The C Programming Language

Level: foundation

Length: 35 – 40 hours

Course objective: a solid introduction of the C programming language, learn and apply the main principles and mechanisms of procedural and modular programming

What You Will Learn
- How the C programs are organized, development cycle of writing software, using an integrated development environment (IDE)
- Code structure: data structures, control structures, fundamental, primitives types
- C support for procedural and modular programming
- Exercise problem solving by using C
- Enhance soft skills: communication, team work, presentation

Who can participate: anybody who wants to learn the C programming language

Prerequisites: the attendees have to be familiar to use a computer; it is not required to know another programming language despite this aspect would be an advantage

Required facilities: VGA projector, white board, workstation, C development tools. It's highly recommended the using of an Integrated Development Environment, for example Microsoft Visual C++ Community Edition.

Bibliography:
- The C Programming Language, Second Edition, Brian W. Kenighan & Dennis M. Ritchie, Prentice-Hall 1988, ISBN 0-13-110370-9
- C Primer Plus, Fifth Edition, Stephen Prata, Sams 2005, ISBN 0-672-32696-5

Related courses: Advanced C Programming, The C++ Programming Language

Description

This course is mainly addressed to the programmers who want to use C in their activity. Knowledge of another programming language facilitates the understanding of common issues to programming languages like data structures, flow control, etc.

It's shown the programs' elements, how data is represented and manipulated, how a typical, standalone program is developed starting with editing the code, building, debugging and finishing with the testing.

By examples and practical projects it's exemplified and exercised the using of this language to solve problems. It is covered the whole process of C development

starting from a set of requirements, how they are reflected into the code, how to debug and validate the code, until the final code which fulfils all the functional and non-functional requirements.

This course is based mainly on the reference book where the C language was defined (see the bibliography, Kernigan & Ritchie), the practical projects were taken from real projects.

Note: the standard curricula will be personalized on the atendees' profile, their background, experience and goals.

Contents

1. Introduction to C: variables and arithmetic expressions, the for statement, symbolic constants, character input and output, arrays, functions, arguments, external variables and scope

2. Types, operators and scope: variable names, data types, constants, declarations, arithmetic operators, relational and logical operators, type conversions, increment and decrement operators, bitwise operators, assignment operators and expressions, conditional expressions, precedence and order of evaluation

3. Control flow: statements and blocks

4. Functions and program structure: functions, return values, external variables, scope rules, header files, static variables, register variables, block structure, initialization, recursion, the C preprocessor, file inclusion, macros substitution, conditional inclusion

5. Pointers and arrays: pointers and addresses, pointers and function arguments, pointers and arrays, address arithmetic, pointer arrays, pointers to pointers, multi-dimensional arrays, initialization of pointer arrays, command line arguments, pointers to functions

6. Structures: basics, structures and functions, arrays of structures, pointers to structures, self-referential structures, table lookup, typedef, unions, bit-fields

7. Input and output: standard input and output, formatted output – printf(), variable-length argument lists, formatted input – scanf(), file access, error handling – stderr and exit, line input and output, standard library functions – string operations, storage management, mathematical functions, random number generation

8. Projects: dynamic memory management using either the best fit, or first fit algorithms, logging system