

7.0 crédits

0 h + 60.0 h

2q

Enseignants:	Mens Kim ;
Langue d'enseignement:	Français
Lieu du cours	Louvain-la-Neuve
Ressources en ligne:	> http://www.icampus.ucl.ac.be/claroline/course/index.php?cid=LINF1151
Préalables :	introduction à la programmation dans un langage de haut niveau (p.e. SINF1160, SINF1161)
Thèmes abordés :	<ul style="list-style-type: none"> -- première expérience de la construction de logiciel à base d'objets en leur permettant de développer une application informatique assez simple en utilisant un IDE (Integrated Development Environment). -- programmation dans un langage de haut niveau -- tests unitaires -- qualité d'une application: correcte, modulaire, lisible, documentée -- programmation orienté-objet: classes, interface, héritage, polymorphisme, ... -- collaboration entre développeurs
Acquis d'apprentissage	<p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> -- réaliser une application d'ampleur réduite, mais correcte, modulaire, lisible, et bien documentée ; -- mettre en pratique les différents concepts de la programmation orientée objets comme l'encapsulation des données, la séparation de l'implémentation et du comportement, les classes, sous-classes, interfaces et l'héritage, le polymorphisme, la hiérarchie des classes, les classes collection, les protocoles d'itération, etc. -- comprendre et étendre une application existante qui sera fournie comme point de départ pour un travail de développement ; -- mettre en oeuvre des tests unitaires pour valider l'exactitude d'un programme; -- structurer une application de façon modulaire en classes et méthodes ; -- analyser les besoins pour une extension simple à cette application. <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"> -- programmer de façon efficace dans un langage de haut niveau (Java); -- utiliser un outil tel que JUnit pour réaliser des tests unitaires; -- utiliser un environnement de programmation comme Eclipse comportant des outils de programmation intégrés comme un éditeur intelligent, compilateur, debugger, et des outils de maniement de fichiers, de tests, de documentation, de réusinage; -- argumenter les choix de conception mis en oeuvre en ce qui concerne les fonctionnalités de la solution recherchée et comparer ses choix avec des solutions alternatives; -- collaborer avec d'autres développeurs plus ou moins expérimentés au travers d'un forum de discussion. <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>Différentes notes contribuent à la note finale :</p> <ul style="list-style-type: none"> -- Deux évaluations intermédiaires, au cours du quadrimestre, sur l'avancement du projet de programmation, basées sur les rapports rédigés par les étudiants et la qualité du code. -- Une évaluation finale, à la fin du projet, comprenant un rapport de synthèse et une défense orale où chaque binôme défend son projet de programmation et démontre le bon fonctionnement de l'application. <p>Comme il s'agit d'un laboratoire de programmation, l'absence à une des évaluations intermédiaires ou la non remise des documents requis, implique une note finale correspondant à une absence pour le cours.</p> <p>En cas de non réussite en première session, l'étudiant devra individuellement finaliser et améliorer le même projet de programmation, y compris les extensions demandées, mais aussi ajouter une extension additionnelle non-triviale au projet de programmation. La note finale pour la session de septembre portera alors sur (et seulement sur) la qualité de l'entièreté du code produit et de sa documentation et sera déterminée sur base d'un rapport et d'une défense du projet, lors de la session de septembre.</p>
Méthodes d'enseignement :	<p>Ce cours est conçu comme un projet de programmation dont la réalisation se déroule en trois phases. Les premières phases sont plus guidées et permettent aux étudiants de découvrir l'application initiale qui leur est fournie, en programmant quelques extensions plutôt faciles. Pour la dernière phase, par contre, l'étudiant a l'opportunité d'aller plus loin dans la réalisation du système logiciel en proposant et en réalisant ses propres extensions. Dans chaque phase, les étudiants travaillent sur une sous-partie différente du problème prédéfini dans l'énoncé du projet.</p> <p>Les étudiants travaillent en binôme pour avoir l'opportunité d'échanger les points de vue. La collaboration entre binômes est organisée via le forum de discussion du cours où chacun peut poser ses questions. Les autres étudiants et les encadrants proposent des pistes de solution.</p>

<p>Contenu :</p>	<ul style="list-style-type: none"> -- Analyser le problème; -- Concevoir la structure de l'application à programmer; -- Programmer cette application en Java; -- Documenter son application (commentaires, préconditions, postconditions, choix d'implémentation alternatives, algorithmes, manuel); -- Tester son application; -- Rédiger un rapport; -- Faire une défense et démonstration de l'application finale.
<p>Bibliographie :</p>	<p>Les notes de cours et livres de référence utilisés pour les cours SINF1160 et SINF1161 restent valables pour le cours SINF1151, ainsi que d'autres livres d'introduction à la programmation en Java.</p>
<p>Cycle et année d'étude: :</p>	<p>> Bachelier en sciences informatiques</p>
<p>Faculté ou entité en charge:</p>	<p>INFO</p>