

## Advanced C Programming

**Level:** intermediate / advanced

**Length:** 35 – 40 hours

**Course objective:** learn and exercise how to design & organize the code by using C features for long-lived, large projects

### What you will learn

- Stereotypes and C idioms
- How to apply the design principles and design patterns in C
- How to design a solution and how to map it to the code
- Introduction to concurrent programming with POSIX threads
- Exercise soft skills of communication and presentation

**Course Audience:** C programmers who want to deepen their understanding of using C for solving problems, how to identify and implement the entities required in programs, how to organize and structure the software to face the volume and time

**Prerequisites:** practical experience and knowledge of C at least at medium level

**Required facilities:** VGA projector, white board, computers, C development tools. It's highly recommended using an IDE, a good (free) example is Microsoft Visual C++ Express Edition or Code Blocks

**Related courses:** The C Programming Language, Design Patterns

**Minimal bibliography:** The C Programming Language, Brian W. Kernighan, Dennis M. Ritchie, ed. 2, 1988

## Description

This course is targeted to C programmers who want to deepen their knowledge about the language and the ways to use it correctly & efficiently in long-lived, large projects.

The training is highly interactive, the attendees are implied in discussing the ideas and in designing solutions which are ultimately expressed in C. The main purpose of this training is to show how to be fluent in C, how to design, organize the code in order to have a maintainable & expandable code, how to manage the dependencies.

It is very important to write good algorithms and good data structures, but of an equal importance is how all these are organized, how they are packed in order to have a flexible and expandable structure, with well-defined interfaces and responsibilities.

The training is a good occasion to learn & apply the design principles, patterns, idioms and rules for a good programming applied to the C world. And last but not least we cover best practices in C for writing defensive and secure code.

We enter the concurrent programming by using POSIX threads by introducing the concurrent ecosystem provided by operating systems, how to use it in C.

**Note:** the subjects are adapted to the attendees' profile, their background, experience and goals. We can approach other subjects depending on the context.

## Contents

1. Outline the language constructs; life cycle of code, how the code is organized
2. How to define and implement modules to represent unique entities
3. How to define and implement modules to represent several entities of the same type – design decisions, consequences; abstract data types in C
4. Pointers to functions, how to use them for solving problems
5. SOLID design principles, how are they applied in C
6. Design patterns in C - the observer pattern, callbacks; interpreter pattern (optional)
7. Finite state machines, standard implementations in C, patterns state, exercise how to implement a FSM by using UML
8. Embedded programming with C – the language support, specific issues; build a proxy for a memory mapped device
9. Defensive programming, secure programming, best practices – their focus & goals, techniques, examples
10. Code optimizations techniques
11. Concurrent programming with POSIX threads – the ecosystem, main entities, their roles, problems & how are they used. Main issues in concurrent programming, how are they solved. Implementation of several concurrent patterns like balking, guarded suspension, and active object. How to write an emulator of a hardware device.