







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.

5 crédits	30.0 h + 30.0 h	Q1
-----------	-----------------	----

Enseignants	Schaus Pierre ;
Langue d'enseignement	Français
Lieu du cours	Louvain-la-Neuve
Préalables	<ul style="list-style-type: none"> • Maîtrise de la programmation dans un langage orienté objet tel que Java • Connaissance de structures de données élémentaires (piles, files, listes, etc.) • Notions de récurrence et de complexité calculatoire et spatiale. <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>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 , 1 - 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, - implémenter efficacement des algorithmes courts mais non triviaux. - apprendre par eux-mêmes dans un ouvrage de référence et dans la documentation technique complémentaire <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. Examen sur ordinateur à l'aide d'Inginious https://inginius.info.ucl.ac.be.</p> <p>Un quizz sur deux points peut être organisé lors de la semaine smart et ne compte dans la note de l'étudiant uniquement si il fait remonter celle-ci.</p> <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, • appliquer les principes de la programmation orientée-objet tels que généricité, abstraction, composition et réutilisation, • concevoir et mettre en oeuvre des variantes des algorithmes étudiés dans des programmes Java de haute qualité. <p>Les étudiants auront développé des compétences méthodologiques et opérationnelles.</p> <p>En particulier, ils auront développé leur capacité à :</p> <ul style="list-style-type: none"> • analyser de façon critique un problème algorithmique posé • apprendre par eux-mêmes dans un ouvrage de référence et dans la documentation technique complémentaire • produire une solution satisfaisante aux problèmes algorithmiques posés en respectant les délais prescrits.
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>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, • Arbres, arbres binaires de recherche, • Arbres équilibrés, • 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://moodleucl.uclouvain.be/course/view.php?id=7682 https://lsinf1121.readthedocs.io</p>
Bibliographie	<p>Required Textbook: Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne, Addison-Wesley Professional. ISBN-13: 978-0321573513 ISBN-10: 032157351X</p> <p>Exercices and documents https://lsinf1121.readthedocs.io Communication with students using moodle http://moodleucl.uclouvain.be/course/view.php?id=7682</p>
Autres infos	<p>Prerequis: LEPL1401 et LEPL1402 ou cours équivalents</p>
Faculté ou entité en charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Approfondissement en statistique et sciences des données	LSTAT100P	5		
Mineure en statistique, sciences actuarielles et science des données	LSTAT100I	5		
Mineure en sciences de l'ingénieur : informatique (accessible uniquement pour réinscription)	LSINF100I	5		
Master [120] : ingénieur civil en mathématiques appliquées	MAP2M	5		
Master [120] en sciences informatiques	SINF2M	5		
Bachelier en sciences mathématiques	MATH1BA	5	LINFO1225	
Master [60] en sciences informatiques	SINF2M1	5		