

Develop the ability to design and write programs in the C++ language, emphasizing object-oriented approaches to designing solutions.

Audience: C programmers responsible for the development of advanced applications or systems programs in C++.

Prerequisites: *C Programming*

Number of Days: 5 days

1. **Classes**
 - Creating a Data Structure
 - Methods
 - Object Scope
 - C++ Input and Output
 - Namespaces
 - Data Abstraction
 - Enforcing Data Encapsulation
 - File Organization
 - Classes in C++
 - Objects
 - this Pointer
2. **Constructors and Destructors**
 - Debug Output
 - The Default Constructor
 - When are Constructors Called?
 - The Destructor
 - The Copy Constructor
 - Other Constructors
 - Why Did It Work Before?
 - Composition
 - The Report Class
 - Code Reuse
 - Initialization Lists
3. **Inheritance**
 - Inheritance
 - Bugreport
 - Protected Access Modifier
 - Access and Inheritance
 - Constructors and Inheritance
 - Initialization Lists Revisited
 - Multiple Inheritance
4. **Virtual Functions**
 - Inheritance and Assignment
- Inside Report's Assignment Operator
- Using Pointers – a Quick Look at Basics
- Class Assignment and Pointers
- Static Binding
- Dynamic Binding
- Polymorphism
- The show_rep() Function
- Using the show_rep() Function
- Designing Member Function Inheritance
5. **Pure Virtual Functions**
 - Bugfix and Its Relationship with Bugreport
 - Bugfix: Association with Bugreport
 - Using Bugfix with show_rep()
 - Adding Bugfix to the Hierarchy
 - Coding for the Document Class
 - Reexamining the Document Class
 - Pure Virtual Functions
 - Updated: Designing Member Function Inheritance
6. **References and Constants**
 - References
 - Displaying References
 - Changing References
 - Pass by Reference
 - Returning by Reference
 - Constant Variables
 - Constant References
 - Constant Methods
7. **new and delete**
 - new and delete
 - Array Allocation
 - The Report Class

- Compiler Version of the Copy Constructor
- Guidelines for Copy Constructors
- The Report Constructors and new
- The Report Destructor and delete
- Virtual Destructors
- 8. Casting in C++**
 - Casting: A Review
 - New Casting Syntax
 - Creating a String Class
 - The String Class
 - The Conversion Constructor
 - Expanding Our Casting Options
 - Casting Operator
 - Using the Casting Operator
- 9. Class Methods and Data**
 - Class Data
 - Class Methods
 - Using the New Data
 - More on Class Methods
- 10. Overloaded Functions**
 - Function Overloading
 - Using Overloaded Functions
 - Rules for Overloading
 - Overloading Based on Constness
 - Default Arguments
 - Invoking Functions with Default Arguments
- 11. Overloaded Operators**
 - The Basics of Overloading
 - Overloading operator+
 - Coping with Commutativity
 - Non-Commutative Operators
 - friends and Their Problems
 - The Assignment Operator
 - Overloading the << Operator
 - Using Date with cout
- 12. Exception Handling**
 - Why Exception Handling?
 - try / catch / throw
 - Exception Classes
 - Standard Exception Hierarchy
 - Multiple catch Blocks
 - Catching Everything
 - Unhandled Exceptions
 - Exception in Constructors and Destructors
- 13. Designing for Exceptions**
 - Standard Template Library**
 - Class Template Concepts
 - Standard Template Library (STL) Overview
 - Containers
 - Iterators
 - Iterator Syntax
 - Non-Mutating Sequential Algorithms
 - Mutating Sequential Algorithms
 - Sorting Algorithms
 - Numeric Algorithms
 - auto_ptr Class
 - string Class
- 14. STL Containers**
 - Container Classes
 - Container Class Algorithms
 - vector Class
 - Additional vector Class Methods
 - deque Class
 - list Class
 - set and multiset Classes
 - map and multimap Classes
- 15. Appendix A – Reference Sheets**
 - Constants, References, and Pointers
 - Input/Output
 - this Pointer
 - The Complete Report/Document Hierarchy
- 16. Appendix B – Templates**
 - Scenario
 - Designing an Array Class
 - Code for FloatArray
 - Code for IntArray
 - Templates
 - Template Syntax
 - Using Templates
 - Using Classes with Templates
 - Additional Template Features
 - Standard Template Library
- 17. Appendix C – Sample Problems**
 - Banking System
 - Library Card Catalog
 - Diagrams for Banking and Library Problems
 - Object Diagram - Banking
 - Event Trace Diagram - Banking

Object Diagram – Library

Event Trace Diagram - Library

18. Appendix D – Other C++ Features

Namespaces

The static_cast and reinterpret_cast
operator

The dynamic_cast operator

The const_cast operator

mutable Data Members

The bool Datatype

new Operator Failure