

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



Cette unité d'enseignement n'est pas accessible aux étudiants d'échange !

Enseignants	Schaus Pierre ;
Langue d'enseignement	Français
Lieu du cours	Charleroi
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 LSINC1402/LEPL1402.</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, AA1.2 • AA2-3.4, AA2-3.5, AA2-3.7 • AA4.2 • AA5.3 <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.M1, S1.3 • S2.2, S2.3, S2.4 • S4.3 • S5.4 • S6.1, S6.3 <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, • 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>Pour l'examen, les étudiants devront programmer des tâches sur inginius https://inginius.info.ucl.ac.be suivi d'une discussion facultative avec l'enseignant en tant qu'examen oral régulier au cas où l'étudiant ne pense pas que le score inginius reflète son expertise dans le cours.</p> <p>Un quiz à mi-parcours peut être proposé sur deux points pendant la semaine intelligente, mais cela ne peut avoir qu'un impact positif sur votre note finale.</p> <p>Une note de participation sera également prise en compte pour deux points pour la première session, mais pas pour la deuxième session.</p>
Méthodes d'enseignement	<p>La méthode de pédagogie active suivie dans ce cours est inspirée des classes inversées. 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 + Questions sur le site du cours, accessible via Moodle.</p>
Bibliographie	<p>Livre obligatoire: Algorithms, 4th Edition by Robert Sedgwick and Kevin Wayne, Addison-Wesley Professional. ISBN-13: 978-0321573513 ISBN-10: 032157351X</p> <p>Et plus généralement les documents (énoncés des missions, conseils pour l'examen, ...) disponibles sur : http://moodleucl.uclouvain.be/course/view.php?id=7682</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:	SINC

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Bachelier en sciences informatiques	SINC1BA	5	LSINC1402	