
TITRE DU COURS :	Structures de données
NUMÉRO DU COURS :	420 3C6 MO
SESSION :	Automne 2024
PONDÉRATION DU COURS :	3-3-3
DÉPARTEMENT :	Techniques de l'informatique, 420.B0
PROGRAMME :	Développement d'applications informatiques

	Nom	Local	Téléphone	Courriel
PROFESSEURS :	Mathieu Bergeron	B3320	(450) 975-6100 p.6727	mathieu.bergeron@cmontmorency.qc.ca
	Marc-Olivier Tremblay	B3322	via Teams	mo.tremblay@cmontmorency.qc.ca
COORDONNATEURS :	Abdelhabib Yahia	B3322	(450) 975-6100 p.7424	abdelhabib.yahia@cmontmorency.qc.ca
	Sylvain Labranche	B3330	(450) 975-6100 p.6769	sylvain.labranche@cmontmorency.qc.ca

Ce plan de cours est sujet à toutes les clauses contenues dans le document décrivant la politique institutionnelle d'évaluation des apprentissages (PIEA) du collège. Il est aussi sujet à toutes les règles et procédures pour les personnes étudiant en techniques de l'informatique. Veuillez-vous référer à ces documents pour plus d'information sur chacune des clauses spécifiques.

Département d'informatique: https://ntro.ca/3c6/plan/regles_departement_informatique.pdf

PIÉA : <https://ntro.ca/3c6/presentation/piea.pdf>

1. DESCRIPTION DU COURS

Ce cours permettra à l'étudiant·e d'écrire, d'appliquer et de mettre en œuvre des programmes plus complexes en utilisant différentes structures de données. Il vérifiera différentes façons de structurer les données.

2. OBJECTIF INTÉGRATEUR

Exploiter les principes avancés de la programmation orientée objet.

3. OBJECTIFS MINISTÉRIELS

00SR (éléments 1, 4, 5) : Effectuer le développement d'applications natives sans base de données.

4. OBJECTIFS D'APPRENTISSAGE

1. Appliquer des techniques de programmation avancées pour résoudre des problèmes.

2. Utiliser des structures des données avancées.
3. Mettre en œuvre la programmation en fonction des structures de données.

5. ATTITUDES PROFESSIONNELLES

1. Rigueur
2. Autonomie

6. HABILITÉS TRANSDISCIPLINAIRES

1. Compétences langagières : déterminer l'information à rédiger et effectuer une notation claire du travail, tout en utilisant la terminologie appropriée à la profession.
2. Profil TIC : exploiter les TIC de manière efficace et responsable; rechercher, traiter et présenter de l'information.
3. Risques en matière de santé et de sécurité au travail : risques chimiques ou dangers d'ordre chimique; risques physiques ou dangers d'ordre physique; risques ergonomiques ou dangers d'ordre ergonomique; risques psychosociaux ou dangers d'ordre psychosocial; maintenir un environnement physique de qualité.
4. Entrepreneuriat : travailler à son compte en développement d'applications ou de réseaux; créer un bureau de consultation

7. PRÉALABLES

Ces cours sont préalables absolus au présent cours 420-ZD5-MO – Programmation orientée objet (2 ^e) 420-ZF5-MO – Programmation structurée (2 ^e)	Ce cours est préalable relatif au présent cours
Le présent cours est préalable absolu aux cours ci-après 420-4E6-MO – Analyse et conception de modèles (4 ^e) 420-5G4-MO – Technologies émergentes des applications (5 ^e) 420-6AM-MO – Stage en développement d'applications (6 ^e)	

8. ACTIVITÉ D'ÉVALUATION FINALE DU COURS

L'étudiant·e écrira des programmes plus complexes en appliquant des techniques de programmation avancées et en utilisant différentes structures de données.

