

PLAN DE COURS

| |
|-------------------------------------|
| Programmation orientée objet |
|-------------------------------------|

Titre du cours

| |
|-------------------------------------|
| Techniques de l'informatique |
|-------------------------------------|

Nom du ou des programme(s) ou de la composante de formation générale

| |
|-------------------------------------|
| Techniques de l'informatique |
|-------------------------------------|

Discipline

| | | |
|-------------------|--------------|-------------|
| 420-210-MV | 2-3-2 | 2,33 |
|-------------------|--------------|-------------|

Numéro du cours

Pondération

Unités

| | | |
|-----------------------|--------------|----------------------------------|
| Samuel Fostiné | S-016 | samuel.fostine@cegepmv.ca |
|-----------------------|--------------|----------------------------------|

Enseignant.e

Numéro de bureau

Poste téléphonique et courriel

| | |
|---------------------|-----------------------|
| Informatique | Olivier Tardif |
|---------------------|-----------------------|

Nom du département

Nom du coordonnateur ou de la
coordonnatrice du département

| | |
|------------------|-------------------|
| 2025-2026 | Hiver 2026 |
|------------------|-------------------|

Année scolaire

Trimestre

PRÉSENTATION GÉNÉRALE DU COURS :

Ce cours a pour préalable le cours Introduction à la programmation informatique et est préalable au cours Structures de données, Architecture de logiciel, Développement de jeux vidéo et Développement d'application mobile. Il est le deuxième cours dans l'axe de formation Fondements de la programmation, qui contient aussi les cours Introduction à la programmation informatique et Structures de données.

Dans ce cours, la personne étudiante apprend à résoudre des problèmes en utilisant le paradigme de programmation le plus utilisé dans les grands projets de l'industrie. À la fin du cours, la personne étudiante devra bien maîtriser les principaux concepts de la programmation orientée objet et devra être prête à les appliquer à différents contextes.

Les habiletés de base en programmation développées lors du cours Introduction à la programmation informatique sont mobilisées par la personne étudiante afin de concevoir des programmes plus complexes suivant le paradigme de programmation orientée objet.

Le concept de classe sert dans le cours Structures de données, dans lequel la personne étudiante doit modéliser des structures de données en utilisant des objets. L'utilisation de fonctionnalités avancées des outils de contrôle de version y est aussi enseignée.

ÉNONCÉ DE LA COMPÉTENCE :

La compétence visée dans le cours est la compétence **00Q6** : *Exploiter les principes de la programmation orientée objet.*

La compétence sollicitée dans le cours est la compétence **00Q2** : *Utiliser des langages de programmation.*

CIBLE D'APPRENTISSAGE :

Concevoir une application en utilisant le paradigme de programmation orientée objet.

ÉTAPES PROGRESSIVES D'APPRENTISSAGE ET ÉVALUATION DES APPRENTISSAGES :

1. Contenus d'apprentissage

Il est important de noter que les éléments suivants sont cités à titre indicatif. Il se pourrait qu'il y ait certains aménagements dictés par le déroulement du cours.

➤ **Environnement Java**

- La machine virtuelle Java (JDK, Open JDK)
- Compilation (la commande javac)
- L'exécution (la commande java)
- La portabilité
- L'API Java
- L'environnement de développement

➤ **Retour rapide sur certains concepts importants du cours d'introduction à la programmation informatique**

- Les structures conditionnelles (if/else if/else)
- Les tableaux
- Les structures de boucles
- Les fonctions

➤ **Concepts propres à la programmation orientée objet**

- Notion de classe
- Notion d'objet
- Constructeur
- Encapsulation des données
- Héritage des classes
- Présentation d'interfaces et de classes abstraites
- Différence entre variables/méthodes d'instances et variables/méthodes statiques
- Diagramme de classe

➤ **L'héritage/Implémentation d'interface/bonne pratique POO**

- Notion d'héritage
- Notion de classe dérivée
- Construction et initialisation des classes dérivées
- Notion de redéfinition
- Polymorphisme
- Classes abstraites, Interfaces, et classes internes

➤ **Exceptions et gestion d'erreur**

- Notion d'exception
- La classe « Exception »
- Le mot clé « throw »
- Gestion de plusieurs exceptions
- Gestion des exceptions
 - Le mot clé « throws »

-
- Le bloc « try catch »
 - Le mot clé « finally »
-
- **Outils de contrôle de version de programme (Git, Mercurial, SVN, etc.)**
 - Utilisation locale d'outils de contrôle de version
 - Présentation de la notion des branches
 - Hébergement d'un projet sur le serveur Github
 - Documentation (README)

 - **Structures de données**
 - Survol des types de structures (listes, piles, arbres, tas, etc.)
 - Exploration de la complexité des algorithmes reliés aux structures de données
 - Références et utilisation du tas (heap) pour la sauvegarde des valeurs, par opposition à la pile (stack)

