

## PLAN DE COURS

Introduction à la programmation informatique
--

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

420-111-MV	2-3-3	2,66
------------	-------	------

Numéro du cours

2-3-3

Pondération

2,66

Unités

Samuel Fostiné	S-012	<a href="mailto:samuel.fostine@cegepmv.ca">samuel.fostine@cegepmv.ca</a>
----------------	-------	--

Enseignant-e

S-012

Numéro de bureau

[samuel.fostine@cegepmv.ca](mailto:samuel.fostine@cegepmv.ca)

Poste téléphonique et courriel

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

Nom du département

Olivier Tardif

Nom du coordonnateur ou de la  
coordonnatrice du département

2025-2026	Automne 2025
-----------	--------------

Année scolaire

Automne 2025

Trimestre

---

## PRÉSENTATION GÉNÉRALE DU COURS :

Ce cours se veut une introduction au monde de la programmation et à la résolution de problèmes avec des ordinateurs. Il n'est pas nécessaire d'avoir déjà programmé pour y participer: il ne suffit que d'avoir de la curiosité et un désir d'apprendre de nouvelles choses.

À la fin du cours, les étudiants et étudiantes seront capables de concevoir des applications simples en utilisant le langage Java, un des plus populaires. Ce langage est très utilisé en entreprise pour sa portabilité ainsi que son très grand écosystème de code existant. Vous le reverrez à plusieurs reprises dans votre parcours au cégep.

Pour atteindre cet objectif, il faudra étudier les types, les fonctions, les structures de contrôle, l'entrée et la sortie d'information, les collections et le débogage. Toutes ces notions seront vues à travers des exercices pratiques de programmation.

La meilleure façon d'apprendre la programmation est de programmer, et c'est ce que nous passerons la majorité de notre temps à faire cette session!

Ce cours est un préalable pour les cours *Programmation orientée objet*, *Applications Web* et *Programmation de plateformes embarquées*.

---

## **CIBLE D'APPRENTISSAGE :**

Utiliser un langage de programmation impératif et procédural dans le cadre du développement d'une application simple.

La compétence reliée à ce cours est 00Q2: « Utiliser des langages de programmation ».

## **ÉTAPES PROGRESSIVES D'APPRENTISSAGE ET ÉVALUATIONS DES APPRENTISSAGES :**

Chaque unité d'apprentissage commencera par l'énoncé d'un problème de programmation à résoudre, suivi d'une présentation de quelques minutes sur les notions reliées et/ou une démonstration indiquant comment attaquer le problème. Du temps est ensuite alloué pour mettre les notions vues en pratique et résoudre le problème de programmation dans une série d'exercices formatifs.

Des périodes entières sont parfois dédiées à la résolution de séries d'exercices de programmation. La compréhension des concepts est vérifiée régulièrement par des devoirs sommatifs. Un travail pratique plus demandant doit être réalisé dans les dernières semaines de la session.

Les éléments de contenu vus au cours de la session sont les suivants:

- **Utilisation d'un langage de programmation de manière impérative et procédurale**
- **Différenciation et manipulation des différents types du langage**
  - Nombres entiers, nombres flottants, chaînes de caractères, booléens, etc.
  - Opérateurs et priorité des opérations
  - Tableaux
- **Syntaxe et vocabulaire (mot-clé) du langage**
- **Structures de contrôles**
  - Structures conditionnelles (*if..else, switch*)
  - Structures de boucle (*while, for, dowhile*)
  - Conditions d'arrêt
  - Identification des boucles infinies
- **Développement et utilisation des fonctions**
  - Types de retour
  - Conventions de noms
  - Paramètres (données) d'entrée
  - Principaux types de passage de paramètres (par référence, par valeur)
- **Préparation du développement**
  - Planification des fonctions (nom des fonctions)
  - Utilisation du pseudocode pour planifier le déroulement d'un algorithme ou d'un programme
  - Calcul des expressions mathématiques (utilisation de bibliothèques de fonctions mathématiques)
- **Mécanismes de communication de l'application** (entrée/sortie en mode console, etc.)
- **Groupement des données apparentées dans des structures telles que tableaux et listes.**

