





5.00 crédits	30.0 h + 30.0 h	Q1
--------------	-----------------	----

Enseignants	Schaus Pierre ;
Langue d'enseignement	Français
Lieu du cours	Louvain-la-Neuve
Préalables	<p>Ce cours suppose acquises la maîtrise de la programmation et de la conception de programmes dans un langage orienté-objet tel que Java, la connaissance de structures de données élémentaires et des notions de récursion et de complexité calculatoire telles que visées par le cours LEPL1402.</p> <p><i>Les prérequis de cette unité d'enseignement (UE) sont précisés à la fin de cette fiche, en regard des programmes/formations qui proposent cette UE.</i></p> <p><i>Le(s) prérequis de cette Unité d'enseignement (UE) sont précisés à la fin de cette fiche, en regard des programmes/formations qui proposent cette UE.</i></p>
Thèmes abordés	<ul style="list-style-type: none"> • Mesures de complexité d'un algorithme et méthodes d'analyse de complexité. • Algorithmes de tris et recherche dichotomiques. • Structures de données de base (listes, arbres, arbres binaires de recherche) : étude de leurs propriétés abstraites, de leurs représentations concrètes, de leur application et des principaux algorithmes qui les manipulent. • Structures de données avancées (union-find, tables de hachage, tas, arbres binaires équilibrés, représentation et manipulation de graphes, traitement de données textuelles, dictionnaires).
Acquis d'apprentissage	<p>A la fin de cette unité d'enseignement, l'étudiant est capable de :</p> <p>Eu égard au référentiel AA du programme « Bachelier ingénieur civil », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • AA1.1. Appliquer les concepts, lois, raisonnements à une problématique disciplinaire de complexité cadrée. • AA1.2. Décrire des outils de modélisation et de calcul adéquats pour résoudre une problématique disciplinaire cadrée. • AA2.5. Implémenter et tester une solution sous la forme d'une maquette, d'un prototype et/ou d'un modèle numérique. • AA2.8. Formuler des recommandations pour améliorer la solution étudiée, le système analysé dans toute sa complexité. • AA3.2. S'autoévaluer de manière critique, continue et collaborative en vue de fonctionner efficacement en équipe. <p>Eu égard au référentiel AA du programme « Bachelier 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"> • S1.L1. Structures discrètes. • S1.L2. Fondements de la programmation. • S1.L3. Algorithmique et complexité. • S1.L5. Méthode de conception de programmes. • S1.L6. Gestion de l'information. • S.2.2. Modéliser le problème et concevoir une ou plusieurs solutions techniques répondant à ce cahier des charges. • S.2.4. Implémenter et tester la solution retenue. • S1.G1. Mathématiques pour modéliser une situation et prouver l'exactitude d'une affirmation; <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> • faire un choix argumenté sur l'utilisation des principales structures de données utilisées pour représenter des collections; • utiliser à bon escient les algorithmes existants pour manipuler ces structures de données et analyser leur performance; • concevoir et mettre en oeuvre des variantes des algorithmes étudiés; • tester des algorithmes et des structures de données; • utiliser à bon escient les algorithmes et structures de données documentées dans une l'API; • abstraire, modéliser et d'implémenter des solutions efficaces à des problèmes de type « puzzle » algorithmiques; <p>Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils auront développé leur capacité à:</p> <ul style="list-style-type: none"> • analyser de façon critique un problème posé; • tester et debugger des programmes algorithmiques;

	<ul style="list-style-type: none"> • implémenter efficacement des algorithmes courts mais non triviaux; • apprendre par eux-mêmes dans un ouvrage de référence et dans la documentation technique complémentaire.
Modes d'évaluation des acquis des étudiants	<p>Examen sur ordinateur à l'aide d'Inginious https://inginius.info.ucl.ac.be.</p> <p>Un quiz sur deux points peut être organisé mi-quadri et ne compte dans la note de l'étudiant uniquement si il fait remonter celle-ci.</p> <p>Nous organiserons éventuellement un concours algorithmique en fin de quadrimestre, qui pourrait rapporter deux points à l'examen si cela permet d'améliorer la note.</p> <p>Les IA génératives ne peuvent être utilisées ni pour le quizz, ni pour l'examen. Le quizz et l'examen sont individuels, aucune discussion ni collaboration n'est autorisée le temps de l'épreuve.</p> <p>Le non-respect de ces directives peut entraîner une réduction des notes ou d'autres sanctions académiques.</p> <p>Les mêmes conséquences s'appliqueront à un étudiant qui partage volontairement son code ou le rend disponible à d'autres étudiants.</p> <p>Si le professeur le juge nécessaire, un entretien supplémentaire pourra également être organisé pour vérification.</p>
Méthodes d'enseignement	<p>La méthode de pédagogie active suivie dans ce cours est inspirée des classes inversée. Il y a six modules de deux semaines. Chaque module comporte un cours d'introduction à la matière, des exercices théoriques à préparer, des chapitres du livre de référence à lire, un TP de correction des exercices en milieu de module, des travaux sur inginius à réaliser (programmes Java) et finalement un cours de restructuration en fin de module. Une des composantes essentielles de cette pédagogie consiste à faire apprendre chaque étudiant par lui-même. La réussite du processus d'apprentissage présuppose donc une implication significative de chaque étudiant.. L'apprentissage proprement dit reste de la responsabilité de chaque étudiant. Pour réussir l'examen il est impératif que l'étudiant programme régulièrement.</p>
Contenu	<ul style="list-style-type: none"> • Complexité calculatoire, • Algorithmes de tris • Arbres, arbres binaires de recherche, • Arbres équilibrés, • Tries • Dictionnaires et tables de hachage, • Files de priorité et tas, • Graphes • Manipulation de données textuelles (pattern matching et de compression)
Ressources en ligne	<p>https://moodle.uclouvain.be/course/view.php?id=1049 (mainly for communications with students)</p> <p>https://pschaus.github.io/LINFO1121/ (main website, with the exercices to do each week)</p>
Bibliographie	<p>Livre obligatoire:</p> <p>Algorithms, 4th Edition by Robert Sedgwick and Kevin Wayne, Addison-Wesley Professional. ISBN-13: 978-0321573513 ISBN-10: 032157351X</p>
Autres infos	<p>Préalables:</p> <ul style="list-style-type: none"> • maîtriser la programmation dans un langage orienté-objet tel que Java • connaître et utiliser correctement de structures de données élémentaires (piles, files, listes, etc.) • avoir des notions en matière de récursion et de complexité calculatoire. <p>Ces préalables sont matières des cours LEPL1401 et LEPL1402.</p>
Faculté ou entité en charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Filière en Informatique	FILINFO	5		
Bachelier en sciences mathématiques	MATH1BA	5	LEPL1402	
Bachelier en sciences informatiques	SINF1BA	5	LEPL1402	
Approfondissement en statistique et sciences des données	APPSTAT	5		
Mineure Polytechnique	MINPOLY	5		