

Advanced C Programmeren

Doelgroep Cursus Advanced C Programmeren

De cursus Advanced C Programmeren is bedoeld voor C developers die zich willen verdiepen in de meer geavanceerde functies en technieken van de taal C.

Voorkennis Cursus Advanced C Programmeren

Om aan deze cursus deel te nemen is kennis van en ervaring met [_programmeren in C](#) vereist.

Uitvoering Training Advanced C

De theorie wordt behandeld aan de hand van presentatie slides. Illustratieve demo's worden gebruikt om de behandelde concepten te verduidelijken. Er is voldoende gelegenheid om te oefenen en afwisseling van theorie en praktijk. De cursustijden zijn van 9.30 tot 16.30.

Officieel Certificaat Advanced C Programmeren

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Advanced C Programmeren.

Duur: 3 dagen

Prijs: € 1999

[Open Rooster](#)



Advanced
C Programming



Inhoud Cursus Advanced C Programmeren

In de cursus Advanced C Programmeren worden geavanceerde aspecten van de programmeertaal C behandeld, waaronder het gebruik van modules en interfaces, object georiënteerd programmeren en encapsulation in C.

Potentiële Gevaren in C

Ook wordt aandacht besteed aan de potentiële gevaren die bepaalde constructies in C hebben, zoals C macro's, evaluatie volgorde en het gebruik van globals.

Pointer Technieken

Verder komt het gebruik van verschillende pointer technieken aan de orde evenals het gebruik van pointers naar functies.

Dynamische Datastructuren

De meest voorkomende dynamische datastructuren zoals arrays, linked lists, hash tables en trees en hun implementatie in C worden eveneens in detail besproken.

Optimalisatie Technieken

En ook wordt aandacht besteed aan optimalisatie technieken voor wat betreft executie snelheid en geheugen gebruik.

Advanced C Library Functies

Verder staan de meer geavanceerde functies van de C library zoals setjmp, longjmp, signals, bsearch en qsort etc. op het programma.

Bit Manipulatie

Tot slot is er aandacht voor bit manipulatie en geavanceerde string handling en parsing.

Modules Cursus Advanced C Programmeren

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 Conversion Strings to Numbers Handling Conversion Errors qsort and Bsearch Advanced String Handling		