




Au vu du contexte sanitaire lié à la propagation du coronavirus, les modalités d'organisation et d'évaluation des unités d'enseignement ont pu, dans différentes situations, être adaptées ; ces éventuelles nouvelles modalités ont été -ou seront- communiquées par les enseignant-es aux étudiant-es.

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

| | |
|------------------------|---|
| Enseignants | Schaus Pierre ; |
| 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> <ul style="list-style-type: none"> • examen écrit • 3 assignements par groupe de 2 • 1 projet relatif aux DSL <p>Le projet et les assignements valent pour 40% de la note et ne peuvent pas être repassés en deuxième session.</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> <ul style="list-style-type: none"> • Cours magistral • Séances d'exercices • Projet (conception et implémentation d'un compilateur) |
| Contenu | <ul style="list-style-type: none"> • 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) • Scala et les spécificités du langage pour conduire des DSL • Quelques notions de programmations fonctionnelles relatives aux DSL (monads, etc.) |
| Ressources en ligne | http://moodleucl.uclouvain.be/course/view.php?id=5423 |
| Bibliographie | <p>Supports du cours en ligne (site web du cours) Ouvrage(s) recommandé(s) :</p> <ul style="list-style-type: none"> • Introduction to Compiler Construction in a Java World, Bill Campbell, Swami Iyer, Bahar Akbal-Deliba http://www.cs.umb.edu/~/ • 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 |
| 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: | INFO |

| 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 | |  |