CES7AJ - CES9AK INTRODUCTION PROGRAMMING INTO C /C++

CES7AJ - CES9AK **Introduction into programming with C/C++**	**ECTS Credits : 4** **Duration : 36 hours**	**Semester : S7 + S9**

Person(s) in charge :

Pierre-Etienne MOREAU, Professeur, pierre-etienne.moreau@mines-nancy.univ-lorraine.fr

Keywords : machine language, programming language, microprocessor, compilation, data structures, memory management, automatic building tools

Prerequisites: Being motivated

Objective:

“This course provides a fast-paced introduction to the C and C++ programming languages.
You will learn the required background knowledge, including memory management, pointers, preprocessor macros, object-oriented programming, and how to find bugs when you inevitably use any of those incorrectly.” (MIT 6.S096)

Programs and contents :

Understand in an abstract way how programs are executed on a given hardware

Be able to understand the mechanisms involved in a program

Be able to write or modify a C or C++ program

How does a computer work

- architecture
- executing a program
- processor and binary programs
- low level language and assembly
- high level language, compilation
- abstract machine
- memory, data, program
- introduction to Unix

Compilation pipeline

- pré-processor
- compilation, déclaration rules, typing rules
- modular compilation, linkage
- modular conception of programs
- building tools (makefile)

Syntax and semantics of core C

- elementary data types
- encoding elementary information (int, short, char)
- control structures
- notion of array

Memory organisation

- notion of pointer
- dereferencing a pointer
- head and stack
- memory allocation

Advanced data structures

- record data type
- dynamic allocation of memory and memory management

Introduction to C++

- classes, name spaces
- memory management, constructors and destructors

Object programming

- inheritance, polymorphism
- dynamic method binding
- input/output

Advanced usages

- standard template library
- references to functions
- exceptions
- standardisation

Abilities :

Levels	Description and operational vocabulary
Know	syntax and semantics of C language constructions typing rules the role of each development tool

Understand	elementary mechanisms involved in a computer and the various level of programming languages what can do a preprocessor and a compiler the mechanisms involved in modular compilation notion of pointer and how to manage the memory elementary principles of object programming
Apply	how to setup an elementary environment to develop software how to compile and execute a software on a given architecture
Analyse	be able to choose a data structure to solve a problem be able to de detect programming errors
Summarise	be able to specify a data structure and operators on this data structure be able to implement an algorithm using C programming language
Assess	be able to evaluate the quality of a software and to suggest improvements

Evaluation :

□ Written test	☑ Continuous Control	□ Oral report	☑ Project	□ Report