

IDENTIFICATION

Course code : DEV501
ECTS : 2

DURATION

Lectures : 0
Tutorial classes : 26
Labs : 10
Total : 36

Project :
Personnal work :

ASSESSMENT

Continuous assessments and practical evaluations.

TEACHING MATERIALS

Slides and guided work sheets.

LANGUAGE

English

CONTACT

Pierre Ugo TOURNoux
pierre.tournoux@univ-reunion.fr

Modifié le : 14 juillet 2026

UE : UE5-DEV

COURSE : Algorithms and Data Structures in C

SCOPE OF THE COURSE

OBJECTIVES

The aim of this course is to learn programming by reusing classic techniques and data structures while integrating complexity constraints. The required language is C, which will provide a deep understanding of data structure implementation, particularly useful in embedded systems.

COMPETENCE AND SKILLS

- Apply a methodology and formalism to design and develop an application
- Write and implement software tests

PROGRAM

General presentation of the C language : instructions and basic types, functions, pointers, files...

- **Program development** Develop a correct program to solve problems using an iterative process, documentation of program components, and consultation with program users. Use appropriate abstractions to facilitate program writing : collections, procedures, application programming interfaces, and libraries. Evaluate how a program is written in terms of program style, expected behavior for specific inputs, correctness of program components, and description of functionalities. Develop a program using tools adapted to current industry practices : version control, project hosting, and deployment services.
- **Fundamental data structures** Write programs that use data structures (built-in, library, and programmer-defined) : strings, lists, and maps. Analyze the performance of different data structure implementations. Choose appropriate data structures to model a given problem. Explain the rationale behind the chosen data structures.
- **Principles and development of algorithms** Describe why and how algorithms solve computational problems. Create algorithms to solve a computational problem. Explain how programs implement algorithms in terms of instruction processing, program execution, and ongoing processes. Apply appropriate mathematical concepts to programming : expressions, abstract data types, recurrence relations, and formal reasoning on algorithm efficiency and correctness. Empirically evaluate the efficiency of an algorithm.

BIBLIOGRAPHY

- Open data structures, Pat Morin (opensource <https://opendatastructures.org/>)
- Introduction to algorithms, Erik Demaine, Jason Ku, Justin Solomon <https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/>
- The C programming language. Kernighan, Brian W.; Ritchie, Dennis M. Prentice Hall, (2016)
- Data Structures and Program Design In C. Robert L. Kruse, Bruce P. Leung, Clovis L. Tondo, (1996, Pearson)

REQUIREMENTS

C and C++ programming from CPI (syntax, pointers, organization of C program files...).