

En raison de la crise du COVID-19, les informations ci-dessous sont susceptibles d'être modifiées, notamment celles qui concernent le mode d'enseignement (en présentiel, en distanciel ou sous un format comodal ou hybride).

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



Cette unité d'enseignement n'est pas dispensée cette année académique !

Langue d'enseignement	Français
Lieu du cours	Charleroi
Préalables	Ce cours suppose acquises les notions de base de la programmation (instructions, variables, boucles, conditions,...) telles qu'enseignées dans le cours LINFO1101 ou LSINC1101.
Thèmes abordés	<ul style="list-style-type: none"> - Le langage Java ; - Types abstraits de données ; - Structures linéaires et arborescentes; - Programmation et algorithmes récursifs ; - Technique de raisonnement sur l'exactitude des programmes ; - Notions de complexité calculatoire d'un algorithme ; - Programmation fonctionnelle ; - Modélisation et programmation orientée-objet ; - Programmation parallèle ; - Test de programmes.
Acquis d'apprentissage	<p>Eu égard au référentiel AA du programme « Bachelier en Sciences de l'Ingénieur, orientation ingénieur civil », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <p>AA 1.1, - AA 1.1, 1.2 1.2 - AA 2.3, 2.4, 2.5, 2.6, 2.7 - - AA 4.2, 4.3, 4.4</p> <p>AA Plus précisément, au terme du cours, l'étudiant sera capable de :</p> <p>2.3, - faire un choix justifié entre plusieurs représentations des informations et plusieurs algorithmes 2.4, pour les traiter, 2.5, - concevoir des (fragment de) programme dans un style fonctionnel, 2.6, - raisonner sur des (fragments de) programmes : complexité des algorithmes et efficacité des 2.7, programmes les mettant en oeuvre, raisonnement récursif, - - appliquer des principes de modélisation orientée-objet, AA 4.2, - concevoir et appliquer des méthodes de test d'un programme, 4.3, - concevoir un programme parallèle simple 4.4</p> <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 un problème de taille moyenne, de proposer une solution informatique pour le résoudre et de la mettre en oeuvre dans le langage Java. <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.</p> <p>Un quizz optionnel a lieu à la mi-quadrant comptant pour deux points uniquement si la note est supérieure à la note de l'examen: $note_finale = \max(quizz_sur_deux + examen_sur_18, examen_sur_20)$.</p> <p>L'examen en fin quadrimestre a pour objectif de vérifier non seulement la connaissance de la matière, mais également la capacité à appliquer les connaissances acquises pour écrire des programmes. L'examen et le quizz utiliseront le système d'évaluation Inginious.</p>

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>Lectures + Exercices Inginius + permanences encadrées en salle informatique</p>
Contenu	<p>Cette unité d'enseignement porte sur :</p> <ul style="list-style-type: none"> - Introduction à Java : compilation, byte-code, machine virtuelle, type primitifs, strings, tableaux - Types abstraits de données ; - Structures linéaires et arborescentes, et leurs applications ; - Formulation récursive d'une solution et algorithmes récursifs ; - Technique de raisonnement : préconditions, postconditions, invariants - Notions de complexité calculatoire et analyse de la complexité temporelle et spatiale d'un algorithme ; - Programmation fonctionnelle et programmation d'ordre supérieur - Modélisation orientée-objet (héritage, composition, réutilisation, polymorphisme, invariant de classe) ; - Introduction aux design patterns ; - Méthodes de tests et de validation de programmes ; - Introduction à la parallélisation : notion de thread et mécanismes de synchronisation. <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> • de concevoir des programmes Java • analyser des programmes en fonction de leur performance • de prouver leurs exactitudes de programmes à l'aide d'invariants • 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. • concevoir et manipuler des structures linéaires et arborescentes et récursives simples • concevoir des tests pour des programmes • concevoir des approches de programmation fonctionnelles pour résoudre de petits problèmes algorithmiques • utiliser à bon escient les designs patterns • concevoir des programmes parallèles simples avec des mécanismes de synchronisation
Ressources en ligne	<p>https://lepl1402.readthedocs.io/ (pour les liens vers les slides et contenu du cours)</p> <p>https://moodleucl.uclouvain.be/course/view.php?id=12884 Moodle (pour la communication avec les étudiants)</p> <p>https://inginius.info.ucl.ac.be/course/LEPL1402 pour les exercices</p>
Faculté ou entité en charge:	EPL

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