

6 crédits	30.0 h + 30.0 h	Q1
-----------	-----------------	----

Enseignants	Bonaventure Olivier ;Pecheur Charles coordinateur ;
Langue d'enseignement	Français
Lieu du cours	Louvain-la-Neuve
Thèmes abordés	<p>Principes de base de la programmation</p> <ul style="list-style-type: none"> <li>• Valeurs, variables, affectation</li> <li>• Types primitifs, représentation des nombres</li> <li>• Expressions et instructions, conditions et boucles</li> <li>• Méthodes, paramètres, résultats et appels</li> <li>• Spécifications</li> <li>• Edition, compilation et exécution de programmes</li> </ul> <p>Programmation orientée-objets</p> <ul style="list-style-type: none"> <li>• Classes, objets, instances</li> <li>• Variables de classe et d'instance</li> <li>• Visibilité, portée et durée de vie d'une variable</li> <li>• Interfaces et implémentations</li> <li>• Extension d'une classe et héritage, polymorphisme</li> </ul> <p>Fonctionnalités et structures de données</p> <ul style="list-style-type: none"> <li>• Chaînes de caractères</li> <li>• Fichiers et entrées-sorties</li> <li>• Mécanismes d'exception</li> <li>• Tableaux simples et tableaux multidimensionnels</li> <li>• Utilisation de packages</li> <li>• Structures chaînées</li> </ul> <p>Méthodologie de la programmation</p> <ul style="list-style-type: none"> <li>• Analyse d'un problème informatique</li> <li>• conception, spécification et mise en oeuvre d'une solution</li> </ul>
Acquis d'apprentissage	<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"> <li>• S1.I2</li> <li>• S2.2, S2.4</li> </ul> <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> <li>• démontrer une bonne compréhension des concepts et de la méthodologie de la programmation orientée-objet,</li> <li>• utiliser à bon escient les éléments d'un langage orienté-objet tel que Java,</li> <li>• analyser un problème simple, proposer une solution informatique pour le résoudre et la mettre en oeuvre en Java.</li> </ul> <p>1</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"> <li>• décomposer un problème simple en sous-problèmes;</li> <li>• exprimer la résolution d'un problème sous forme d'algorithme;</li> <li>• définir et spécifier avec rigueur et précision les fonctionnalités attendues d'un programme informatique;</li> <li>• lire et comprendre des programmes existants;</li> <li>• utiliser des outils informatiques de support à la programmation;</li> <li>• exploiter la documentation technique d'un langage informatique;</li> <li>• réaliser une application informatique de petite taille sous forme de programme Java de bonne qualité;</li> <li>• identifier les cas à tester pour la validation d'un programme simple.</li> </ul> <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>Une évaluation intermédiaire a lieu en milieu de quadrimestre. La note obtenue pour cette évaluation compte pour 1/3 de la note finale, uniquement si elle est supérieure à la note de l'examen.</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 de programmes Java simples, mais corrects.</p>
Méthodes d'enseignement	<p>Les méthodes utilisées privilégieront l'apprentissage actif des étudiants.</p> <p>Les modalités précises de mise en oeuvre d'une participation active de l'étudiant dans son apprentissage sont laissées aux titulaires, dans le respect des orientations pédagogiques de la Faculté.</p>
Contenu	<p>Interface, classe, object, instance :</p> <ul style="list-style-type: none"> <li>• Types primitifs, valeurs, variables, affectation</li> <li>• Expressions</li> <li>• Types d'instruction</li> <li>• Méthodes, paramètres, résultats et appels</li> <li>• Spécifications</li> <li>• Création d'instances, références vers un objet</li> <li>• Variables de classe, d'instance, locales et leur portée</li> <li>• Polymorphisme Visibilité des méthodes et des variables</li> <li>• Extension d'une classe et héritage</li> <li>• Mécanismes d'exception</li> <li>• Fichiers et entrées-sorties</li> <li>• Tableaux simples et tableaux multidimensionnels</li> <li>• Vecteurs et utilisation de packages</li> <li>• Listes chaînées Interfaces graphiques et programmation événementielle</li> </ul>
Ressources en ligne	<p><a href="http://moodleucl.uclouvain.be/course/view.php?id=7654">http://moodleucl.uclouvain.be/course/view.php?id=7654</a></p>
Bibliographie	<p>Le cours s'appuie sur l'ouvrage de référence suivant (en anglais):  D. Eck, Introduction to Programming Using Java, Sixth Edition, 2013, <a href="http://math.hws.edu/javanotes/">http://math.hws.edu/javanotes/</a></p>
Faculté ou entité en charge:	<p>INFO</p>

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] en linguistique	LING2M	6		
Master [120] en anthropologie	ANTR2M	6		
Bachelier en sciences mathématiques	MATH1BA	6		
Bachelier en sciences informatiques	SINF1BA	6		
Master [120] en science des données, orientation statistique	DATS2M	6		
Mineure d'accès en statistique et science des données	LSTAT100I	6		
Mineure en sciences informatiques	LINFO100I	6		
Approfondissement en sciences mathématiques	LMATH100P	6		
Approfondissement en statistique et science des données	LSTAT100P	6		
Approfondissement en sciences mathématiques	TMATH100P	6		