COURSE DESCRIPTION: C++ For Experienced Programmers

Absolute Software Co., Inc.

Duration: 5 days **Format:** Lecture/Workshops **Maximum Size:** limited

Overview:

This course teaches the student how to write high quality, internally documented, well-structured C++ programs. Students will learn how C++ supports software engineering principles, such as abstraction, information hiding, localization, and modularity, and how to apply these principles in software development. Students will see how C++ incorporates and improves upon ISO C, then adds features supporting object-oriented design & programming, error management, and templates. The student will gain experience with syntax and semantics of the ISO standard C++ language. Finally, we will look at how the new Standard Library reduces the amount of code that needs to be developed and improves reliability through re-use.

After this course a student should be able to:

- ❑ Build C++ programs using object-oriented programming
- ❑ Avoid common pointer problems
- ❑ Use C++ to support software re-use efforts
- ❑ Use C++ to develop more reliable programs
- ❑ Contribute to the design of C++ software applications

In the lectures, extensive examples will be used to illustrate the new features of C++. In hands-on workshops, the students will practice using the C++ features in typical C++ development environments.

Target Population: Software development personnel, including their management and QA engineers, who intend to program in C++, design for C++, or review C++ code.

Prerequisites: Programming experience, familiarity with a high-level language.

Materials: Each student will receive a copy of all lecture materials, lab notes, and reference materials.

Topics:

- Introduction
  - History of C++
  - Overview of C++ features
- Basic Constructs
  - Programs
  - Compilation units
  - Preprocessor Directives
  - Declarations
  - Built-in types
  - Pointers, References, Arrays, & Strings
  - Structures & Unions
  - Constants & Literals
  - Quick look at Stream I/O
  - Expressions & Statements
- Functions
  - Function prototypes
  - Default parameters
  - Parameter passing
- Organizing Code
  - Files
  - Namespaces
- Classes
  - Data abstraction
  - Encapsulation
  - Information hiding
  - Member data & function
  - Inline functions
  - Constant & Static members
  - Constructors & destructors
  - Friends
- Derived Classes
  - Single inheritance
  - Multiple inheritance
  - Virtual inheritance
  - Polymorphism
  - Function overloading
  - "Run-time" binding
  - Virtual functions
  - Base class access control
  - Run-time type identification
- Operator Overloading
- Templates
  - Template classes
  - Template functions
- Exception Handling
  - Catch
  - Throw
  - Try
- C++ Standard Libraries Overview
  - Standard Template Library (STL)
    - Containers
    - Utilties
    - Iterators
    - Algorithms
  - Strings
  - I/O
    - Basic streams
    - File I/O with streams
    - Stream class hierarchy