Expressions - constexpr (C++11) and std::initializer_list (C++11)
- Lambda Expressions and Function Wrappers (C++11)
- Generic Lambda, Arbitrary capture (C++14)

Day 2:

- Lambda Expressions and Callbacks Revisited - Function Pointers (C++11)
- Lambda Expressions - Internals and Lambda Expressions Capture List (C++11)
- Variadic lambda (C++ 14)
- constexpr lambda (C++17)
- Template syntax for lambda's (C++20)
- Lambda capture of parameter_pack (C++20)
- Function Wrappers (Function type and Binding)
- Deprecate implicate capture of 'this' (C++20)
- RValue Reference and move semantics
- Reference collapsing and perfect forwarding.
- Move copy and move assignment

Day 3:

- Modern C++ class features (C++11)
- Designated initializers (C++20)
- Range-based for loop (C++11)
- Range-based for loop with initializer (C++20)
- Variadic templates (C++11) and fold expressions (C++17)
- Template argument deduction for class templates (C++17)
- Declaring non-type template parameters with 'auto' (C++17)
- type_traits and static assertion (C++11)
- concepts (C++20)

Day 4:

- nested namespaces (C++17)
- structured binding (C++ 17)
- inline variables (C++ 17)
- std::variant (C++17)
- std::optional (C++17)
- std::any (C++17)
- std::unordered_map
- tuple, pair, std::tie
- std::apply, std::transform
- Memory management libraries [unique_ptr, shared_ptr & weak_ptr] (C++11)

- make_shared supports arrays (C++20)
- New identifiers (import, module) (C++20)

Day 5:

- Calendar and time zone library
- Threading: Creating and terminating threads using
- modern features (Call_once function, Thread Local Storage, Tasks using 'async' and 'future').
- Synchronization: Mutexes and Atomic operations
- std::string functions (ends_with(), starts_with()) (C++20)
- Array bounded/unbounded and std::to_array (C++20)
- Likely and unlikely attributes (C++20)

C++ Evaluation / Post test

Subject to availability of time:

- Regular Expression support in C++11

8. Course Structure:

Activity	Indicative Number of Hours
Pre-Read Hours	
Teaching Hours	15 hrs
Hands on Sessions Hours	25 hrs
Assignments & Tutorial Hours	
Mock Project Hours	
Lab/Tool Access post Training	

9. Course Evaluation:

Method of Assessment	Yes/No	Weightage
Pre-Assessment	yes	20
Mid-Assessment	no	--
Post-Assessment	yes	25
Project Work	yes	25

10. Course Resources:

Would be shared with the participants during the training.

a. C++ notes
b. C++ examples
c. PPT slides
d. Additional C++ examples for further reading

11. Recommended Reading Links:

a. https://www.learncpp.com/
b. https://www.modernescpp.com/index.php/what-is-modern-c
c. https://thispointer.com/c11-tutorial/

12. Course Owner (s):

Employee Name	Employee Mail ID	Business Unit
Ganesh Babu Gunasekaran	ganeshbabu.gunasekaran@in.bosch.com	MS/ECX4-TM-XC
Hassan Suresh Thejas	thejas.hassansuresh@in.bosch.com	(BGSW/PJ-ETA-SAT)