linfo1121

2020

Due to the COVID-19 crisis, the information below is subject to change, in particular that concerning the teaching mode (presential, distance or in a comodal or hybrid format).

5 credits	30.0 h + 30.0 h	Q1

Teacher(s)	Derval Guillaume (compensates Schaus Pierre) ;Schaus Pierre ;
Language :	French
Place of the course	Louvain-la-Neuve
Prerequisites	*The prerequisite(s) for this Teaching Unit (Unité d'enseignement – UE) for the programmes/courses that offer this Teaching Unit are specified at the end of this sheet.*
Main themes	• Complexity measures of an algorithm and complexity analysis methods. • Dichotomic sorting and search algorithms. • Basic data structures (lists, trees, binary search trees): study of their abstract properties, their concrete representations, their application and the main algorithms that manipulate them. • Advanced data structures (union-find, hash tables, heaps, balanced binary trees, graph representation and manipulation, textual data processing, dictionaries).
Aims	Given the learning outcomes of the "Bachelor in Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • AA1.1, AA1.2 • AA2.4, AA2.5, AA2.7 • AA3.2 • AA4.3 Given the learning outcomes of the "Bachelor in Computer science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • S1.I1, S1.I3 • S2.2, S2.3, S2.4 • S4.3 • S5.4 • S6.1, S6.3 1 Students who have successfully completed this course will be able to : • make an informed choice on the use of the main data structures used to represent collections, • make good use of existing algorithms to manipulate these data structures and analyze their performance, • design and implement variants of the algorithms studied, • test algorithms and data structures, • make good use of algorithms and data structures documented in an API • abstract, model and implement effective solutions to algorithmic puzzle problems. **Students will have developed methodological and operational skills. In particular, they will have developed their ability to:** • analyze critically a problem, • to test and debug algorithmic programs, • effectively implement short but non-trivial algorithms. learn for themselves in a reference book and in the complementary technical documentation - - - - *The contribution of this Teaching Unit to the development and command of the skills and learning outcomes of the programme(s) can be accessed at the end of this sheet, in the section entitled "Programmes/courses offering this Teaching Unit".*

Evaluation methods	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** • The participation to the tutored sessions accounts for 20% of the final note. This total includes: • The **active** participation to the tutored session by the student, evaluated by the tutor; • The exercises made on INGInious (they should be submitted before the deadlines) The participation is also taken into account during the August exam session. • The exam amounts for 80% of the final grade. • A mid-term is organized on INGInious around week 7. The mid-term accounts for 10% of the total of the exam (8% of the overall total) only if it increases the overall grade of the student. The grade of the mid-term is kept during the August exam session. The content of the mid-term is based on modules 1 to 3.
Teaching methods	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** The active teaching method followed in this course is inspired from the *Problem Solving Approach* and is a flipped classroom. This method is based on several *phases* of work, some supervised by *tutors*. In addition to tutored sessions, an essential component of this pedagogical approach is to promote each *student to learn by himself*. The success of the learning process presupposes a ***significant involvement*** of each student. The role of *group work* is mainly to discuss the concepts studied and, secondarily, to organize the work of each. **Learning itself remains the responsibility of each student.** The work is organized into modules that each group of students must perform with **strict deadline** (typically 2 weeks per module, with 2 deadlines per module). These missions include *questions* for which each group must make the best possible answers and *programming problems*, for which each student must produce Java programs individually.
Content	• Computational complexity, • Trees, binary search trees, • Balanced trees, • Dictionaries and hash tables, • Priority queues and heaps • Graphs, • Text processing (pattern matching, compression algorithms)
Inline resources	https://moodleucl.uclouvain.be/course/view.php?id=7682 + questions on the course website (reachable from Moodle).
Bibliography	Livre obligatoire: Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional. ISBN-13: 978-0321573513 ISBN-10: 032157351X Et plus généralement les documents (énoncés des missions, conseils pour l'examen, ...) disponibles sur : http://moodleucl.uclouvain.be/course/view.php?id=7682
Other infos	Background: • master an object-oriented programming language (p.e. Java) • know an use correctly basic data structures (stacks, queues, lists, etc..) • have basic knowledge of recursion and computational complexity. This background is part of the content of LEPL1401 and LEPL1402.
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)

Program title	Acronym	Credits	Prerequisite	Aims
Bachelor in Computer Science	SINF1BA	5	LEPL1402	🔍
Specialization track in Computer Science	FILINFO	5		🔍
Bachelor in Mathematics	MATH1BA	5		🔍
Approfondissement en statistique et sciences des données	APPSTAT	5		🔍