

Enseignants:	
Langue d'enseignement:	Français
Lieu du cours	Louvain-la-Neuve
Ressources en ligne:	> http://www.icampus.ucl.ac.be/claroline/course/index.php?cid=SINF1150
Préalables :	-- Connaissance de base en programmation dans un langage de haut niveau -- Capacité à fonder la programmation en langage de haut niveau sur une analyse rigoureuse des concepts sous-jacents. L'acquisition de ces prérequis est l'objet du cours SINF1160
Thèmes abordés :	-- Entrées/Sorties en langage machine et multiprogrammation -- Fichiers séquentiels et fichiers de texte -- Récursivité -- Classes graphiques en Java (AWT) -- Mécanismes de la programmation orienté objet en Java -- Structures de données logiques et physiques -- Structures de données récursives -- Structures de données chaînées
Acquis d'apprentissage	Les étudiants ayant suivi avec fruit ce cours seront capables de -- programmer des codes de taille réduite correctement et non par essais et erreurs, -- décrire et justifier l'emploi de concepts de base de la programmation tels que les fichiers, les interfaces graphiques et les structures de données chaînées et récursives, -- décrire les mécanismes de base de la programmation orientée objet, et les illustrer dans le langage Java Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils ont développé leur capacité à -- avoir un regard critique sur leurs réalisations et justifier les étapes du raisonnement qui les y a conduit <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 :	-- Examen écrit -- Les projets and devoirs sont évalués et cette évaluation contribue à 30% de la cote finale de l'examen.
Méthodes d'enseignement :	L'enseignement est organisé avec un cours magistral complété de séances d'exercices et de travaux à rendre. Les travaux pratiques consistent d'une part en séances en salles de cours où des exercices simples sont résolus avec l'aide d'un assistant et, d'autre part, en devoirs individuels et/ou projets de taille réduite résolus par groupes de 2 étudiants. Les devoirs consistent à finaliser les exercices résolus en séances en allant jusqu'à leur implémentation (incluant quelques spécification et commentaires dans le code produit).
Contenu :	Le contenu du cours est le suivant (dans cet ordre). -- On complète l'étude du langage machine en étudiant le fonctionnement des instructions d'entrées/sorties au niveau du langage machine. On motive ainsi l'existence du système d'exploitation et l'introduction de la multiprogrammation. On en déduit également la nécessité du concept logique de fichier. -- On donne deux modèles de données complètement spécifiés pour manipuler des fichiers en Java. Comme les classes standard permettant de manipuler des fichiers en Java sont trop complexes pour être décrite complètement avec rigueur à ce niveau, d'autres classes spécifiées et implémentées par le titulaire du cours sont fournies. Elles proposent un modèle semblable à celui des fichiers du langage Pascal. Il n'y a que deux classes à étudier : l'une permet de manipuler des fichiers de texte. L'autre permet de manipuler des fichiers binaires (suites quelconques d'octets). - On introduit la récursivité comme méthode de raisonnement, basée sur la récurrence, pour construire des algorithmes. On insiste, dans ce contexte, sur le rôle des variables locales et des paramètres. -- La notion de classe "définissable par l'utilisateur" est introduite, ainsi que les notions associées : variables d'instance, méthodes non statiques, constructeurs, attributs d'accessibilité. -- On introduit la notion d'héritage en programmation orientée objet en montrant qu'elle est motivée par l'objectif de réutiliser à moindre frais du code existant. -- On explique les principes à la base des bibliothèques standard pour construire des interfaces graphiques simples. On illustre avec cela la théorie du point précédent (héritage). Un très petit nombre de classes graphiques sont expliquées. Elles suffisent pour réaliser des interfaces graphiques simples. Par contre, le modèle des événements est élucidé avec soin.

	<p>-- Les éléments essentiels de la programmation orientée objet (en Java) sont expliqués : interfaces, classes abstraites, méthodes abstraites, appel dynamique des méthodes. On introduit aussi les exceptions.</p> <p>-- Les classes abstraites sont illustrées par leur utilisation pour définir la structure de données d'arbre binaire (version ``récursive").</p> <p>-- Les interfaces sont utilisés pour illustrer le concept de type de données logique.</p> <p>-- Les structures de données de type liste sont présentées selon les deux approches de type de données logique (interfaces) et physique (classes fournissant une implémentation). On montre que les concepts d'interface et de classe permettent une forme limitée d'héritage multiple permettant de réutiliser une structure physique (classe) pour implémenter une structure logique (interface).</p>
<p>Bibliographie :</p>	<p>-- slides available online</p>
<p>Autres infos :</p>	
<p>Cycle et année d'étude: :</p>	<p>> Master [120] en linguistique > Master [120] en sciences et technologies de l'information et de la communication > Master [120] en statistiques, orientation générale > Bachelier en sciences de l'ingénieur, orientation ingénieur civil architecte > Bachelier en sciences économiques et de gestion > Bachelier en sciences mathématiques</p>
<p>Faculté ou entité en charge:</p>	<p>INFO</p>