



Real-Time Design Patterns for Embedded Systems

Course category	C++ Training Courses
Training area	Design Techniques
Course code	DP-303
Duration	3 days
Price exc VAT	£1600.00
Additional information	Contact Feabhas for delivery dates

From 2011, the ISO C++ standard included a library of concurrency (multi-threading) features. However, this library made the implicit assumption that your C++ code would be executing in a hosted environment – Linux, Windows or similar.

For many embedded systems this is not the case - they are based around real-time operating systems (RTOS), which often have different characteristics or fewer features. In such situations the C++ standard concurrency library is unlikely to be available, or suitable.

This course is designed to introduce C++ developers to the concepts of concurrent, multi-threaded design within the constraints on an embedded environment. The focus of the course is developing the core features of an object-oriented RTOS abstraction layer using proven design patterns and idioms.

Many of the concepts developed in the course are present in the C++ standard library, so attendees will gain a unique insight into those features. This should significantly improve their programming should they transition to a hosted environment.

This three day course will provide practical, hands-on experience with the core real-time design patterns, uniquely addressing their suitability to a non-PC programming environment.

Course objectives:

- To understand what design patterns are and how the language of patterns can aid designers and developers to be more productive
- To develop your understanding of concurrency and how it relates to program design
- To understand the issues with multi-threaded programs and how patterns can help to circumvent these issues.
- To provide practical experience of building concurrent, multi-threaded systems.

Pre-requisites:

- Good knowledge of C++
- An understanding of Object-Oriented principles
- An understanding of multi-threading is useful, but not essential
- UML class modelling is useful, but not essential

Who should attend:

The course is aimed at software developers, designers, and architects who are looking to deploy C++ in a multi-tasking, embedded environment.

It is also suitable for those wishing to improve their object-oriented design skills.

Duration:

- Three days

Course materials:

- Delegate handbook

Course workshop:

The course exercises are designed to reinforce the attendee's knowledge of concurrency, C++ and design patterns.

Attendees will construct a set of framework of concurrency patterns and apply those to an embedded target application.

Course exercises are designed to build upon each other, resulting in a complete, operating, multi-threaded application. The attendees will be required to code, download and debug using PC-based development tools.

Design patterns

- What is a pattern?
- The benefit of patterns
- Categorising patterns

Concurrency architecture

- Concurrency design principles
- Active and passive objects
- Communication between objects
- Governors for selection active objects

- Refining your concurrency design

Implementing concurrency

- Parallelism versus concurrency
- Processes, threads and tasks
- Simulating concurrency
- Scheduling patterns

Thread patterns

- Thread-Runs-Function
- Thread-Is-Polymorphic-Object
- Thread-Runs-Polymorphic-Object

Mutual exclusion

- Data-race conditions
- Critical sections
- Semaphores and Mutexes
- The Scope-Locked idiom

Mutual exclusion issues

- Priority inversion
- Priority inheritance
- Deadlock, livelock and starvation
- Recursive deadlock
- Cyclic dependency deadlocks
- Deadlock-through-death

Thread synchronisation

- Busy waiting
- Synchronisation objects
- Synchronisation patterns
- Semaphore-as-Signal
- Condition objects
- Monitor pattern
- Single-Writer-Multi-Reader locks

Communication patterns

- Asynchronous message
- Synchronous message
- Promise and Future

Higher-order patterns

- Proxy
- Observer
- Command
- State

Feabhas Ltd - UK Office 15-17 Lotmead Business Park, Wanborough, Swindon, SN4 0UY, UK ·

+44 (0) 1793 792909 · info@feabhas.com · www.feabhas.com