f1 computer services

Course Content

C++ Expert

Format: Three Days
Max Capacity: 6

This course is designed as a follow on from the C++ 3-day Introduction & 2 day intermediate courses. It is only open to delegates who have completed those courses and preferably used the knowledge gained from them for several months. 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.

Course Content

Composite Data Types
- Unions
- Bit structures
- Structure alignment

Functions
- Calling convention cdecl, fastcall, stdcall &this call
- C calling convention (__cdecl)
- Standard calling convention (__stdcall)
- Fasting caling convention (++fastcall)
- Thiscall
- Inline functions

Dynamic Memory Allocation
- Memory leakage
- The new & delete operators
- Malloc
- Free
- Calloc
- Realloc
- Placement new operator

Pointers
- Pointers to char
- Character manipulation routines
- Strlen()
- Strcpy() and strncpy()
- Strcat() and Strncat()
- Strcmp() and Strncmp()
- Character test functions
- Character conversion functions
- Microsoft & C++11 extentions to character
- Manipulation routines
- Tokenising strings with Strtok_s()
- Null pointers
- C++11 nullptr
- Smart pointers (C++11)

More Pointers
- Pointer Arithmetic
- The void pointer
- Pointers to functions
- 64 bit pointers

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

Continued on the next page...

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

Course Content

C++ Expert

Format: Three Days
Max Capacity 6

Course Content (continued)

Functions

- Modular programming
- Header files
- Source files
- Function Declarations
- Calling Functions
- Function definition
- Returning alues using the return statement
- Function overloading
- Default arguments
- References
- Passing parameters
- Pasing copied parameters - by value
- Passing referenced parameters - by reference
- Arguement types
- Return types
- Inline functions
- Scope of variables
- Storage class
- Automatic local variables
- Global variables
- Extern variables
- Variable linkage
- Dynamically allocated objects
- 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 arrrays
- Pointer arithmetic
- The void pointer
- Pointers to functions
- 64 bit pointers
- Common pitfalls

Continued on the next page...

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

Course Content

C++ Expert

Format: Three Days
Max Capacity 6

Course Content (continued)

Multiple Inheritance
- Constructors & Destructors in multiple inheritance
- Multiple inheritance from a common base class
- Alternatives to multiple inheritance
- Inheritance guidelines

Polymorphism
- Polymorphism - review
- Virtual functions
- Virtual destructors
- Pure virtual functions
- Abstract base classes
- Polymorphism guidelines

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

Introduction to Lambda expressions
- Pointers to functions review

The Boost Library
- Chapter objectives
- Boost
- Boost example

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_asset and type traits
- Move semantics
- 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

Continued on the next page..

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

Course Content

C++ Expert

Format: Three Days
Max Capacity 6

Course Content (continued)

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

- Digit separators
- 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 multithreading
- When to use multiple threads
- Advantages of multiple threads
- Disadvantages of multiple threads
- Creating your own threads
- Starting threads
- Identifying threads
- Start a thread with a lambda
- Thread synchronisation issues
- Using a mutex
- Exceptions 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