 - **Librairies d'interfaces graphiques (ex.: JavaFX, Swing)**
 - Installation d'une librairie permettant d'implémenter un GUI
 - Choix des bons éléments pour faire l'entrée/la sortie des données
 - Affichage des informations
 - Modélisation des éléments du GUI comme des objets
 - Utilisation de l'héritage pour créer ses propres éléments graphiques

 - **Bonnes pratiques de documentation**
 - Documentation des fonctions et classes
 - Rédaction de commentaires explicatifs pertinents
 - Utilisation d'outils de génération de documentation (ex: Javadoc)

2. Évaluation des apprentissages

➤ Évaluations formatives

Afin de fournir une rétroaction continue aux étudiants, chaque semaine, pendant les périodes pratiques, des exercices formatifs en lien avec la matière enseignée sont distribués et l'enseignant donne une rétroaction immédiate aux étudiants pendant ces périodes.

➤ Évaluations sommatives, critères d'évaluation et pondération

L'évaluation sommative est composée de 10 quiz (10%), de 2 examens de 2h chacun pendant la session (30 %), d'un laboratoire (12.5%), d'un projet final (17.5%) et d'une épreuve finale sous la forme d'un examen de 3h (30 %).

Les examens sont à réaliser en présentiel. Chaque examen (sauf l'épreuve finale) est composé de 2 parties :

- Une partie théorique d'une durée approximative égale à 1/3 de la durée de l'évaluation. Cette partie est à réaliser à livres et ordinateurs fermés.
- Une partie pratique d'une durée approximative égale à 2/3 de la durée de l'évaluation. Cette partie de l'examen est à réaliser sur les ordinateurs des laboratoires seulement. Mis à part l'accès à LÉA, aucun accès internet ne sera autorisé.

Le laboratoire est à réaliser en présentiel dans un laboratoire informatique. La personne étudiante doit implémenter un programme avec plusieurs classes en utilisant les notions apprises dans le cours.

Le projet final est à réaliser individuellement ou en équipe de 2 étudiants maximum. Il s'étend sur plusieurs semaines. À la fin du projet, les personnes étudiantes doivent rencontrer l'enseignant pour que ce dernier puisse les questionner par rapport au code.

Tableau des évaluations

| Type d'évaluation | Pondération |
|--|--------------|
| 10 quiz de révision sur Wooclap (8 meilleures notes, 1.25% chaque) | 10% |
| Examen 1 | 15% |
| Laboratoire | 10 % |
| Examen 2 | 20% |
| Projet final accompagné d'une entrevue de code | 15% |
| Examen final | 30 % |
| Total des évaluations | 100 % |

Épreuve finale

- **Type de l'épreuve**

Au cours d'une séance d'examen dans un laboratoire d'informatique, la personne étudiante doit concevoir une application capable d'effectuer quelques tâches prédéterminées, et dont le code contient plusieurs classes modélisant des concepts et actions différentes du processus. Cette épreuve sera réalisée de manière individuelle.

Les algorithmes présents dans le projet doivent être simples, l'utilisation de fonctions de la librairie standard (ou équivalent) du langage étant permise.

- **Critères d'évaluation de la performance de l'étudiant**

- Respect rigoureux des consignes
- Le code de l'application devra contenir les éléments suivants :
 - Au moins une classe de données
 - Au moins une classe ayant plusieurs méthodes associées
 - Une classe utilisant l'héritage
- Adéquation juste entre les concepts importants de l'application et leur représentation en classes
- Application judicieuse des principes d'encapsulation et d'héritage
- Choix approprié des instructions et des algorithmes
- Organisation logique des instructions et lisibilité du code
- Repérage complet des erreurs et fonctionnement correct du programme

Veuillez noter que l'échéancier suivant dépendra de l'avancement du cours. Il peut donc varier durant la session.

| <u>CALENDRIER SYNTHÈSE</u> | | | |
|----------------------------|--|--|-------------------------------------|
| Semaine d'enseignement | Nature et date de remise des évaluations | Points alloués | Autres informations (s'il y a lieu) |
| Semaine 1 | <ul style="list-style-type: none"> Présentation du plan de cours Introduction à l'environnement Java Retour sur les concepts de base en programmation – Les structures conditionnelles/boucles/tableaux | | |
| Semaine 2 | <ul style="list-style-type: none"> Quiz 1 Retour sur les concepts de base en programmation – Les fonctions Conversion de types | Quiz 1 (1.25%) | |
| Semaine 3 | <ul style="list-style-type: none"> Quiz 2 Les classes et les objets Représentation graphique d'une classe (diagramme de classe UML) | Quiz 2 (1.25%) | |
| Semaine 4 | <ul style="list-style-type: none"> Quiz 3 Les classes et les objets – Continuation Encapsulation des données | Quiz 3 (1.25%) | |
| Semaine 5 | <ul style="list-style-type: none"> Quiz 4 Révision Examen 1 | Quiz 4 (1.25%) Examen 1 (15%) | |
| Semaine 6 | <ul style="list-style-type: none"> Notion d'héritage Représentation graphique de l'héritage | | |
| Semaine 7 | <ul style="list-style-type: none"> Quiz 5 Polymorphisme : Surcharge et Redéfinition | Quiz 5 (1.25%) | |
| Semaine 8 | <ul style="list-style-type: none"> Laboratoire | Laboratoire (10%) | |

