



Collège de
Maisonneuve

Automne 2022

Plan de cours

420-930-MA gr. 21624

Algorithmes et modèles de programmation

Programme : LEA.8F Sécurité des applications mobiles, Web et de bureau
Formation continue

Pondération : 2-4-2

Préalables : 420-910-MA Concepts de programmation 1
420-920-MA Concepts de programmation 2

Professeur : Antonio Tavares

Courriel : atavares@cmaisonneuve.qc.ca

Présentation du cours

Le rôle de ce cours dans le programme est d'initier les étudiants aux structures de données et aux modèles de conception de logiciels. Il s'appuie sur les cours Concepts de programmation 1 et Concepts de programmation 2 pour faire passer les étudiants au développement d'applications dans un langage de programmation familier. Les cours de développement d'applications utiliseront les concepts abordés dans ce cours comme base.

La compétence OOSR sera continuellement développée et finalisée dans le cours Développement d'applications 1 (Bureau) du bloc 3 et le cours Développement d'applications 2 (Mobile) du bloc 4.

Les étudiants apprennent les algorithmes de base courants, comme les algorithmes de tri, les modèles de conception et les modèles architecturaux. À l'issue de ce cours, les étudiants devraient être en mesure d'analyser des exigences données, de choisir un algorithme en fonction de son efficacité, d'appliquer les algorithmes appropriés, de mettre en œuvre les modèles de conception adéquats et de mettre en œuvre les modèles architecturaux pertinents. Les étudiants devraient également être capables de documenter, de déployer et de contrôler la qualité du programme.

Compétence(s) développée(s) dans ce cours :

NO	Énoncé	Éléments
OOSR	Effectuer le développement d'applications natives sans base de données.	<ul style="list-style-type: none">• Analyser le projet de développement de l'application• Préparer l'environnement de développement informatique• Générer ou programmer l'interface graphique• Programmer la logique de l'application• Contrôler la qualité de l'application• Participer au déploiement de l'application• Produire la documentation

Déroulement du cours

Semaine	Théorie	Laboratoire
1 à 2	<p>Révision</p> <p>Aspects avancés de la POO</p> <ul style="list-style-type: none"> • Exceptions • Classes abstraites • Interfaces (ex : Interface Comparable) • Héritage multiple vs interface 	<p>Lectures, recherches internet</p> <p>Installation de l'environnement</p> <p><u>Exercices formatifs</u></p> <p><u>Laboratoire 1</u></p>
3 à 4	<p>Algorithmique</p> <ul style="list-style-type: none"> • Études d'algorithmes de base et mesures d'efficacité temporelle <ul style="list-style-type: none"> • Recherche <ul style="list-style-type: none"> • Recherche linéaire • Recherche dichotomique (binaire) • Tris <ul style="list-style-type: none"> • Par comparaison • Bulles (bubble sort) • Rapide (quicksort) 	<p><u>Exercices formatifs</u></p> <p><u>Examen 1</u></p>
5 à 10	<p>Structures de données et leur efficacité temporelle</p> <ul style="list-style-type: none"> • Récursivité • Expressions lambda et le traitement de flux pour simplifier le code • Planification des tests • Étude théorique/pratique sur les différentes structures de données et leur représentation <ul style="list-style-type: none"> • Piles • Listes <ul style="list-style-type: none"> • Simples • Doublement chaînées • Circulaires • Arbres • Graphes • Collections en JAVA <ul style="list-style-type: none"> • ArrayList, • LinkedList, • Les Map • Les Set • Stack • etc. • Itérateurs 	<p><u>Exercices formatifs</u></p> <p><u>Laboratoire 2</u></p> <p><u>Examen 2</u></p>

9 à 12	<p>Se connecter à des bases de données existantes et manipuler leurs données.</p> <ul style="list-style-type: none"> • JDBC • Choix des structures de données pour manipuler les données reçues de la base de données <p>Application de certains patrons (modèles) de conception lors de la conception d'un projet</p> <ul style="list-style-type: none"> • Itérateur • Fabrique abstraite • Singleton • Décorateur • Modèle MVC (Modèle-Vue-Contrôleur) • Modèle MVP (Modèle-Vue-Présentateur) • Modèle MVVC (Modèle-Vue-Vuemodèle) • etc. (selon les besoins) 	<p><u>Exercices formatifs</u></p> <p><u>Laboratoire 3</u></p>
13 à 15	<p>Multi-threading</p> <ul style="list-style-type: none"> • Implémentation des cas simples • Threads de travail • Pool de threads • Étude de plusieurs exemples et application à des cas réels <p>Révision et exercices pour l'épreuve finale</p>	<p><u>ÉXAMEN DE SYNTHÈSE</u></p>

Activités d’enseignement et d’apprentissage

Les principales méthodes pédagogiques utilisées dans ce cours sont :

Des exercices formatifs.

Travaux pratiques, réalisés en classe et en dehors des heures de cours prévues à l’horaire, Ces travaux pratiques sont réalisés soit individuellement soit en équipe.

Évaluation formative

L’évaluation formative se fera à l’aide d’exercices formatifs réalisés en classe ou en dehors des heures de cours.

Évaluation sommative

L’évaluation sommative s’effectuera au moyen de trois examens et de trois travaux pratiques.

Laboratoires 1	10%
Laboratoires 2	15%
Laboratoires 3	15%
Examen 1	15% *
Examen 2.....	15% *
Épreuve finale	30% *

* Les évaluations annotées d’une étoile (*) font partie de l’ensemble des évaluations dont le seuil de passage pour l’ensemble de ces évaluations est de 50%. Ces évaluations se font de manière individuelle sous environnement contrôlé, en classe.

CRITÈRES D’ÉVALUATION DE L’ÉPREUVE FINALE

L’épreuve finale, qui compte pour 30% de la note du cours, sera constituée d’un l’examen pratique en classe dont les étudiants ont droit à toutes leurs notes de cours.

Critères d’évaluation et pondération (sur 100%)

À partir d’un énoncé :

• Concevoir le diagramme de classes UML	10%
• Création des interfaces graphiques	15%
• Création et gestion des différentes classes	20%
• Structures de données utilisée et leur efficacité selon le contexte	20%
• Ergonomie générale de l’application	10%
• Validations, gestion des exceptions et patrons de conception	15%
• Vidéo des tests de l’application	10%

Médiagraphie

Notes de cours en LEA

Ouvrages de référence non obligatoires

- Les fondamentaux du langage Java, Thierry Groussard, Éd. ENI – Ressources informatiques.
- Comment programmer en Java par Deitel et Deitel, Éditions Reynald Goulet
- Programmer en Java, Claude Delannoy ed. Eyrolles.

Sites intéressants pour le cours (d'autres dans LEA)

<https://docs.oracle.com/javase/8/docs/api/overview-summary.html>

https://www.w3schools.com/java/java_date.asp

<https://www.emse.fr/~picard/cours/1A/java/livretJava.pdf>