

**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**

Numéro du cours

**2-3-2**

Pondération

**2,33**

Unités

**Samuel Fostiné**

Enseignant

**S-012**

Numéro de bureau

**samuel.fostine@collegemv.qc.ca**

Poste téléphonique et courriel

**Informatique**

Nom du département

**Olivier Tardif**Nom du coordonnateur ou de la  
coordonnatrice du département**2023-2024**

Année scolaire

**Hiver 2024**

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 Structure 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.

La personne étudiante apprend à coder avec un langage de programmation différent de celui utilisé dans le cours Introduction à la programmation informatique, lui permettant de voir comment les mêmes idées se retrouvent sous des formes différentes dans des langages différents. Elle pourra ainsi plus rapidement apprendre les différents langages de programmation qu'elle retrouvera sur son parcours.

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**

- Historique du langage
- La machine virtuelle Java (JDK, Open JDK)
- Compilation (la commande javac)
- L'exécution (la commande java)
- La portabilité
- L'API Java
- L'environnement du développement Eclipse
- L'environnement du développement IntelliJ
- Processus d'acquisition de la licence académique IntelliJ

#### ➤ **Les classes et les objets**

- Notion d'objet
- Notion de classe
- La notion de constructeur
- Initialisation d'un objet
- Champs et méthodes de classe
- Les droits d'accès aux attributs et méthodes d'une classe

#### ➤ **Les types primitifs et les opérateurs**

- Les entiers
- Les types flottants
- Type caractère
- Type booléen
- Opérateurs arithmétiques
- Opérateurs logiques
- Opérateurs d'incrément et décrémentation
- Transtypage de types primitifs et d'objets

#### ➤ **Les instructions de contrôle**

- L'instruction « if »
- L'instruction « switch »
- L'instruction « do ... while »
- L'instruction « for »
- L'instruction « break »

#### ➤ **L'héritage**

- Notion d'héritage
- Notion de classe dérivée
- Construction et initialisation des classes dérivées

- 
- Notion de redéfinition
  - Encapsulation des données
  - Polymorphisme
  - Classes abstraites, Interfaces, et classes internes
  - Méthode et passage de paramètre
  - Différence entre variables/méthodes d'instances et variables/méthodes statiques
  - Transtypage de types primitifs et d'objets
  - Diagramme de classe

➤ **Les exceptions et les assertions**

- 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 »

➤ **Notions de tests unitaires :**

- Écriture des tests unitaires
- Les techniques simples de tests unitaires

➤ **Java et la programmation graphique**

- Les composants graphiques Java
- La librairie JavaFX ou Swing

➤ **Gestion du code source**

- Le but d'un gestionnaire de source Git
- Le modèle centralisé Vs le modèle distribué
- Le fonctionnement de Git
- Les librairies externes
- Les Java JARs

---

## 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.

De plus, le corrigé de ces exercices est expliqué en classe et distribué aux étudiants.

### ➤ Évaluations sommatives, critères d'évaluation et pondération

L'évaluation sommative est composée d'un travail pratique (10%), de 2 laboratoires (10%) d'un projet final (25 %), d'un examen de mi-session (15 %), de 10 quiz (10%) et d'une épreuve finale sous la forme d'un examen (30 %).

Toutes les évaluations sont à réaliser individuellement.

Les examens sont à réaliser en présentiel (laboratoire). Chaque examen est composé de 2 parties :

- Une partie théorique.
- Une partie pratique. Cette partie de l'examen est à réaliser sur les ordinateurs des laboratoires seulement.

Tableau des évaluations

Type d'évaluation	Pondération
Travail pratique 1	10 %
Laboratoire 1	5%
Laboratoire 2	5%
Projet final	25 %
<b>Total des travaux pratiques</b>	<b>45 %</b>
10 quiz (1% chaque)	10 %
Examen 1	15 %
<b>Examen 2 (Épreuve finale)</b>	<b>30 %</b>
<b>Total des examens (présentiel)</b>	<b>55 %</b>
<b>Total des évaluations</b>	<b>100 %</b>

