

6.0 crédits

30.0 h + 30.0 h

2q

Enseignants:	Schaus Pierre ;
Langue d'enseignement:	Anglais
Lieu du cours	Louvain-la-Neuve
Ressources en ligne:	> <a href="http://icampus.uclouvain.be/claroline/course/index.php?cid=ingi2132">http://icampus.uclouvain.be/claroline/course/index.php?cid=ingi2132</a>
Préalables :	-- Méthodes rigoureuses de conception de programmes (p.e. INGI1122) -- langage de programmation de haut niveau, algorithmique et structures de données (p.e. SINF1121) -- Logique et structures discrètes (p.e. INGI1101)
Thèmes abordés :	-- Méthodes d'analyse de langages "context-free", méthodes ascendantes et méthodes descendantes. Applications -- Générateurs d'analyseurs lexicaux et syntaxiques -- Sémantique statique et grammaires attribuées -- Méthode de définition de traduction du code source en code cible et génération du code cible -- Machine virtuelle et byte-code (JVM) -- Garbage Collection et gestion mémoire -- Domain Specific Languages (DSL)
Acquis d'apprentissage	Les étudiants ayant suivi avec fruit ce cours seront capables de -- expliquer de façon pratique la structure des compilateurs pour des langages algorithmiques -- concevoir et implémenter un compilateur pour un langage pratique qui résout un problème à intérêt intrinsèque -- montrer l'intérêt des techniques de compilation dans la résolution de problèmes dans d'autres domaines Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils ont développé leur capacité à -- traiter avec rigueur une problématique en justifiant et validant chaque étape d'un projet pour pouvoir s'appuyer sur celle-ci pour mettre en oeuvre la suivante -- expliquer de façon pratique comment un code-source (Java) est finalement traduit en byte-code. -- expliquer les mécanismes d'exécution du byte-code par une JVM -- expliquer la gestion mémoire lors de l'exécution d'un programme -- expliquer le fonctionnement des mécanismes de garbage collection <i>La contribution de cette UE au développement et à la maîtrise des compétences et acquis du (des) programme(s) est accessible à la fin de cette fiche, dans la partie « Programmes/formations proposant cette unité d'enseignement (UE) ».</i>
Modes d'évaluation des acquis des étudiants :	-- examen oral -- 3 assignments par groupe de 2 -- 1 projet relatif aux DSL Le projet et les assignments ne peuvent pas être repassés en deuxième session.

<p>Méthodes d'enseignement :</p>	<p>-- Cours magistral -- Séances d'exercices -- Projet (conception et implémentation d'un compilateur)</p>
<p>Contenu :</p>	<p>-- Introduction -- Langages formels -- Grammaires Formelles de Chomsky -- Langages et Expressions Régulières, Automates à Ensemble Fini d'États -- Analyse Lexicale -- Analyse syntaxique descendante : méthode générale -- Analyse syntaxique descendante basée sur les grammaires LL(1) -- Analyse syntaxique ascendante et relations de priorités -- Scala et les spécificités du langage pour conduire des DSL -- Quelques notions de programmations fonctionnelles relatives aux DSL (monads, etc.)</p>
<p>Bibliographie :</p>	<p>Supports du cours en ligne (site web du cours) Ouvrage(s) recommandé(s) : Introduction to Compiler Construction in a Java World, Bill Campbell, Swami Iyer, Bahar Akbal-Delibai' <a href="http://www.cs.umb.edu/~j--/">http://www.cs.umb.edu/~j--/</a> Scala for the Impatient, Cay Horstmann, Addison-Wesley 2012 Programming in Scala: A Comprehensive Step-by-Step Guide, 2nd Edition, Martin Odersky , Lex Spoon, Bill Venners</p>
<p>Cycle et année d'étude: :</p>	<p>&gt; <a href="#">Master [120] : ingénieur civil en informatique</a> &gt; <a href="#">Master [120] en sciences informatiques</a> &gt; <a href="#">Master [60] en sciences informatiques</a> &gt; <a href="#">Master [120] : ingénieur civil biomédical</a></p>
<p>Faculté ou entité en charge:</p>	<p>INFO</p>