

Design Patterns in C++ for Embedded Systems

Course category	C++ Training Courses
Training area	Design Techniques
Course code	DP-403
Duration	4 days
Additional information	For price: Contact Feabhas (available for on-site delivery only)

Everyone seems to be talking about design patterns these days.

This course is designed to provide delegates with a basic understanding of design patterns and how they can be applied to real-time C++ embedded systems.

It addresses the following questions:

- What are design patterns and why do I need to know about them, especially in embedded systems?
- Who are the “Gang of Four” and how is their work relevant?
- Which patterns are easy to work with and are most commonly found in embedded systems?
- Are there any specific patterns to embedded systems?
- What is the difference between a pattern and an idiom?

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

Course objectives:

- Provide an understanding of the principles of object oriented design and how they relate to patterns.
- Provide practical experience of working with Design Patterns.
- Provide an understanding of the significant “Gang of Four” set of classical patterns and patterns associated specifically with multi-tasking embedded systems.
- Demonstrate how the patterns language can be used to document bad designs as, so called, “anti-patterns”.
- Gain the confidence to apply these new concepts to your next real-time project.
- Develop an understanding of Object Orientated principles through the lessons of others that are captured by design patterns.

Pre-requisites:

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

Who should attend:

The course is aimed at software developers, designers, and architects wishing to improve their object-oriented design skills.

Duration:

- Four days

Course materials:

- Delegate handbook

Course workshop:

The course exercises are designed to foster an understanding of design patterns, object orientation, and C++.

Ample opportunity is provided for delegates to consider the implications of patterns to the size and space concerns of embedded systems whilst reflecting on the broader quality issues that they directly address.

Overview of OO Principles

- Constructing objects
- Associations
- Composition
- Specialisation
- Polymorphism
- Abstract base classes
- Interfaces

Construction patterns

- Singleton
- Factory
- Abstract Factory
- Builder

Changing the interface

- Smart pointers
- Adapters
- Port
- Decorator

Behavioural patterns

- Bridge
- Remote Proxies
- Virtual Proxies
- Protection Proxies

Event-driven patterns

- Callback
- Observer
- Command

Changing behaviour

- State
- Strategy

Real-time patterns

- Thread-Runs-Polymorphic-Object
- Mutual exclusion
- Scope-locked idiom
- Monitor
- Asynchronous message
- Future
- Promise

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

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