

Advanced C Programming

Audience Course Advanced C Programming

The course Advanced C Programming is intended for C developers who want to understand and use the more advanced features and techniques of the C language.

Prerequisites Course Advanced C Programming

To join this course knowledge of the basic concepts of [programming in C](#) and experience in C is required.

Realization Training Advanced C Programming

The concepts are treated on the basis of presentation slides. Illustrative demo programs are used to clarify the treated concepts. There is ample opportunity to practice and theory and practice is interspersed.

Official Certificate Advanced C Programming

Participants receive an official certificate Advanced C Programming after successful completion of the course.

Duration: 3 days

Price: € 1999

[Open Schedule](#)



Advanced
C Programming



Content Course Advanced C Programming

In the course Advanced C Programming the more advanced aspects of C programming like the use of modules and interfaces and object oriented programming in C using encapsulation are discussed.

Potential Dangers in C

Potential dangers of certain constructs in C are highlighted such as C macros, evaluation order and the use of globals.

Pointer Techniques

Attention is paid to pointer techniques and the use of pointers to functions is discussed.

Dynamic Data Structures

The most common dynamic data structures such as arrays, linked lists, hash tables and trees and their implementation in C are addressed in detail.

Optimization Techniques

Also attention is paid to optimization techniques regarding speed and memory.

Advanced C Library Function

More advanced functions of the C library such as setjmp, longjmp, signals, bsearch, and qsort etc. are also on the course program.

Bit Manipulation

Finally bit manipulation and advanced string handling and parsing are discussed.

Modules Course Advanced C Programming

Module 1 : C Pitfalls	Module 2 : Object Orientation in C	Module 3 : Pointers and Arrays
C Standards Pitfalls C Language Write Clean Code Good Programming Style Rules for Functions Deep versus Flat Code Evaluation Order Avoid Macros Without const Using const Code Reuse Avoid Globals Pre and Post Conditions Invariants Code Smells	Object Orientation Classes and Objects Example Class and Objects Object Orientation in C Simple Objects in C C versus C++ Class Constructor in C Member Functions in C Encapsulation in C Public Function Members in C Inheritance Extending C++ Classes Virtual Functions Virtual Function Table Polymorphism	Pointers Revisited Pointer Arithmetic Pointers and Arrays Function Pointers Optional Ampersand Variable Length Arguments Varargs Example Array Initialization Array Traversal Arrays of Structs Arrays versus Pointers Multidimensional Arrays Indices in 3d Array Dynamic Arrays Ragged Arrays
Module 4 : Data Structures in C	Module 5 : Bit Manipulation	Module 6 : C Standard Library
Dynamic Data Structures Singly Linked Lists Linked List Element Type Creating List Elements Doubly Linked Lists Stacks and Queues Hash Tables Load Factor Hash Functions Trees Trees Traversal	Bit Manipulation Bitwise Operators Bitwise AND Bitwise OR Bitwise XOR Bitwise NOT Bitshift Operators Bit Shifting Bitwise Assignment Operators Bit Rotation BitFlag Functions	What are Signals? ANSI C-Signal Types Handling Signals SIGABRT, SEGV, SEGFPE setjmp and longjmp Coroutines atexit Function assert Function perror Function Raising Signals Alarm Signal
Module 7 : String Handling		
Looking for Characters Looking for Substrings Count Matching Characters Looking for Character Sets String Comparison String Tokenizing Converting Strings to Numbers Handling Conversion Errors qsort and Bsearch Advanced String Handling		