L'étudiant·e sera évalué·e à partir des critères de performance suivants :

1. Analyse juste des documents de conception de représentation des algorithmes
2. Détermination correcte des tâches à effectuer
3. Programmation correcte des interactions et des structures de données
4. Application rigoureuse des techniques de programmation dans les classes
5. Pertinence des correctifs

9. OBJECTIFS DÉTAILLÉS

Pour des raisons pédagogiques, les points suivants seront traités progressivement au cours de la session et non de façon séquentielle.

Objectif d'apprentissage	Contenus	Poids
1. Appliquer des techniques de programmation avancées pour résoudre des problèmes (OOSR : élément 1, 4, 5)	1.1 Récursivité. 1.2 Algorithmes de tri, p.ex. tri fusion 1.3 Gestion de tableaux dynamiques (à taille variable) 1.4 Utilisation des classes du langage de programmation pour manipuler des tableaux 1.5 Algorithmes de mise à jour d'un fichier d'objets, p.ex. JSON	25%
2. Utiliser des structures des données avancées (OOSR : élément 1, 4, 5)	2.1 Définition et implémentation de listes 2.2 Utilisation des classes du langage de programmation pour gérer des listes 2.3 Définition et implémentation d'un mappage 2.4 Utilisation des classes du langage de programmation pour gérer des mappages 2.5 Adressage dispersé (hachage) : fonctions de hachage, traitement des collisions 2.6 Notion de Modèle	40%
3. Mettre en œuvre la programmation en fonction des structures de données (OOSR : élément 1, 4, 5)	3.1 Programmation des classes du programme 3.2 Choix des types de données élémentaires et des structures de données 3.3 Intégration des classes dans le programme 3.4 Documentation du programme 3.5 Contrôle de la qualité du code 3.6 Vérification du fonctionnement du programme 3.7 Respect des normes de programmation 3.8 Utilisation d'outils de gestion de version (p.ex. Git)	35%

10. DÉROULEMENT DU COURS

Le calendrier peut être sujet à changement.

Étape	Objectifs	Contenu
Étape 1 (Semaines 1-3)	1.4,1.5,3	1. Installation et introduction aux outils du cours 2. Rappel concepts POO 3. Rappel tableau d'objets (boucle for, trouver le plus petit élément)
EXAMEN 1 : séance 3.2		
Étape 2 (Semaines 4-6)	1.5,2.1,2.2,2.6,3	1. Modélisation Json 2. Modélisation Java (graphe d'objets) 3. Récursivité (dans les données Vs pile d'appel)
EXAMEN 2 : séance 6.2		
Étape 3 (Semaines 6-8)	1.2,3	1. Structures génériques (paramètres de type) 2. Tri naïf + notion d'efficacité 3. Tri fusion
EXAMEN 3 : séance 8.2		
Étape 4 (Semaines 9-11)	1.3,2.1,2.2,3	1. Liste naïve 2. Liste avec tableau 3. Liste chaînée (simple et double)

EXAMEN 4: séance 11.2		
Étape 5 (Semaines 12-15)	2.3,2.4,2.5,3	<ol style="list-style-type: none"> 1. Map naïf 2. Map avec table de hachage 3. Map avec arbre
EXAMEN 5: séance 15.2		

11. MÉTHODOLOGIE

11.1. COURS MAGISTRAUX ET LABORATOIRES

1. Ce cours est d'une durée de 90 périodes : 45 périodes de théorie (3 périodes par semaine) et 45 périodes de laboratoire (3 périodes par semaine) selon l'horaire en vigueur.
2. La présence au cours constitue une condition déterminante de la réussite. Elle permet de placer l'étudiante et l'étudiant en situation d'apprentissage supervisé par le professeur. La présence au cours est nécessaire à l'atteinte des objectifs, elle est donc obligatoire.
3. La ponctualité est de rigueur, le professeur peut refuser l'accès en classe à une personne qui s'y présente après un retard indu.
4. Une personne absente sans raison valable ne recevra pas d'aide supplémentaire de la part du professeur sur la matière manquée. Il est de la responsabilité de l'étudiante et de l'étudiant de voir à reprendre autrement les activités manquées.
5. Pendant la partie théorique, le professeur expliquera la matière du cours.
6. Pendant les laboratoires, les étudiant·es devront vérifier leur compréhension des notions vues pendant la partie théorique et/ou travailler sur les ateliers.
7. Un·e étudiant·e doit utiliser uniquement les logiciels enseignés dans le cadre de ce cours et doit respecter un climat propice au travail. Dans le cas contraire, le professeur peut lui demander de quitter le local et la personne sera considérée absente.

