




| | | |
|-----------|-----------------|----|
| 6 crédits | 30.0 h + 30.0 h | Q2 |
|-----------|-----------------|----|

| | |
|---|---|
| Enseignants | Laurent Nicolas ; |
| Langue d'enseignement | Anglais |
| Lieu du cours | Louvain-la-Neuve |
| Thèmes abordés | <ul style="list-style-type: none"> • 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 | <p>Eu égard au référentiel AA du programme « Master ingénieur civil en informatique », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • INFO1.1-3 • INFO2.2-4 • INFO5.2, INFO5.4, INFO5.5 • INFO6.1, INFO6.4 <p>Eu égard au référentiel AA du programme « Master [120] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • SINF1.M2 • SINF2.2-4 • SINF5.2, SINF5.4, SINF5.5 • SINF6.1, SINF6.4 <p>Eu égard au référentiel AA du programme « Master [60] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <p>1</p> <ul style="list-style-type: none"> • 1SINF1.M2 • 1SINF2.2-4 • 1SINF5.2, 1SINF5.4, 1SINF5.5 • 1SINF6.1, 1SINF6.4 <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> • 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 <p>Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils ont développé leur capacité à</p> <ul style="list-style-type: none"> • 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 <p>-----</p> <p><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></p> |
| Modes d'évaluation des acquis des étudiants | <p>En raison de la crise du COVID-19, les informations de cette rubrique sont particulièrement susceptibles d'être modifiées.</p> <p>Le projet (fait par groupe de deux) compte pour deux tiers de la note du cours. Il consiste à étendre l'implémentation d'un langage de programmation basique avec de nouveaux paradigmes. Un examen écrit compte pour le tiers restant.</p> <p>En cas de seconde session, les étudiants sont libres de représenter le projet, l'examen, ou les deux.</p> |

| | |
|------------------------------|--|
| Méthodes d'enseignement | <p>En raison de la crise du COVID-19, les informations de cette rubrique sont particulièrement susceptibles d'être modifiées.</p> <p>Le cours consiste d'une série de présentations pré-enregistrées, ainsi que de sessions de consolidation hebdomadaire ou bimensuelles où les étudiants ont l'occasion de poser leurs questions. Il y aura également des séances de travaux pratique afin de préparer les étudiants au projet. Au cours du quadrimestre, les étudiants devront réaliser le projet, qui consiste à étendre un interpréteur/compilateur pour un langage de programmation basique avec de nouveaux paradigmes.</p> |
| Contenu | <p>Le cours présente la théorie et la pratique de l'implémentation des langages de programmation, et de l'architecture des compilateurs. Nous balayerons les composants standards d'un compilateur, du front-end (analyse lexicale et syntaxique) au back-end (émission de code machine, ou interpréteur) en passant par l'analyse statique et les systèmes de types. A terme, les étudiants seront capables de comprendre les tenants et les aboutissants des différentes méthodes d'implémentation de langage en usage aujourd'hui.</p> <p>Au cours du quadrimestre, les étudiants seront amenés à implémenter leur propre langage de programmation.</p> |
| Ressources en ligne | <p>http://moodleucl.uclouvain.be/course/view.php?id=5423</p> |
| Bibliographie | <p>Ouvrage(s) recommandé(s) :</p> <ul style="list-style-type: none"> • 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 |
| Autres infos | <p>Préalables :</p> <ul style="list-style-type: none"> • LINGI1122 : Méthodes rigoureuses de conception de programmes • LSINF1121 : langage de programmation de haut niveau, algorithmique et structures de données • LINGI1101 : Logique et structures discrètes |
| Faculté ou entité en charge: | <p>INFO</p> |

| Programmes / formations proposant cette unité d'enseignement (UE) | | | | |
|--|---------|---------|-----------|---|
| Intitulé du programme | Sigle | Crédits | Prérequis | Acquis d'apprentissage |
| Master [120] : ingénieur civil en informatique | INFO2M | 6 | |  |
| Master [60] en sciences informatiques | SINF2M1 | 6 | |  |
| Master [120] en sciences informatiques | SINF2M | 6 | |  |