---

## 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"> <li>Présentation du plan de cours</li> <li>Introduction à l'environnement Java</li> <li>Création d'un projet et d'une classe Java</li> <li>Explication de la méthode main</li> <li>Variables et les types primitifs en Java</li> </ul>		
Semaine 2	<ul style="list-style-type: none"> <li>Les types primitifs et les opérateurs</li> <li>La classe String</li> <li>Les instructions de contrôle</li> <li>Boucles</li> <li>Les types énumérés</li> <li>Conversion de types</li> </ul>		
Semaine 3	<ul style="list-style-type: none"> <li><b>Quiz 1</b></li> <li>Les tableaux</li> <li>Notion d'objet et de classe</li> </ul>	Quiz 1 : 1%	
Semaine 4	<ul style="list-style-type: none"> <li><b>Quiz 2</b></li> <li>Notion d'objet et de classe – Continuation</li> <li><b>Travail pratique 1</b></li> </ul>	Quiz 2 : 1% TP 1 : 10%	Distribution de l'énoncé du TP-1
Semaine 5	<ul style="list-style-type: none"> <li><b>Quiz 3</b></li> <li><b>Remise du TP1</b></li> <li>Correction du TP1</li> <li>Révision pour l'examen 1</li> </ul>	Quiz 3 : 1%	
<b>Semaine 6</b>	<b>Examen 1</b>	<b>15 %</b>	
Semaine 6/7	<b>Quiz 4</b> (semaine 7)	Quiz 4 : 1%	

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"> <li>▪ Notion d'héritage</li> <li>▪ Notion de classe dérivée</li> <li>▪ Encapsulation des données</li> <li>▪ Polymorphisme</li> <li>▪ Représentation graphique de l'héritage</li> </ul>		
Semaine 8	<ul style="list-style-type: none"> <li>▪ <b>Quiz 5</b></li> <li>▪ Les packages</li> <li>▪ Notion d'exception</li> <li>▪ La notion d'annotation</li> <li>▪ <b>Laboratoire 1</b></li> </ul>	Quiz 5 : 1%  Laboratoire 1 : 5%	Le laboratoire 1 va avoir lieu en classe. Une démonstration doit être faite au prof avant la fin du laboratoire.
Semaine 9	<ul style="list-style-type: none"> <li>▪ <b>Quiz 6</b></li> <li>▪ Le gestionnaire de source Git</li> <li>▪ Le modèle centralisé Vs le modèle distribué</li> <li>▪ La notion de JAR Java</li> <li>▪ Technique d'ajout de bibliothèques externes</li> <li>▪ Les Apache Commons</li> </ul>	Quiz 6 : 1%	
Semaine 10	<ul style="list-style-type: none"> <li>▪ <b>Quiz 7</b></li> <li>▪ Les collections Java</li> <li>▪ Présentation de la bibliothèque JavaFX ou Swing</li> <li>▪ Création d'équipe pour le projet</li> <li>▪ <b>Distribution des consignes du projet final – En équipe</b></li> </ul>	Quiz 7 : 1%	Distribution de l'énoncé du projet final
Semaine 11	<ul style="list-style-type: none"> <li>▪ <b>Quiz 8</b></li> <li>▪ <b>Laboratoire 2</b></li> <li>▪ La bibliothèque JavaFX – Continuation</li> <li>▪ La communication entre les composants JavaFX</li> <li>▪ Notion FXML</li> <li>▪ Les animations JavaFX</li> </ul>	Quiz 8 : 1% Laboratoire 2 : 5%	Le laboratoire 2 va avoir lieu en classe. Une démonstration doit être faite au prof avant la fin du laboratoire.



---

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"><li>Noter bien, si on utilise Swing au lieu de JavaFX, on va couvrir les notions de Swing de préférence.</li></ul>		
Semaine 12	<ul style="list-style-type: none"><li><b>Quiz 9</b></li><li>Lecture et écriture dans des fichiers</li><li>Notion du fichier JSON</li></ul>	Quiz 9 : 1%	
Semaine 13	<ul style="list-style-type: none"><li><b>Quiz 10</b></li><li>Les relations entre objets et leurs représentations graphiques</li><li>Les tests unitaires</li><li>Écriture des tests unitaires simples</li></ul>	Quiz 10 : 1%	
Semaine 14	<ul style="list-style-type: none"><li>Révision</li><li><b>Remise du projet final</b></li></ul>		
<b>Semaine 15</b>	<b>Examen final</b>	<b>30%</b>	

---

## **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.

---

#### 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](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.