11.2. STRUCTURE DU COURS

ÉTAPES

Le cours est divisé en 5 étapes. Chaque étape se termine par un examen contenant un volet pratique et un volet théorique.

MODULES

Un module comprend les activités suivantes, **à effectuer le jour où le module est présenté:**

1. Un court exposé théorique
2. Une entrevue formative à effectuer devant le professeur
3. Un mini-test contenant quelques questions théoriques et un retour sur l'entrevue

Chaque module contient aussi un **atelier**, à remettre **au plus tard une semaine après** la présentation du module.

11.3. FILE D'ATTENTE POUR QUESTIONS

Un système de file d'attente permettra de consigner les questions des étudiant.es. Durant les séances de travail en classe, le professeur répondra aux questions dans l'ordre d'entrée dans ce système.

11.4. ÉVALUATIONS

ENTREVUES

L'entrevue est une *évaluation formative* à effectuer devant le professeur. Durant l'entrevue, l'étudiant·e doit démontrer sa compréhension de la théorie en réalisant en expliquant une notion à réutiliser dans l'atelier.

MINI-TESTS

Chaque mini-test contient quelques questions théoriques ainsi qu'un retour sur l'entrevue. Le mini-test vise à valider que la personne étudiante comprend la théorie et est prête à effectuer l'atelier.

ATELIERS

Les ateliers sont des travaux pratiques à réaliser en Java. L'atelier doit être remis via Git. Une pénalité de 10% est appliquée si la remise Git est non-fonctionnelle.

EXAMENS

Les examens contiennent un volet pratique et un volet théorique. Ils sont à réaliser au Collège à la date et dans le local indiqué par le professeur. Il s'agit d'un travail **strictement individuel**.

11.5. TRAVAUX ET EXAMENS

1. Aucun retard ne sera toléré pour les évaluations de la dernière étape. Pour les autres étapes, une pénalité de 10% par jour ouvrable de retard sera calculée.
2. Aucun retard ne sera accepté sans raison valable après cinq jours de retard.
3. Lorsque le professeur doute du travail effectué par un·e étudiant·e, il peut avoir recours à une vérification orale ou écrite du niveau de connaissance de la personne concernée.
4. **Chaque personne est responsable de vérifier la qualité de ses fichiers et de ses imprimés remis dans le cadre d'une évaluation.**
5. Les examens seront conservés par le professeur pendant 6 mois et pourront être consultés sur demande.
6. Lors d'un examen, l'étudiant·e doit se présenter au local désigné par le Collège, et réaliser l'examen sous la surveillance d'un membre du personnel.
7. Dans le cas d'une absence à un examen, la personne devra fournir, la journée même, une justification pour motiver son absence auprès du professeur. Si la raison de l'absence n'est pas valable ou qu'elle n'est pas justifiée la journée même, alors l'étudiante ou l'étudiant se verra attribuer la note zéro pour l'examen concerné.

8. Tel que mentionné dans la politique institutionnelle d'évaluation des apprentissages, 10% de la note de chaque activité demandant de produire un document écrit fera référence à la qualité de la langue.

12. INTÉGRITÉ INTELLECTUELLE

1. Tel que mentionné dans la politique institutionnelle d'évaluation des apprentissages, tout manquement à l'intégrité intellectuelle (plagiat, fraude, tentative ou collaboration à l'un ou l'autre de ces événements) entraîne la note zéro pour l'activité d'évaluation en question et un rapport d'événement sera remis au comité départemental de plagiat et à la direction.
2. En cas de récidive dans le même cours, l'étudiant·e se verra attribuer la mention échec pour le cours concerné.

