

5.0 crédits	30.0 h + 30.0 h	1q
-------------	-----------------	----

Enseignants:	Dupont Pierre ;
Langue d'enseignement:	Français
Lieu du cours	Louvain-la-Neuve
Ressources en ligne:	> http://icampus.uclouvain.be/claroline/course/index.php?cid=sinf1121
Préalables :	-- Maîtrise de la programmation dans un langage orienté-objet tel que Java -- Connaissance de structures de données élémentaires (piles, files, listes, etc.) -- Notions de récursion et de complexité calculatoire.
Thèmes abordés :	-- Mesures de complexité d'un algorithme et méthodes d'analyse de complexité. -- Méthodes de conception de programmes : spécifications et conception orientée-objet. -- 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 (tables de hachage, tas, arbres binaires équilibrés, représentation et manipulation de graphes, traitement de données textuelles, dictionnaires).
Acquis d'apprentissage	Les étudiants ayant suivi avec fruit ce cours seront capables de -- 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. Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils auront développé leur capacité à : -- analyser de façon critique un problème posé -- apprendre par eux-mêmes dans un ouvrage de référence et dans la documentation technique complémentaire -- travailler efficacement en groupe pour l'analyse d'un problème posé, la conception, l'implémentation et la documentation des programmes produits -- équilibrer le travail individuel et le travail en groupe -- gérer le temps d'apprentissage et produire une solution satisfaisante aux problèmes posés en respectant les délais prescrits. <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>
Modes d'évaluation des acquis des étudiants :	Une note de PARTICIPATION sur les différentes missions compte pour 20 % de la note finale + 80 % pour l'examen final. La note de participation ne peut pas être réévaluée en seconde session, mais seul l'examen final compte à 100 % dans la note finale de seconde session. Note: un étudiant qui présenterait pour la *première* fois l'examen final en juin ou en septembre sera évalué de la même manière qu'en première session (20 % participation + 80 % examen)

<p>Méthodes d'enseignement :</p>	<p>La méthode de pédagogie active suivie dans ce cours est inspirée de l'Approche Par Problèmes (APP). Cette méthode repose sur plusieurs phases de travail, certaines encadrées par des tuteurs. Outre les séances tutorées, 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. Le rôle du travail en groupe est principalement de débattre des concepts étudiés et, secondairement, d'organiser le travail de chacun. L'apprentissage proprement dit reste de la responsabilité de chaque étudiant.</p> <p>Le travail est organisé en missions que chaque groupe d'étudiants doit accomplir dans un délai strict (typiquement de 2 semaines par mission). Ces missions comportent des questions auxquelles il faut apporter les meilleures réponses possibles et des problèmes de programmation, pour lesquels il faut produire des programmes Java.</p> <p>Les missions ont pour but principal de créer un contexte et une motivation pour l'apprentissage de la matière et des capacités méthodologiques à développer. Arriver au bout d'une mission n'est donc pas une fin en soi. Il importe d'y être arrivé en ayant profité de l'occasion pour développer les objectifs d'apprentissage explicités dans l'énoncé de chaque mission.</p>
<p>Contenu :</p>	<p>-- 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)</p>
<p>Bibliographie :</p>	<p>Textbook obligatoire : Data Structures and Algorithms in Java, 5th edition, Goodrich and Tamassia, John Wiley & Sons, 2011, ISBN: 978-0-470-39880-7 Et plus généralement les documents (énoncés des missions, conseils pour l'examen, ...) disponibles sur : http://www.icampus.ucl.ac.be/claroline/course/index.php?cid=SINF1121</p>
<p>Cycle et année d'étude :</p>	<p>> Master [120] en sciences et technologies de l'information et de la communication > Master [120] en linguistique > Master [120] : ingénieur civil en mathématiques appliquées > Bachelier en sciences informatiques > Année d'études préparatoire au master en sciences informatiques > Bachelier en sciences économiques et de gestion > Bachelier en sciences mathématiques > Bachelier en sciences de l'ingénieur, orientation ingénieur civil</p>
<p>Faculté ou entité en charge:</p>	<p>INFO</p>