- 
- **Environnement de développement**
  - Utilisation des ressources de l'environnement de développement (ex: autocomplétion)
  - Compilation, exécution ou interprétation du code produit
  - **Respect des normes** (*camelCase*, *snake\_case*, espace blanc, etc.)
  - **Erreurs de programmation communes**

La table suivante détaille le nombre et le type d'évaluations sommatives:

Type d'épreuve	Pondération
12 quiz de révision (sur Wooclap)	10% (1% chacun – les 10 meilleures notes)
3 laboratoires	35% (10% les deux premiers et 15% le dernier)
Travail pratique	5% (travail pratique sur Scratch)
Examen 1	20%
Examen final	30%

### Travail pratique et laboratoires

Réalisé individuellement, il permet aux étudiants de se familiariser et d'explorer les notions de base d'algorithme en utilisant Scratch. Le but est d'apprendre à fournir des instructions claires à l'ordinateur permettant de réaliser une petite tâche.

### Modalités de l'examen 1

Vers la mi-session, les étudiants auront à réaliser un examen en classe permettant de vérifier que la compétence est en voie d'être acquise. L'examen demande à la personne étudiante de:

- Répondre à des questions théoriques sur ces concepts de la programmation
- Prédire le comportement de code fourni
- Écrire du code simple pour satisfaire un énoncé
- Trouver des erreurs dans du code fourni

Les réponses aux questions sont évaluées selon leur exactitude. Le code écrit est évalué selon son exactitude, son organisation et sa clarté.

### Modalités de l'épreuve finale

Au cours d'une séance d'examen de 3h dans un laboratoire d'informatique, la personne étudiante développe une application pouvant effectuer une tâche utile à partir d'un cahier de charge fourni par la personne enseignante. L'application doit être décomposée en quelques fonctions. L'exécution de l'application doit pouvoir suivre plusieurs chemins tout dépendant des entrées de l'utilisateur. Aucun GUI n'est requis dans ce projet. Cette épreuve est réalisée de manière individuelle.

---

**Critères d'évaluation :**

- **Respect rigoureux des consignes**
  - Le code de l'application devra contenir les éléments suivants :
    - Algorithmes simples
    - Organisation du code en fonctions
    - Utilisation des méthodes entrée/sortie du langage
    - Utilisation de plusieurs méthodes de la bibliothèque standard du langage (ou son équivalent)
    - Utilisation de boucles
    - Utilisation de structures conditionnelles
- **Découpage cohérent et efficace du programme**
- **Choix approprié des instructions et des algorithmes**
- **Organisation logique des instructions et lisibilité du code**
- **Repérage complet des erreurs et fonctionnement correct du programme**

CALENDRIER SYNTHÈSE

Semaine d'enseignement	Activités d'apprentissage et évaluations	Points alloués	Autres informations (s'il y a lieu)
Semaine 1	Présentation du plan de cours et du cours Activités brise-glace Création de compte sur Scratch		
Semaine 2	Introduction à l'algorithme avec Scratch		
Semaine 3	<b>Quiz 1</b> Continuation – Scratch <b>Travail pratique – Scratch</b> Introduction à la programmation en Java Les variables et les types – Partie 1	Quiz 1 : 1% Travail pratique – Scratch : 5%	
Semaine 4	<b>Quiz 2</b> Les variables et les types – Partie 2 Entrées d'utilisateurs	Quiz 2 : 1%	
Semaine 5	<b>Quiz 3</b> Conversion de type et arguments Les opérateurs	Quiz 3 : 1%	
Semaine 6	<b>Quiz 4</b> Structures de contrôle conditionnelles <b>Laboratoire 1/ Travaux dirigés</b>	Quiz 4 : 1% Laboratoire 1 : 10%	
Semaine 7	<b>Quiz 5</b> Structure de contrôles conditionnelles - Suite	Quiz 5 : 1%	
Semaine 8	<b>Quiz 6</b> Les tableaux	Quiz 6 : 1%	
Semaine 9	<b>Quiz 7</b> Révision <b>Examen 1</b>	Quiz 7 : 1% Examen 1 : 20%	

