Programming in C

★★★★★

SII-161 **5 Days (35 Hours)**

Description

This intensive course will allow you to acquire a truly operational knowledge of the language. He will explain to you how the different mechanisms work and show you their implementation through numerous practical exercises. At the end of this internship, you will be able to write robust and portable C programs.

Who is this training for ?

For whom

Developers, engineers, project managers close to development.

Prerequisites

Aucune

Training objectives

- Master the production chain of a program written in the C language
- Implement the operators, expressions and control structures of the C language
- Manipulate data structures, tables, pointers and character strings
- Organize the code a program using functions.
- Exploit the main standard libraries of the C language

Training program

Premiers pas en C

- Presentation of the C language, its advantages.
- C++ compared to C.
- C++11 and C11 standards.
- Source files (.
- c, .
- h).
- General structure of a program.
- Basic syntax of the language.
- Data types and basic constants.
- Global and local variables.
- Storing and passing parameters.
- Formatted inputs/outputs.
- Comments.
- Basic use of the production chain.
- Editing, compilation and 'execution.
- Execution of a first program.

Opérateurs et expressions

- Arithmetic operators.
- Expression evaluation mechanisms.
- Post and decrement pre-increment.
- Precedence and associativity of operators .
- Assignment operators.
- How logical expressions work.
- Logical expressions in while, if statements.
- Comparison operators: , >, ==, !=.
- Logical operators: AND, OR, negation.
- Compound numeric types.
- Rule conversions in mixed expressions.
- Implicit/explicit conversions.
- Initialization of variables.
- Arithmetic on addresses.
- Input/output formats associated with numeric types.
- Bitwise operators: AND, OR, exclusive OR, 1's complement, negation.
- Shift operators: >> , .
- Conditional expression with the ternary operator.
- Practical work Implementation of operators and expressions.

Structures de contrôle

- Concept of blocks.
- Loop structures: while, for.
- Loop control instructions: break, continue.
- Choice structures: if, else, else if.
- Multiple choice structure: switch.
- Practical work Implementation of control structures.

Tableaux, pointeurs et chaînes de caractères

- Definition, initialization and access to the elements of an array.
- Definition of a pointer.
- Retrieve the memory address of an object.
- Access the content of a pointer.
- Pointer/array equivalences.
- Calculations on pointers.
- Character strings.
- Examples of manipulating character strings.
- Unicode character strings of C11.
- Practical work Manipulation of arrays, pointers and strings of characters.

Les structures

- Interests of structures.
- Declare, initialize and access the fields of a structure.
- Use nested structures.
- Create new new types using Typedef.
- Bit fields.
- Unions.
- Enumerations.
- Anonymous structures and enumerations of C11.
- Define pointers to structures.
- Practical work Implementation of new data structures.

Les fonctions

- Definition of a function.
- Calling a function.
- Passing parameters: by value or by reference.
- Code return of a function.
- Return types.
- The "main" function.
- Practical work Cut your code 'function help.
- Handle function calls.

Compilation séparée, classe d'allocation

- Operating mechanism of the production line.
- Use of source library.
- Notion of Makefile.
- Memory configuration of a C program (stack, heap.
-).
- Variable allocation classes (auto, register, static, extern).
- Different cases of figure of the separate compilation.
- Notion of external object.
- Case of global and static data.
- Case of local data.
- Visibility rule.
- Additions on functions and initializations.

Le préprocesseur

- Using predefined macros (symbolic constants).
- Define your own macros with #define.
- Define macros as functions.
- Using # and ## markers.
- Undo constant definition with #undef.
- Conditional compilation: #if, #ifdef, #ifndef, #elif, #endif .
- Include resources with #include.
- Practical work Using preprocessor directives.
- Setting up conditional compilation.

Les bibliothèques standard

- Mathematical calculation functions (sqrt, sin.
-).
- Input/output functions (fprintf, fscanf.
-).
- File access functions (fread, fwrite.
-).
- Character string manipulation functions (strlen, strcat.
-).
- Memory management functions (malloc, free.
-).
- Setting up linked structures (linked lists, n-ary trees.
-).
- The “secure” functions of the C11 standard library (strcat_s, strlen_s, .
-).
- Practical work Use of the main functions of standard libraries.