UCLouvain

lsinf1121

2017

Algorithmics and data structures

5 credits	30.0 h + 30.0 h	Q1

Teacher(s)	Schaus Pierre ;
Language :	French
Place of the course	Louvain-la-Neuve
Prerequisites	Within SINF1BA : LSINF1101 and LSINF1103 Within FSA1BA : LFSAB1401, LFSAB1101, LFSAB1102, LFSAB1201, LFSAB1202, FSAB1301 *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	• Computational complexity • Specifications and object-oriented design • Basic data structures (lists, trees, binary search trees): study of their abstract properties, practical representations, concrete applications and associated algorithms • Advanced data structures and algorithms: hash tables, heaps, balanced search trees, text processing techniques, dictionaries, graph representation and processing
Aims	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: • AA1.1, AA1.2 • AA2.4, AA2.5, AA2.7 • AA3.2 • AA4.3 Given the learning outcomes of the "Bachelor in Engineering" 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 completing successfully this course will be able to • make a reasoned choice between the main data structures used to represent collections, • make good use of existing algorithms for manipulating these data structures and analyze their performance, • apply the principles of object-oriented programming such as genericity, abstraction, composition and reuse, • design and implement variants of the studied algorithms in Java programs of high quality. Students will have developed skills and operational methodology. In particular, they have developed their ability to: • critically analyze a problem, • learn by themselves in a reference book and in other technical documentation, • work effectively in groups to analyze a problem, design, implementation and documentation of programs, • balance the individual and group work, • manage the learning curve and produce a satisfactory solution to the problems within time constraints. - - - - *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	**A note of PARTICIPATION on different missions and programming problems (Inginious , etc.) accounts for 15% of the final grade in January. Only the final exam is taken into consideration for the final grade of second session (the participation is no longer considered) .**

Teaching methods	The active teaching method followed in this course is based on a Problem Solving Approach. 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 missions that each group of students must perform with **strict deadline** (typically 2 weeks per mission). These missions include questions for which each group must make the best possible answers and programming problems, for which the group must produce Java programs. The missions are intended primarily to create a context and motivation for learning new concepts and for developping methodological skills. Reach the end of a mission is not an end in itself. It is important to keep in mind that each mission is primarily an opportunity to develop learning goals explicit in the statement of each mission.
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
Bibliography	Textbook 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.
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)

Program title	Acronym	Credits	Prerequisite	Aims
Master [120] in Linguistics	LING2M	5	LSINF1103	🔍
Master [120] in Mathematical Engineering	MAP2M	5		🔍
Bachelor in Mathematics	MATH1BA	5		🔍
Bachelor in Computer Science	SINF1BA	5	LSINF1101 AND LSINF1102 AND LSINF1103	🔍
Master [120] in data Science: Statistic	DATS2M	5		🔍
Minor in Statistics and data sciences	LSTAT100I	5		🔍
Minor in Engineering Sciences: Computer Sciences	LSINF100I	5		🔍
Minor in Computer Sciences	LINFO100I	5	LSINF1101	🔍
Additionnal module in Statistics and data science	LSTAT100P	5		🔍