

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

Enseignants	. SOMEBODY ;Jodogne Sébastien ;Sadre Ramin ;Schaus Pierre ;
Langue d'enseignement	Français
Lieu du cours	Louvain-la-Neuve
Préalables	<p>Ce cours suppose acquises les notions de base de la programmation (instructions, variables, boucles, conditions,...) telles qu'enseignées dans le cours <b>LEPL1401</b> ou <b>LINFO1101</b>.</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"> <li>• Le langage Java ;</li> <li>• Types abstraits de données ;</li> <li>• Structures linéaires et arborescentes;</li> <li>• Programmation et algorithmes récursifs ;</li> <li>• Technique de raisonnement sur l'exactitude des programmes ;</li> <li>• Notions de complexité calculatoire d'un algorithme ;</li> <li>• Programmation fonctionnelle ;</li> <li>• Modélisation et programmation orientée-objet ;</li> <li>• Programmation parallèle ;</li> <li>• Test de programmes.</li> </ul>
Acquis d'apprentissage	<p><b>A la fin de cette unité d'enseignement, l'étudiant est capable de :</b></p> <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> <ul style="list-style-type: none"> <li>• AA 1.1, 1.2</li> <li>• AA 2.3, 2.4, 2.5, 2.6, 2.7</li> <li>• AA 4.2, 4.3, 4.4</li> </ul> <p>Plus précisément, au terme du cours, l'étudiant sera capable de :</p> <ul style="list-style-type: none"> <li>• faire un choix justifié entre plusieurs représentations des informations et plusieurs algorithmes pour les traiter;</li> <li>• concevoir des (fragment de) programme dans un style fonctionnel;</li> <li>• raisonner sur des (fragments de) programmes : complexité des algorithmes et efficacité des programmes les mettant en oeuvre, raisonnement récursif;</li> <li>• appliquer des principes de modélisation orientée-objet;</li> <li>• concevoir et appliquer des méthodes de test d'un programme;</li> <li>• concevoir un programme parallèle simple.</li> </ul> <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"> <li>• analyser un problème de taille moyenne, de proposer une solution informatique pour le résoudre et de la mettre en oeuvre dans le langage Java.</li> </ul>
Modes d'évaluation des acquis des étudiants	<p>Un examen sur INGINIOUS a lieu en fin quadrimestre et 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.</p> <p>Une interrogation optionnelle et individuelle a lieu à la moitié du quadrimestre, comptant pour maximum deux points dans la note finale du cours, et uniquement si cela est au bénéfice de l'étudiant. L'utilisation d'outil type IA générative et toute collaboration est strictement interdite durant cette interrogation.</p> <p>La note finale est calculée selon la formule : <math>note\_finale\_sur\_20 = \max(interrogation\_sur\_2 + examen\_sur\_18, examen\_sur\_20)</math>. Les points obtenus pour l'interrogation seront conservés pour la session d'août.</p> <p>En cas de suspicion de plagiat (par exemple détecté à l'aide d'un outil de plagiat), les titulaires du cours se réservent le droit de demander à l'étudiant de passer une interrogation orale sur le quizz ou l'examen.</p>

<p>Méthodes d'enseignement</p>	<p>Les enseignants présenteront la matière en auditoire lors d'un cours traditionnel, avec pour support le syllabus intitulé "Programming Concepts in Java".</p> <p>Pour chaque sujet abordé, des exercices sont accessibles en continu sur la plateforme INGINIOUS, permettant d'appliquer les concepts théoriques.</p> <p>Lors des séances de travaux pratiques, des assistants ou tuteurs seront présents pour soutenir et guider les étudiants.</p> <p>Chaque étudiant est pleinement responsable de son apprentissage. Afin de réussir l'examen informatique, il est essentiel pour l'étudiant de s'exercer régulièrement en Java avec l'outil IntelliJ.</p>
<p>Contenu</p>	<p>Cette unité d'enseignement porte sur :</p> <ul style="list-style-type: none"> <li>• Introduction à Java : compilation, byte-code, machine virtuelle, type primitifs, strings, tableaux ;</li> <li>• Types abstraits de données ;</li> <li>• Structures linéaires et arborescentes, et leurs applications ;</li> <li>• Formulation récursive d'une solution et algorithmes récursifs ;</li> <li>• Technique de raisonnement : préconditions, postconditions, invariants ;</li> <li>• Notions de complexité calculatoire et analyse de la complexité temporelle et spatiale d'un algorithme ;</li> <li>• Programmation fonctionnelle et programmation d'ordre supérieur ;</li> <li>• Modélisation orientée-objet (héritage, composition, réutilisation, polymorphisme, invariant de classe) ;</li> <li>• Introduction aux design patterns ;</li> <li>• Méthodes de tests et de validation de programmes ;</li> <li>• Introduction à la parallélisation : notion de threads et mécanismes de synchronisation.</li> </ul> <p>Les étudiants ayant suivi avec fruit ce cours seront capables de :</p> <ul style="list-style-type: none"> <li>• Concevoir des programmes Java ;</li> <li>• Analyser des programmes en fonction de leur performance ;</li> <li>• Prouver l'exactitude de programmes à l'aide d'invariants ;</li> <li>• Appliquer les principes de la programmation orientée-objet tels que généralité, abstraction, composition et réutilisation ;</li> <li>• Concevoir et mettre en oeuvre des variantes des algorithmes étudiés dans des programmes Java de haute qualité ;</li> <li>• Concevoir et manipuler des structures linéaires et arborescentes et récursives simples ;</li> <li>• Concevoir des tests pour des programmes ;</li> <li>• Concevoir des approches de programmation fonctionnelles pour résoudre de petits problèmes algorithmiques ;</li> <li>• Utiliser à bon escient les designs patterns ;</li> <li>• Concevoir des programmes parallèles simples avec des mécanismes de synchronisation.</li> </ul>
<p>Ressources en ligne</p>	<p><a href="https://pschaus.github.io/book_lepl1402/">https://pschaus.github.io/book_lepl1402/</a> (pour le syllabus)</p> <p><a href="https://moodle.uclouvain.be/course/view.php?id=3831">https://moodle.uclouvain.be/course/view.php?id=3831</a> (Moodle pour les slides et la communication avec les étudiants)</p> <p><a href="https://inginius.info.ucl.ac.be/course/LEPL1402">https://inginius.info.ucl.ac.be/course/LEPL1402</a> (pour les exercices)</p>
<p>Faculté ou entité en charge:</p>	<p>BTCI</p>

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Bachelier en sciences de l'ingénieur, orientation ingénieur civil	FSA1BA	5	LEPL1401	
Master [120] en linguistique	LING2M	5		
Bachelier en sciences mathématiques	MATH1BA	6	LINFO1101	
Bachelier en sciences informatiques	SINF1BA	5	LINFO1101	
Approfondissement en statistique et sciences des données	APPSTAT	5		
Mineure en sciences informatiques	MINSINF	5		
Mineure en statistique, sciences actuarielles et science des données	MINSTAT	5		