13. FAUTE MÉTHODOLOGIQUE

3. Tel que mentionné dans la politique institutionnelle d'évaluation des apprentissages, toute faute méthodologique (p.ex. ne pas citer la source d'un extrait de code tiré du Web) peut entraîner une pénalité d'au plus 10% pour l'évaluation concernée. Un rapport sera transmis au comité départemental de plagiat.
4. En cas de récidive, le comité pourrait recommander de traiter l'événement comme un manquement à l'intégrité intellectuelle.

14. ACTIVITÉS D'ÉVALUATION FORMATIVES

Catégorie	Activité	Portera sur	Date
Entrevues	Étape 1 : 3 entrevues	Objectifs 1.4,1.5,3	Séances 1.1,1.2,2.1
	Étape 2 : 3 entrevues	Objectifs 1.5,2.1,2.2,2.6,3	Séances 4.1,4.2,5.1
	Étape 3 : 3 entrevues	Objectifs 1.2,3	Séances 6.1,6.2,7.1
	Étape 4 : 3 entrevues	Objectifs 1.3,2.1,2.2,3	Séances 9.1,9.2,10.1
	Étape 5 : 3 entrevues	Objectifs 2.3,2.4,2.5,3	Séances 12.1,12.2,13.1

15. ACTIVITÉS D'ÉVALUATION SOMMATIVES

Le barème de correction peut être modifié durant la session.

Catégorie	Évaluations	Portera sur	Date limite	Poids
Mini-tests (25%)	Étape 1 : 3 mini-tests	Objectifs 1.4,1.5,3	Séances 1.1,1.2,2.1	5%
	Étape 2 : 3 mini-tests	Objectifs 1.5,2.1,2.2,2.6,3	Séances 4.1,4.2,5.1	5%
	Étape 3 : 3 mini-tests	Objectifs 1.2,3	Séances 6.1,6.2,7.1	5%
	Étape 4 : 3 mini-tests	Objectifs 1.3,2.1,2.2,3	Séances 9.1,9.2,10.1	5%
	Étape 5 : 3 mini-tests	Objectifs 2.3,2.4,2.5,3	Séances 12.1,12.2,13.1	5%
Ateliers	Étape 1 : 3 ateliers	Objectif 1.4,1.5,3	Séances 2.1,2.2,3.1	5%

(25%)	<i>Étape 2</i> : 3 ateliers	Objectif 1.5,2.1,2.2,2.6,3	Séances 5.1,5.2,6.1	5%
	<i>Étape 3</i> : 3 ateliers	Objectifs 1.2,3	Séances 7.1,7.2,8.1	5%
	<i>Étape 4</i> : 3 ateliers	Objectifs 1.3,2.1,2.2,3	Séances 10.1,10.2,11.1	5%
	<i>Étape 5</i> : 3 ateliers	Objectifs 2.3,2.4,2.5,3	Séances 13.1,13.2,14.1	5%
Examens (25%)	<i>Examen 1</i>	Objectif 1.4,1.5,3-4	Séance 3.2	10%
	<i>Examen 2</i>	Objectif 1.5,2.1,2.2,2.6,3	Séance 6.2	10%
	<i>Examen 3</i>	Objectifs 1.2,3	Séance 8.2	10%
	<i>Examen 4</i>	Objectifs 1.3,2.1,2.2,3	Séance 9.2	10%
	<i>Examen 5</i>	Objectifs 2.3,2.4,2.5,3	Séance 15.2	10%

16. MATÉRIEL REQUIS

- Matériel pour écrire **à la main**, p.ex. papier/crayon ou tablette/stylet
- Des écouteurs pour visionner les vidéos du cours
- Un compte GitLab (<https://gitlab.com>)
- Un disque dur externe **clairement identifié.**
- *Optionel* : un paquet de carte à jouer