



6.00 crédits	30.0 h + 30.0 h	Q2
--------------	-----------------	----

Enseignants	Sadre Ramin ;
Langue d'enseignement	Anglais > Facilités pour suivre le cours en français
Lieu du cours	Louvain-la-Neuve
Préalables	Requis : concepts, paradigmes et sémantique des langages de programmation tels que visés par le cours LINFO1104 Requis : notions approfondies d'algorithmique et de programmation orientée objets telles que visées par le cours LEPL1402 Souhaitable : principes des systèmes informatiques, tels que visés par le cours LINFO1252
Thèmes abordés	<ul style="list-style-type: none"> <li>• Méthodes d'analyse de langages "context-free", méthodes ascendantes et méthodes descendantes. Applications</li> <li>• Générateurs d'analyseurs lexicaux et syntaxiques</li> <li>• Sémantique statique et grammaires attribuées</li> <li>• Méthode de définition de traduction du code source en code cible et génération du code cible</li> <li>• Machine virtuelle et byte-code (JVM)</li> <li>• Garbage Collection et gestion mémoire</li> <li>• Domain Specific Languages (DSL)</li> </ul>
Acquis d'apprentissage	<p><b>A la fin de cette unité d'enseignement, l'étudiant est capable de :</b></p> <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"> <li>• INFO1.1-3</li> <li>• INFO2.2-4</li> <li>• INFO5.4, INFO5.5</li> <li>• INFO6.4</li> </ul> <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"> <li>• SINF1.M2</li> <li>• SINF2.2-4, SINF2.5</li> <li>• SINF5.4, SINF5.5</li> <li>• SINF6.4</li> </ul> <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> <ul style="list-style-type: none"> <li>• 1SINF1.M2</li> <li>• 1SINF2.2-4, 1SINF2.5</li> <li>• 1SINF5.4, 1SINF5.5</li> <li>• 1SINF6.4</li> </ul> <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> <li>• expliquer de façon pratique la structure des compilateurs pour des langages algorithmiques;</li> <li>• concevoir et implémenter un compilateur pour un langage pratique qui résout un problème à intérêt intrinsèque;</li> <li>• montrer l'intérêt des techniques de compilation dans la résolution de problèmes dans d'autres domaines.</li> </ul> <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"> <li>• 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;</li> <li>• expliquer de façon pratique comment un code-source (Java) est finalement traduit en byte-code;</li> <li>• expliquer les mécanismes d'exécution du byte-code par une JVM;</li> <li>• expliquer la gestion mémoire lors de l'exécution d'un programme;</li> <li>• expliquer le fonctionnement des mécanismes de garbage collection.</li> </ul>

Modes d'évaluation des acquis des étudiants	<p>Session de juin : L'évaluation se compose de deux composantes: Le projet (réalisé en groupe) compte pour 60% de la note finale du cours. Un examen écrit compte pour 40%.</p> <p>Session d'août : Si l'étudiant.e n'a pas réussi le cours lors de la première session (c'est-à-dire qu'il/elle n'a pas obtenu au moins 10/20 pour la note finale), il/elle est autorisée à refaire les composantes (projet ou examen ou les deux) de l'évaluation pour lesquelles il/elle n'a pas obtenu au moins 50% des points respectifs. Il/elle conservera les points de la composante qu'il/elle a réussie (le cas échéant). Les mêmes pondérations que lors de la session de juin sont appliquées pour le calcul de la note finale.</p> <p>Concernant l'utilisation de l'IA générative : les étudiants sont autorisés à utiliser des outils d'IA générative pour soutenir leur travail de projet, par exemple pour comparer des approches de solution, trouver des erreurs dans le code ou améliorer la formulation des rapports. Cependant, les soumissions, qu'il s'agisse de code source, de rapports ou d'autres formes, doivent être principalement votre propre travail. Si des outils d'IA générative ont été utilisés, leur nature et leur portée doivent être décrites en détail dans le travail remis (voir les instructions du projet concerné) et seule la partie originale du travail de l'étudiant sera évaluée. Cela s'applique également lorsque l'on s'inspire du travail de tiers, qu'il provienne d'Internet ou d'une autre personne. Le non-respect de ces directives peut entraîner une réduction des notes ou d'autres sanctions académiques.</p> <p>Le professeur peut demander à un.e étudiant.e de passer un examen oral supplémentaire en complément de l'examen et/ou des activités du projet, dans des cas incluant, mais non limités à, des problèmes techniques, ou des suspicions d'irrégularités.</p>
Méthodes d'enseignement	<p>Le cours consiste en une série de vidéos préenregistrées, de leçons traditionnelles, de séances de questions-réponses et de séances de laboratoire.</p> <p>Un projet sera réalisé avec plusieurs échéances réparties sur le quadrimestre.</p>
Contenu	<p>Le cours présente la théorie et la pratique de l'implémentation des langages de programmation, ainsi que l'architecture des compilateurs. Nous passerons en revue les composants standards d'un compilateur, du front-end (parsing, analyse lexicale) au back-end (génération de code).</p> <p>Nous aborderons entre autres :</p> <ul style="list-style-type: none"> <li>• Les scanners/lexers et les expressions régulières</li> <li>• L'analyse syntaxique et les grammaires non contextuelle</li> <li>• Type checking</li> <li>• L'allocation de registres et la génération de code</li> </ul> <p>Pendant le cours, les étudiants implémenteront un compilateur pour un nouveau langage de programmation.</p>
Ressources en ligne	Teams et/ou Moodle
Autres infos	<p>Les étudiants doivent avoir de solides connaissances en programmation dans un langage orienté objet compilé, en algorithmes et structures de données, ainsi qu'en architecture informatique, telles qu'enseignées par exemple dans les cours suivants :</p> <ul style="list-style-type: none"> <li>• LEPL1402 et/ou LEPL1509 : Programmation en Java</li> <li>• LINFO1252 et/ou LINFO2241 : Systèmes informatiques/computer architecture</li> <li>• LINFO1121 : Algorithmes et structures de données</li> </ul> <p>De plus, des notations formelles seront utilisées pour décrire les propriétés des langages de programmation.</p>
Faculté ou entité en charge:	INFO

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