UCL Université catholique de Louvain

LSINF1121

2010-2011

Algorithmics and data structures

5.0 credits	30.0 h + 30.0 h	1q

Teacher(s) :	Dupont Pierre ;
Language :	Français
Place of the course	Louvain-la-Neuve
Inline resources:	http://www.icampus.ucl.ac.be/claroline/course/index.php?cid=SINF1121
Prerequisites :	-- Master of a object-oriented programming language (p.e. Java) -- Knowledge of basic data structures (stacks, queues, lists, etc..) -- Notions of recursion and computational complexity.
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 :	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 :	The mark for this course is based on a dual evaluation: -- The effective participation of each student during the tutored sessions and the timeliness (20%, evaluated by each tutor and teacher). -- Note the final exam (80%, assessed by the teacher). In second session, only the score for the final examination during the second session will be taken into account (for 100% of the final mark). The subjects required for the final exam consists of the contents of all documents available on iCampus for this course, information provided orally at weekly tutored sessions, and all chapters of the reference book mentioned in the statement of at least one of the missions. The reference book is the only resource whose consultation is permitted during the final exam.
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)
Bibliography :	Mandatory book: -- Goodrich and Tamassia, Data structures & mp; algorithms in Java, 5th edition, John Wiley & mp; suns, 2011, ISBN: 978-0-470-39880-7 Recommended books: -- Cormen T.H. et al. , "Introduction to Algorithms, Second Edition" , MIT Press, 2001. -- Brassard G. & mp; Bratley P., "Fundamentals of Algorithms" , Prentice Hall, 1996.
Cycle and year of study :	> Master [120] in Linguistics > Master [120] in Information and Communication Science and Technology > Master [120] in Mathematical Engineering > Master [120] in Statistics: General > Bachelor in Engineering > Bachelor in Computer Science > Bachelor in Mathematics > Preparatory year for Master in Computer science > Bachelor in Engineering : Architecture > Bachelor in Economics and Management
Faculty or entity in charge:	INFO