linfo2132

2021

Languages and translators

6.00 credits	30.0 h + 30.0 h	Q2

Teacher(s)	Laurent Nicolas ;
Language :	English
Place of the course	Louvain-la-Neuve
Main themes	• Methods to analyze context-free languages, upstream and downstream methods • Generators of lexical analyzers and parsers • Statistical semantics and attributed grammars • Methods to translate a source code in a target code, and generation of target code • Machine virtuelle et byte-code (JVM) • Garbage Collection et gestion mémoire • Domain Specific Languages (DSL)
Learning outcomes	**At the end of this learning unit, the student is able to :** Given the learning outcomes of the "Master in Computer Science and Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • INFO1.1-3 • INFO2.2-4 • INFO5.2, INFO5.4, INFO5.5 • INFO6.1, INFO6.4 Given the learning outcomes of the "Master [120] in Computer Science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: • SINF1.M2 • SINF2.2-4 • SINF5.2, SINF5.4, SINF5.5 • SINF6.1, SINF6.4 Given the learning outcomes of the "Master [60] in Computer Science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: 1 • 1SINF1.M2 • 1SINF2.2-4 • 1SINF5.2, 1SINF5.4, 1SINF5.5 • 1SINF6.1, 1SINF6.4 Students completing successfully this course will be able to • explain in a practical way the structure of compilers dealing with algorithmic languages • design and implement a compiler for a practical language which solves a interesting problem • show the interest of compiling techniques in problem resolving Students will have developed skills and operational methodology. In particular, they have developed their ability to • treat rigorously a problem, justifying and validating each step of a project to be able to rely on it to implement the following one • explain in practical terms how a source code (Java) is finally translated into byte-code. • explain the enforcement mechanisms byte code by JVM • explain memory management during the execution of a program • explain how garbage collection mechanisms
Evaluation methods	The project (done in groups of two) accounts for two third of the course's grade. The project consists in extending the implementation of a basic programming language with new paradigms. A written exam accounts for the final third. In case of a second session, the students are free to resubmit the project, re-present the exam, or both.

Teaching methods	The course consists in a series of pre-recorded video lectures, as well as weekly or bi-weekly consolidation sessions where students can ask their questions. There will also be a couple of lab sessions to best prepare the students for the project. During the semester, students will have to complete the project, which consists of extending a basic programming language interpreter/compiler with new paradigms.
Content	The course presents the theory and practice of programming language implementation, as well as compiler architecture. We will review the standard components of a compiler, from front-end (parsing, lexical analysis) to back-end (machine code generation, or interpreters), also touching on static semantics and type systems. Ultimately, students should be able to understand the ins and outs of the various programming language implementation techniques in use today. During the course, the students will implement their own programming language.
Inline resources	http://moodleucl.uclouvain.be/course/view.php?id=5423
Bibliography	Ouvrage(s) recommandé(s) : • Crafting Interpreters, Bob Nystrom (https://craftinginterpreters.com/) • How To Create Pragmatic Lightweight Languages, Federico Tomassetti • Introduction to Compiler Construction in a Java World, Bill Campbell, Swamilyer, Bahar Akbal-Deliba • Modern Compiler Implementation in Jaav, Andrew W. Appel
Other infos	Background : • LINGI1122 : Program design • LSINF1121 : High-level programming language, algorithmics and data structures • LINGI1101 : Logic and discrete structures
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)

Program title	Acronym	Credits	Prerequisite	Learning outcomes
Master [120] in Computer Science and Engineering	INFO2M	6		🔍
Master [60] in Computer Science	SINF2M1	6		🔍
Master [120] in Computer Science	SINF2M	6		🔍