CALENDRIER SYNTHÈSE

| Semaine d'enseignement | Nature et date de remise des évaluations | Points alloués | Autres informations (s'il y a lieu) |
|------------------------|--|---|-------------------------------------|
| | <ul style="list-style-type: none"> ▪ Polymorphisme : Surcharge et Redéfinition - Continuation ▪ Interface et classes abstraites | | |
| Semaine 9 | <ul style="list-style-type: none"> ▪ Quiz 6 ▪ Outils de contrôle de version de programme ▪ Introduction au Framework JavaFX | Quiz 6 (1.25%) | |
| Semaine 10 | <ul style="list-style-type: none"> ▪ Quiz 7 ▪ Présentation du projet final ▪ Framework Java FX - Continuation ▪ Découpage d'un problème en classes ▪ Révision | Quiz 7 (1.25%) | |
| Semaine 11 | <ul style="list-style-type: none"> ▪ Examen 2 ▪ Lecture et écriture de fichiers ▪ Notion du fichier JSON | Examen 2 (20%) | |
| Semaine 12 | <ul style="list-style-type: none"> ▪ Quiz 8 ▪ Fichier JSON – Continuation ▪ Travail en classe sur le projet | Quiz 8 (1.25%) | |
| Semaine 13 | <ul style="list-style-type: none"> ▪ Quiz 9 ▪ Documentation Javadoc ▪ Travaux pratiques en classe / résolution de problème ▪ Continuation du projet | Quiz 9 (1.25%) | |
| Semaine 14 | <ul style="list-style-type: none"> ▪ Quiz 10 ▪ Révision ▪ Remise projet final et entrevue de code | Quiz 10 (1.25%) Remise du projet final (15%) | |
| Semaine 15 | Examen final | Examen final (30%) | |

EXIGENCES PARTICULIÈRES DU COURS :

RÈGLES INSTITUTIONNELLES ET DÉPARTEMENTALES :

Usage du cellulaire en classe

L'utilisation du téléphone cellulaire et autres appareils électroniques apparentés sont formellement interdits durant les cours. Tout contrevenant pourra être expulsé sans préavis. Ces appareils doivent être rangés hors de vue pour toute la durée des séances de cours.

Présence aux évaluations sommatives

Conformément à l'article 4.4.1 de la PIEA:

La présence à une évaluation sommative est obligatoire. L'étudiant qui s'absente, sans motif grave à l'appui, reçoit la note zéro. C'est à l'étudiant qu'il revient d'aviser son professeur des motifs de son absence dans le plus bref délai et de lui fournir, s'il y a lieu, une pièce justificative. Seul un motif grave (ex. mortalité, accident ou maladie) peut être reconnu comme valable par le professeur. Dans un tel cas, selon la nature de l'évaluation, le professeur proposera à l'étudiant une modalité de récupération.

Lors d'un examen, l'étudiant doit se présenter au moment et à l'endroit prévus. S'il arrive en retard et qu'un autre étudiant a déjà terminé et quitté la salle, l'accès lui est refusé, à moins que la nature de l'évaluation le permette.

Remise des travaux

Conformément avec l'article 4.4.2 de la PIEA:

Dans le cas d'un travail, le professeur détermine les modalités de remise, à savoir le lieu et le support (version électronique, version imprimée ou document original). Tout travail qui ne respecte pas ces modalités pourra être refusé. Le professeur détermine également la date et le moment de la remise du travail. L'étudiant qui remet son travail en retard se verra, sauf dans des situations jugées exceptionnelles par le professeur, attribuer une pénalité de 10% de la pondération prévue au départ de ce travail, par jour ouvrable, à compter du jour et de l'heure de la remise du travail.

Par ailleurs, un travail qui n'est pas remis à temps peut être refusé à compter du moment où le professeur utilisera le contenu de ce travail dans le cadre de son cours, ou qu'il sera requis pour poursuivre un travail en équipe. Une telle condition pédagogique doit être indiquée à l'avance aux étudiants, avec les consignes du travail.

Tout travail remis au professeur après que les étudiants ont reçu leurs travaux corrigés est refusé. Seul le professeur, s'il le juge à propos, peut proposer un autre travail et accorder un délai.

