lepl1402

2019

In view of the health context linked to the spread of the coronavirus, the methods of organisation and evaluation of the learning units could be adapted in different situations; these possible new methods have been - or will be - communicated by the teachers to the students.

5 credits	30.0 h + 30.0 h	Q1

Teacher(s)	Sadre Ramin ;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.*
Aims	*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.** An optional quiz takes place in the middle of the quad counting for two points only if the score is higher than the exam score: final note = max (quizz_on_2 + examen_on_18, examen_on_20). The examination at the end of the semester aims to verify not only the knowledge of the subject, but also the ability to apply the knowledge acquired to write programs. The exam and the quiz will use the Inginious evaluation system.
Teaching methods	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** Lectures + Exercices on Inginious + lab sessions with TA
Content	This teaching unit focuses on: - Introduction to Java: compilation, byte-code, virtual machine, primitive type, strings, tables - Abstract data types; - Linear and tree structures, and their applications; - recursive solution formulation and recursive algorithms; - reasoning technique: preconditions, postconditions, invariants - Notions of computational complexity and analysis of the temporal and spatial complexity of an algorithm; - Functional programming and higher order programming - Object-oriented modeling (inheritance, composition, reuse, polymorphism, class invariant); - Introduction to design patterns; - Program testing and validation methods; - Introduction to parallelization: notion of thread and synchronization mechanisms. Students who have successfully completed this course will be able to • to design Java programs • analyze programs according to their performance • to prove correctness of programs using invariants • apply the principles of object-oriented programming such as genericity, abstraction, composition and reuse • design and implement variants of the algorithms studied in high quality Java programs. • design and manipulate simple linear and tree and recursive structures • design tests for programs • design functional programming approaches to solve small algorithmic problems • use design patterns • design simple parallel programs with synchronization mechanisms
Inline resources	https://lepl1402.readthedocs.io/ (pour les liens vers les slides et contenu du cours) https://moodleucl.uclouvain.be/course/view.php?id=12884 Moodle (pour la communication avec les étudiants) https://inginious.info.ucl.ac.be/course/LEPL1402 pour les exercices
Other infos	Background: LEPL1401 or a similar course
Faculty or entity in charge	BTCI

Programmes containing this learning unit (UE)

Program title	Acronym	Credits	Prerequisite	Aims
Minor in Computer Sciences	LINFO100I	5		🔍
Minor in Information and Communication Studies and Technologies	LSTIC100I	5		🔍
Master [120] in Data Science : Statistic	DATS2M	5		🔍
Bachelor in Mathematics	MATH1BA	5	LINFO1101	🔍
Master [120] in Linguistics	LING2M	5		🔍
Bachelor in Engineering	FSA1BA	5	LEPL1401	🔍
Bachelor in Computer Science	SINF1BA	5	LINFO1101	🔍