

6.0 credits

30.0 h + 30.0 h

2q

Teacher(s) :	Schaus Pierre ;
Language :	Anglais
Place of the course	Louvain-la-Neuve
Inline resources:	 > http://icampus.uclouvain.be/claroline/course/index.php?cid=ing2132
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)
Aims :	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: -- 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 --

	<p>treat rigorously a problem, justifying and validating each step of a project to be able to rely on it to implement the following one</p> <p>--</p> <p>explain in practical terms how a source code (Java) is finally translated into byte-code.</p> <p>--</p> <p>explain the enforcement mechanisms byte code by JVM</p> <p>--</p> <p>explain memory management during the execution of a program</p> <p>--</p> <p>explain how garbage collection mechanisms</p> <p><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></p>
Evaluation methods :	<p>--</p> <p>oral exam</p> <p>--</p> <p>3 assignments per groups of 2</p> <p>--</p> <p>1 project on DSL</p> <p>The project and assignments can not be represented in second session.</p>
Teaching methods :	<p>--</p> <p>Lectures</p> <p>--</p> <p>Practical sessions</p> <p>--</p> <p>Project (design and implementation of a compiler)</p>
Content :	<p>--</p> <p>Introduction</p> <p>--</p> <p>Formal Languages</p> <p>--</p> <p>Chomsky's Formal Grammars</p> <p>--</p> <p>Languages and Regular Expressions, Automata finite set of states</p> <p>--</p> <p>Lexical Analysis</p> <p>--</p> <p>Top-down Parsing: general method</p> <p>--</p> <p>Top-down Parsing based on grammars LL (1)</p> <p>--</p> <p>Bottom-up Parsing and relations of priorities</p> <p>--</p> <p>Scala and specific language to conduct DSL</p> <p>--</p> <p>Some functional programming concepts relating to DSL (monads, etc.)</p>
Bibliography :	<p>Course material available online (course website)</p> <p>Recommended readings</p> <p>Introduction to Compiler Construction in a Java World, Bill Campbell, Swami Iyer, Bahar Akbal-Deliba' http://www.cs.umb.edu/~/</p> <p>Scala for the Impatient, Cay Horstmann, Addison-Wesley 2012</p> <p>Programming in Scala: A Comprehensive Step-by-Step Guide, 2nd Edition, Martin Odersky , Lex Spoon, Bill Venners</p>
Other infos :	<p>Background :</p> <p>--</p> <p>LINGI1122 : Rigorous Methods okprogram design</p> <p>--</p> <p>LSINF1121 : High-level programming language, algorithmics and data structures</p> <p>--</p> <p>LINGI1101 : Logic and discrete structures</p>
Cycle and year of study :	<p>> Master [120] in Computer Science</p> <p>> Master [120] in Biomedical Engineering</p> <p>> Master [60] in Computer Science</p> <p>> Master [120] in Computer Science and Engineering</p>
Faculty or entity in charge:	INFO