Basic Language Facilities

- What is C++
- First C++ Program
- The C++ Compilation Process
- Primitive Types & Variables
- Basic Input/Output
- Functions Basics - Part I
- Functions Basics - Part II
- Overview of Debugging in Visual Studio
- Reference Vs Pointer
- The const Qualifier
- const Qualifier & Compound Types
- Assignment
- Automatic Type Inference (C++11)
- Range-Based for Loop (C++11)
- Default Function Arguments
- Inline Functions
- Function Pointers
- Namespace

Memory Management Part 1

- Dynamic Memory Allocation using (malloc function)
- Dynamic Memory Allocation with new operator)
- Dynamic Memory Allocation with new [] operator)
- Dynamic Memory Allocation with (2D arrays)

Classes and Objects

- Object Oriented Programming Basics
- Class
- Constructor & Destructor
- Structures
- Non-static Data Member Initializers (C++11)
- this Pointer
- Static Class Members
- Constant Member Functions
- Copy Constructor
- Delegating Constructors (C++11)
- Default & Deleted Functions (C++11)

Operator Overloading

    - Basics
    - Assignment Operator

    - Global Overloads
    - Friend Keyword
    - Smart Pointer Basics
    - Smart Pointers in C++11
- Type Conversions
    - Basics
    - Primitive to User Type
    - User to Primitive Type
    - User Defined to User Defined
- Initialization Vs. Assignment & Member Initialization List

Memory Management Part II

    - Raw Pointers
    - std::unique_ptr
    - Sharing Pointers
    - Sharing std::unique_ptr
    - std::shared_ptr
    - Weak Ownership
    - std::weak_ptr Internals
    - Circular References
    - Deleter
    - Dynamic Arrays
    - Make Functions

More C++ Goodies

- Enums
    - Basics
    - Scoped Enums C++11
- Strings
    - Raw Strings
    - std::string
    - String Streams
- User-Defined Literals
    - Constant Expressions - constexpr (C++11)
    - std::initializer_list (C++11)
    - Dynamic Array (std::vector)
- Union

OOPs

- Inheritance & Composition
- Inheritance & Access Modifiers
- Project
- C++11 Inheriting Constructors
- Virtual Keyword

- Virtual Mechanism Internals
- override & final specifier in C++11
- Object Slicing
- typeid Operator
- dynamic_cast Operator
- Abstract Class
- Multiple (Diamond) Inheritance

Exception Handling

- Basic
- Multiple Catch Blocks
- Stack Unwinding
- Nested Exceptions
- Constructor & Destructor
- noexcept keyword in C++11

Templates

- Introduction to Templates
- Template Argument Deduction & Instantiation
- Explicit Specialization
- Non-type Template Arguments
- Perfect Forwarding (C++11)
- Variadic Templates (C++11)
- Class Templates
- Class Template Explicit Specialization
- Class Template Partial Specialization
- Typedef, Type Alias & Alias Templates (C++11)
- Type Traits (C++11)
- static_assert (C++11)

Lambda Expressions

- Callbacks Revisited - Function Pointers
- Callbacks - Function Objects
- Lambda Expressions
- Lambda Expressions - Internals
- Lambda Expressions Capture List
- Generalized Lambda Capture

Standard Template Library

- Introduction
- std::array (C++11)
- std::vector

- std::deque
- std::list & std::forward_list (C++11)
- Sequence Containers
- std::set & std::multiset
- std::map & std::multimap
- Associative Containers
- Unordered Containers (C++11)
- Unordered Containers (C++11)
- std::hash (C++11)
- Unordered Containers
- Big O Notation & Performance of Containers
- Algorithms
- Container Changes in C++11
- STL Project

C++ Concurrency

- Concurrency Source Code
- Concurrency Basics
- Thread Creation (std::thread)
- Passing Arguments To Threads
- Thread Synchronization (std::mutex)
- std::lock_guard
- std::thread Functions & std::this_thread Namespace
- Task Based Concurrency
- std::future Wait Functions
- Using std::promise
- Propagating Exceptions Across Thread

C++ 17 Core Language Features

- Deprecated & Removed Features
- Changes
- Attributes
- Feature Test Macros
- If & switch With Initialization
- inline Variables
- Nested Namespaces
- noexcept
- constexpr Lambda
- Structured Bindings
- Expression Evaluation Order