Course Content

C++ Complete

Format: Five Days
Max Capacity: 6

At the end of this course, delegates will be able to develop complex object-oriented C++ console programs that can take advantage of multi core processors and implement parallelism. The course will also cover efficient C++ programming. No prior C++ knowledge is assumed although knowledge of the Microsoft Visual Studio development environment is required. This course is a very intense 5-day course and is not suitable for absolute beginners. As such some prior programming knowledge is essential.

Course Content

Introduction
- C++ programs
- Components of a C++ program
- A simple C++ program
- Adding two numbers together
- Variables
- Keywords

C++ Simple Data Types & Expressions
- Fundamental data types
- Declaring variables
- Integer numbers
- Integer operators
- Compound assignments
- Increment and decrement
- Prefixand postfix operators
- Floating point numbers
- Character data
- Boolean data
- Unsigned variables
- Constants
- Type conversion
- The side of operator

Composite Data Types
- Typedefs
- Enumerations
- Arrays
- The #include compiler directive
- Structures
- Uninions
- Bit structures
- Structure alignment

Flow Control
- Sequential statements
- The if statement
- The while statement
- The for statement
- Range based for loop
- Break and continue statements
- The switch statement
- The do while loop
- The ternary conditional operator

Continued on next page...

F1 Computer Services – Tel 01202 876529
Email: training@f1co.co.uk
www.f1co.co.uk

Course Content

C++ Complete

Format: Five Days
Max Capacity 6

Course Content (continued)

Functions

- Modular programming
- Header files
- Source files
- Function Declarations
- Calling Functions
- Function definition
- Returning values using the return statement
- Function overloading
- Default arguments
- References
- Passing parameters
- Pasting copied parameters - by value
- Passing referenced parameters - by reference
- Argument types
- Return types
- Inline functions
- Scope of variables
- Storage class
- Automatic local variables
- Global variables
- Extern variables
- Variable linkage
- Dynamically allocated objects

Dynamic Memory Allocation

- The stack & the heap
- Memory leakage
- The new & delete operators
- Placement new operator
- Manipulating pointers
- Pointers to structures
- Strings in C++
- Pointers to char
- Character manipulation routines
- Strlen()
- Strcpy() and Strncpy()
- Strcat() and Strncat()
- Strcmp() and Strncmp()
- Character test functions
- Character conversion functions
- Microsoft & C++ extentions to character
- Manipulation routines
- Tokenising strings with Strtok_s()
- Null pointers
- C++11 Nullptr
- Smart pointers (C++11)

More Pointers

- Pointers to arrays
- Pointer arithmetic
- The void pointer
- Pointers to functions
- 64 bit pointers
- Common pitfalls

F1 Computer Services – Tel 01202 876529
Email: training@f1co.co.uk
www.f1co.co.uk

Course Content

C++ Complete

Format: Five Days
Max Capacity 6

Course Content (continued)

Architectural Issues
- Namespaces
- Header & source files
- #Defined constants
- #Defined macros
- Deleting a definition

Files In C++
- Files - review
- File types
- Text file examples
- Binary file examples

The Principles Of Object Orientated Programming - Review
- Object orientated principles

Classes & Objects
- Class implementation
- Writing classes
- The public interface
- The private interface
- Class de inition
- Inserting class definitions
- Using objects
- Object lifetime
- Constructors
- Destructors

Static Class Members
- Static class data members
- Static class member functions

The C++ Const Modifier
- Objectives
- Constant variables & pointers
- Constants & functions
- Constant class member functions

Class Hierarchies & Inheritance
- Chapter objectives
- Inheritance review
- Deriving classes
- Public, private and protected inheritance
- Member function access is derived classes
- Protected access
- Member access rights
- Base class initialisation
- Class scoping issues
- Multiple inheritance
- Constructors & destructors
- Inheritance
- Inheritance guidelines

Polymorphism
- Polymorphism - review
- Virtual functions
- Virtual destructions
- Abstract base classes
- Polymorphism guidelines

Continued on the next page...

F1 Computer Services – Tel 01202 876529
Email: training@f1co.co.uk
www.f1co.co.uk

Course Content

C++ Complete

Format: Five Days
Max Capacity 6

Course Content (continued)

Advanced Casting Operators
- Casting - Review
- Advanced casting operators
- Static-Cast
- Dynamic_Cast
- Const_cast
- Reinterpret_cast
- The mutable keyword
- The Typeid (C++11) operator

Advanced Constructors & Operator Overloading
- Copy constructors
- Move constructors
- Conversion constructors
- The explicit keyword
- Operator overloading
- Operator =
- Move assignment
- Operator +
- Operator ++
- Boolean operators
- Overloading the <<&>> operators
- Overloading the subscript operator
- Global operators

Exception Handling
- Chapter objectives
- Exception
- Exceptions during the object construction
- Exception function throw lists
- Assertions
- Statis_assert C++11
- Error handling methodologies

Standard Template library
- STL- overview
- Strings
- Vectors
- Queues
- Stacks
- Lists
- Maps
- Iterators
- STL Algorithms
- Which STL container?
- STL conclusion

Templates
- Template functions
- Template classes
- Creating template classes
- Friends and templates

Introduction to Lambda Expressions
Pointers to functions review

Continued on the next page...

F1 Computer Services – Tel 01202 876529
Email: training@f1co.co.uk
www.f1co.co.uk

Course Content

C++ Complete

Format: Five Days
Max Capacity 6

Course Content (continued)

C++11, C++14, C++17

- C++11
- Auto
- Nullptr
- Range-based for loops
- Override and final
- Strongly-typed enums
- Smart pointers
- Non-Member begin () and end()
- Lambdas
- Static_assert and type traits
- Move semnatics
- Uniform initialisation syntax
- Deleted and defaulted functions
- Delegating constructors
- Threading library
- C++11 feature list
- C++14
- Function return type deduction
- Alternate type deduction on declaration
- Relaxed constexpr restrictions
- Variable templates
- Aggregate member initialisation
- binary literals
- Digit Seperators
- Generic Lambdas
- Lambda capture expressions
- The attribute [[deprecated]]
- Shared mutexes and locking
- Heterogeneous Lookup in associative
- Containers
- Standard user-defined literals
- Tuple addressing via type
- Smaller library features
- C++17

Multithreading In C++

- What is multi-threading?
- When to use multiple threads
- Advantages of multiple threads
- Disadvantages of multiple threads
- Creating your own threads
- Starting threads
- Identifying threads
- State a thread with Lambda
- Thread synchronisation issues
- Using Mutex
- Exception and locks
- Automatic lock management
- Recursive locking
- Timed locking
- Call once
- Condition variables
- Atomic types
- C++11 synchronisation benchmark
- Semaphore
- Thread suspend, resume & terminate
- Thread priority & processor affinity
- Thread local storage
- Threading conclusion

F1 Computer Services – Tel 01202 876529
Email: training@f1co.co.uk
www.f1co.co.uk