



TRAINING IN REAL-TIME  
EMBEDDED DEVELOPMENT

## Course C++-502: Real-Time C++

### Course Description:

This course introduces the C++ language for use on real-time applications. The first part of the course focuses on the language itself, highlighting the areas of concern for real-time development. The latter part covers the application of C++ to real-time systems focusing on concurrency/threading issues. These are fundamental issues that are core to any C++ Course.

### Overview:

A five day course covering C++ in general for the first three days, and real-time issues on the remaining two. 50% percent of the course is spent on practical work.

### Course Objectives:

- To provide a solid understanding of the essentials of the C++ programming language.
- To give you practical experience of writing C++ for real-time applications.
- To give you the confidence to apply these new concepts to your next real-time project.

### Delegates will learn:

- The core C++ syntax and semantics
- About memory and performance issues associated with C++
- How threading affects the use of the language

### Pre-requisites:

- A working knowledge of C

### Who Should Attend:

This course is designed for real-time engineers who are embarking on a project using C++ for the first time. It is also targeted at developers currently reluctant to move to C++ from C as they believe it poses too great an overhead. This course will clearly demonstrate both the strengths and weaknesses of C++ versus C.

### Duration:

Five days.

### Course Materials:

- Delegate handbook

### Related Courses:

- OO-503 Real-Time Systems Design with UML 2.0
- OO-303 Applying Real-Time UML
- AC++-501 Advanced C++ for Embedded systems
- C++-501 C++ for Embedded Developers
- C++-503 C++ for Non-C programmers
- C-501 C for Real-Time Developers
- RTOS-201 Fundamentals of Real-Time Operating Systems

### Course Workshop:

All sections are backed up with comprehensive exercises. Exercises include: developing UML-based associations; implementing design patterns; and developing solutions to multi-threading problems. The threading exercises use the Win32 API.

### Course Outline:

#### Introduction to real-time systems

- What is a real-time computer system
- The need for a rigorous development procedure

#### From C to C++

- Non object-oriented C++ enhancements to basic C
- Conveniences of C++ over and above C

#### Introduction to Object Oriented (OO)

##### Principles

- Key characteristics of OO development
- OO techniques and the real-time software development process

#### Introduction to Classes

- Classes & class instances
- Methods
- Constructors & destructors

#### More on Classes

- Inlining member functions
- const member functions
- static class members and functions
- arrays of classes
- implementing object relationships

#### Inheritance

- Building class hierarchies
- Dynamic binding for class methods, virtual functions
- Polymorphism

#### Multiple inheritance (MI)

- MI and interfaces

#### Functions and Operators

- Class defined conversions
- Overloading and function selection
- Friend functions
- Overloading operators
- Dynamic memory allocation revisited

#### Exception Handling

- What are exceptions?
- Throwing and Catching exceptions
- Rethrowing exceptions
- Catch all handlers
- Exception specifications

#### Templates

- Introduce parameterised types and functions
- Function templates
- Class templates

#### The Standard Library

- Introducing the Standard Template Library

#### Software Structuring

- Structuring large scale software systems
- Separate implementation from interface header files
- Dealing with name conflicts
- Linking with other languages

#### Patterns:

- Pattern origins and history
- Singleton
- Proxy
- Command

#### Implementing Associations:

- Associations
- Uni- and Bidirectional 1:1, 1:N, 1:\*
- Aggregation & Composition aggregate

#### Threading Basics:

- Concurrency;
- CreateThread API.

#### C++ Threading:

- Thread-Runs-Function
- Thread-Is-Polymorphic object;
- Thread-Runs-Polymorphic object.

#### Mutual Exclusion:

- Resource access in multitasking systems
- Mutex Class
- Mutex Interface
- Guards
- Template Locking Policy

#### Mutual Exclusion Issues:

- Priority Inversion
- Deadlock

#### Thread Synchronisation:

- Signalling
- Condition Objects

FEABHAS

#### Feabhas Ltd

5, Lowesden Works  
Lambourn Woodlands  
Hungerford, Berkshire  
RG17 7RY, UK

Tel: +44 (0) 1488 73050

Fax: +44 (0) 1488 73051

#### Email:

info@feabhas.com

#### Web:

www.feabhas.com