

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)	Sadre Ramin ;Schaus Pierre ;SOMEBODY ;
Language :	French
Place of the course	Louvain-la-Neuve
Prerequisites	<i>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.</i>
Aims	<i>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".</i>
Evaluation methods	<p>Due to the COVID-19 crisis, the information in this section is particularly likely to change.</p> <p>An optional quiz (in two parts) is held, counting towards the Continuing Assessment only if the score is higher than the exam score.</p> <p>The purpose of the exam in January is to test not only knowledge of the subject, but also the ability to apply the knowledge acquired to write programs. The exam and quiz will use the Ingenious grading system.</p> <p>In case of detection of plagiarism confirmed by a plagiarism detection tool the course teachers reserve the right to let the student take an oral test.</p>
Teaching methods	<p>Due to the COVID-19 crisis, the information in this section is particularly likely to change.</p> <p>Lectures + Exercices on Ingenious + lab sessions with TA</p>
Content	<p>This teaching unit focuses on:</p> <ul style="list-style-type: none"> - 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. <p>Students who have successfully completed this course will be able to</p> <ul style="list-style-type: none"> • 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	<p>https://lepl1402.readthedocs.io/ (pour les liens vers les slides et contenu du cours)</p> <p>https://moodleucl.uclouvain.be/course/view.php?id=12884 Moodle (pour la communication avec les étudiants)</p> <p>https://ingenious.info.ucl.ac.be/course/LEPL1402 pour les exercices</p>
Other infos	Background: LEPL1401 or a similar course

Faculty or entity in charge	BTCI
-----------------------------	------

Force majeure

Evaluation methods	<p>An optional quiz (in two parts) is held, counting towards the Continuing Assessment only if the score is higher than the exam score.</p> <p>The purpose of the exam in January is to test not only knowledge of the subject, but also the ability to apply the knowledge acquired to write programs. The exam and quiz will use the Ingenious grading system.</p> <p>In case of detection of plagiarism confirmed by a plagiarism detection tool the course teachers reserve the right to let the student take an oral test.</p>
--------------------	---

Programmes containing this learning unit (UE)				
Program title	Acronym	Credits	Prerequisite	Aims
Master [120] in Data Science : Statistic	DATS2M	5		
Master [120] in Linguistics	LING2M	5		
Minor in Computer Sciences	MINSINF	5		
Minor in Statistics, Actuarial Sciences and Data Sciences	MINSTAT	5		
Bachelor in Computer Science	SINF1BA	5	LINFO1101	
Bachelor in Mathematics	MATH1BA	5		
Bachelor in Engineering	FSA1BA	5	LEPL1401	
Approfondissement en statistique et sciences des données	APPSTAT	5		
Minor in numerical technologies and society	MINSTIC	5		