Dans tous les cas où le type de travail le permet, l'étudiant doit conserver un brouillon, un fichier électronique ou une photocopie de son travail.

Correction du Français

Conformément avec l'article 4.6.2 de la PIEA:

Dans les productions écrites (examens, travaux, projets), la correction du français est obligatoire et elle constitue une pénalité jusqu'à concurrence de 10% de la note. Pour établir cette pénalité, les productions écrites sont corrigées à l'aide d'une grille à échelle descriptive, selon le type de travail exigé.

Présence aux cours

Conformément avec l'article 4.7.1 de la PIEA:

L'étudiant a la responsabilité d'assister aux cours, car ils constituent le premier lieu d'apprentissage. L'étudiant a aussi la responsabilité d'adopter un comportement qui favorise l'apprentissage. Il respecte l'horaire prévu de même que le temps de pause. L'étudiant qui ne respecte pas ces exigences pourra subir les sanctions prévues au Règlement relatif aux conditions de vie au Cégep Marie Victorin (Règlement numéro 9). Les professeurs sont invités à inscrire la présence des étudiants à leur cours. En cas d'absence, l'étudiant doit récupérer par lui-même les apprentissages selon les modalités particulières inscrites au plan de cours, ainsi que les documents ou le matériel manquants.

Retard aux cours

En cas de retard de 15 minutes ou plus, le professeur pourra refuser l'accès au local jusqu'à la pause.

Plagiat ou fraude

Toute tentative de plagiat ou de fraude entraînera une note de 0 pour l'évaluation concernée.

Selon l'article 4.9.1 de la PIEA:

Le plagiat se définit comme l'acte de faire passer pour sien un contenu ou une production d'autrui sans en identifier la source. Commet un plagiat l'étudiant qui par exemple :

- Recopie un extrait d'un texte sans utiliser les normes de citation.
- S'approprie l'idée ou le texte d'un auteur en le paraphrasant incorrectement ou en omettant d'utiliser les normes de citation.
- Utilise un concept, une image ou une musique sans en indiquer la source.

La fraude se définit comme l'acte de tromper dans le but d'en tirer un avantage personnel. Commet une fraude l'étudiant qui par exemple:

- Utilise un autre matériel que celui qui est autorisé, incluant le matériel qu'il a produit dans une évaluation pour un autre cours.
- Copie le travail ou les réponses d'examen d'une autre personne.

-
- Aide une autre personne à copier.
 - Participe au vol, à la falsification de données, de document ou de matériel reliés à une évaluation ou à la justification d'une absence lors d'une évaluation (par exemple, un papier de médecin).
 - Utilise de l'aide non permise pour réaliser un travail.

Réussite du cours

Conformément avec l'article 4.3.1 de la PIEA:

Le seuil de réussite Au terme du cours, la note doit exprimer le degré de développement de la ou des compétences visées. Tel que spécifié dans le Règlement sur le régime des études collégiales (RREC), le seuil de réussite d'un cours est fixé à 60% ; la réussite du cours entraîne l'obtention de l'unité ou des unités attachées à ce cours. L'assemblée départementale ou le conseiller ou la conseillère pédagogique à la Formation continue peut définir les modalités et les conditions de réussite d'un cours par exemple, dans la situation où la démonstration de l'atteinte de la cible d'apprentissage, lors de l'évaluation sommative finale, serait faite malgré l'insuffisance des résultats antérieurs. Ces modalités et ces conditions de réussite devront être approuvées par la Commission des études.

REFERENCE POUR CONSULTER LA POLITIQUE COMPLETE :

La Politique institutionnelle d'évaluation des apprentissages (PIEA) se trouve à l'adresse suivante sur le site internet du collège et est accessible en tout:

https://www.collegemv.qc.ca/sn_uploads/fck/a_propos_du_cegep/documents_officiels/politiques_et_reglements/PIEA-amendee-2021-12-14_2.pdf

MÉDIAGRAPHIE:

Bloch, J. (2018). Effective Java (Third edition). Addison-Wesley.

Downey, A., & Mayfield, C. (2019). Think Java : How to think like a computer scientist (Second edition). O'Reilly Media, Inc.

Java Documentation. (s. d.). Oracle Help Center. Consulté 15 février 2022, à l'adresse <https://docs.oracle.com/en/java/index.html>

Phillips, D. (2018). Python 3 Object-Oriented Programming—Third Edition. Packt Publishing. <https://www.safaribooksonline.com/library/view/title/9781789615852/?ar?orpq&email=^u>

Python 3.10.2 Documentation. (s. d.). Consulté 15 février 2022, à l'adresse <https://docs.python.org/3/>
Sierra, K., & Bates, B. (2005). Head first Java (2nd ed). O'Reilly.