










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"> • Valeurs, variables, affectation • Types primitifs, représentation des nombres • Expressions et instructions, conditions et boucles • Méthodes, paramètres, résultats et appels • Spécifications • Edition, compilation et exécution de programmes <p>Programmation orientée-objets</p> <ul style="list-style-type: none"> • Classes, objets, instances • Variables de classe et d'instance • Visibilité, portée et durée de vie d'une variable • Interfaces et implémentations • Extension d'une classe et héritage, polymorphisme <p>Fonctionnalités et structures de données</p> <ul style="list-style-type: none"> • Chaînes de caractères • Fichiers et entrées-sorties • Mécanismes d'exception • Tableaux simples et tableaux multidimensionnels • Utilisation de packages • Structures chaînées <p>Méthodologie de la programmation</p> <ul style="list-style-type: none"> • Analyse d'un problème informatique • conception, spécification et mise en oeuvre d'une solution
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"> • S1.I2 • S2.2, S2.4 <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> • démontrer une bonne compréhension des concepts et de la méthodologie de la programmation orientée-objet, • utiliser à bon escient les éléments d'un langage orienté-objet tel que Java, • analyser un problème simple, proposer une solution informatique pour le résoudre et la mettre en oeuvre en Java. <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"> • décomposer un problème simple en sous-problèmes; • exprimer la résolution d'un problème sous forme d'algorithme; • définir et spécifier avec rigueur et précision les fonctionnalités attendues d'un programme informatique; • lire et comprendre des programmes existants; • utiliser des outils informatiques de support à la programmation; • exploiter la documentation technique d'un langage informatique; • réaliser une application informatique de petite taille sous forme de programme Java de bonne qualité; • identifier les cas à tester pour la validation d'un programme simple. <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>

<p>Modes d'évaluation des acquis des étudiants</p>	<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>
<p>Méthodes d'enseignement</p>	<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>
<p>Contenu</p>	<p>Interface, classe, object, instance :</p> <ul style="list-style-type: none"> • Types primitifs, valeurs, variables, affectation • Expressions • Types d'instruction • Méthodes, paramètres, résultats et appels • Spécifications • Création d'instances, références vers un objet • Variables de classe, d'instance, locales et leur portée • Polymorphisme Visibilité des méthodes et des variables • Extension d'une classe et héritage • Mécanismes d'exception • Fichiers et entrées-sorties • Tableaux simples et tableaux multidimensionnels • Vecteurs et utilisation de packages • Listes chaînées Interfaces graphiques et programmation événementielle
<p>Ressources en ligne</p>	<p>http://moodleucl.uclouvain.be/course/view.php?id=7654</p>
<p>Bibliographie</p>	<p>Le cours s'appuie sur l'ouvrage de référence suivant (en anglais): D. Eck, Introduction to Programming Using Java, Sixth Edition, 2013, http://math.hws.edu/javanotes/</p>
<p>Faculté ou entité en charge:</p>	<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		