---

Semaine 10	<b>Quiz 8</b> Les boucles	Quiz 8 : 1%	
Semaine 11	<b>Quiz 9</b> Les boucles <b>Laboratoire 2</b>	Quiz 9 : 1% Laboratoire 2 : 10%	
Semaine 12	<b>Quiz 10</b> Débogueur et fonctions	Quiz 10 : 1%	
Semaine 13	<b>Quiz 11</b> Gestion d'erreurs/exceptions Les fichiers	Quiz 11 : 1%	
Semaine 14	<b>Quiz 12</b> <b>Laboratoire 3</b>	Quiz 12 : 1% Laboratoire 3 : 15%	
Semaine 15	Révision <b>Examen final (3h)</b>	Examen final: 30%	

Les contenus à voir chaque semaine sont donnés à titre indicatif seulement. Ils peuvent changer en fonction du rythme du groupe.

---

### **EXIGENCES PARTICULIÈRES DU COURS :**

Locaux: laboratoires informatiques de développement d'application

Tous les travaux sont à faire individuellement. Remettre en partie ou en totalité le travail d'un autre constitue du plagiat. Les travaux seront vérifiés avec un logiciel anti-plagiat spécialement adapté à l'analyse de code source.

Les travaux doivent être remis selon les normes demandées afin de faciliter les vérifications automatiques. Ne pas suivre les instructions de remises entraînera une pénalité sur la note.

### **RÈGLES INSTITUTIONNELLES ET DÉPARTEMENTALES :**

#### **Usage du cellulaire et appareils électroniques en classe**

Dans les lieux d'enseignement, l'utilisation d'ordinateurs portables et d'appareils électroniques (téléphones cellulaires, téléavertisseurs, lecteurs audionumériques, agendas électroniques, caméras numériques, assistants numériques personnels etc.) est interdite. Tout contrevenant pourra être expulsé sans préavis. Ces appareils doivent être rangés hors de vue pour toute la durée des séances de cours.

#### ***Enregistrement vocal ou vidéo***

Par ailleurs, les usagers de tels appareils doivent respecter l'intégrité physique et morale des personnes. En conséquence, en tout temps et en tous lieux, il est formellement interdit d'enregistrer, de photographier ou de filmer sans le consentement des individus concernés.

Les modalités d'application de la Politique institutionnelle d'évaluation de l'apprentissage (PIEA) sont rendues disponibles aux étudiants et il appartient à ceux-ci d'en prendre connaissance.

Les articles ici-bas qui font l'objet de modalités particulières d'application font référence à ceux de la PIEA en vigueur disponible sur le portail du Cégep Marin-Victorin.

#### **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 en classe**

Conformément à l'article 4.7.1 de la PIEA, il appartient à l'étudiant :

- De fournir les efforts nécessaires pour atteindre les objectifs du cours.
- D'être présent, à l'heure, à toutes les périodes de cours prévues à son horaire et d'y participer activement. À défaut d'être présent, il doit récupérer par lui-même les apprentissages manqués.
- De respecter 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).
- De respecter les délais de remise des travaux ou, si cela est impossible, il a la responsabilité d'entrer en contact avec son professeur dans les meilleurs délais (l'article 4.4.2 de la PIEA).

Il appartient aussi à l'étudiant:

- De respecter toutes les autres règles prévues à la politique relative à l'utilisation des technologies de l'information et de la communication
- D'utiliser un langage approprié, courtois et professionnel dans ses communications numériques

De plus, il est strictement interdit de boire ou manger dans les laboratoires.

---

## Plagiat ou fraude

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, n papier de médecin).
- Utilise de l'aide non permise pour réaliser un travail.

Tout plagiat, toute tentative de plagiat, toute collaboration à un plagiat entraîne la note zéro « 0 » pour l'évaluation en cause et doit faire l'objet d'un rapport écrit au Service des programmes et du développement pédagogique de la part de l'enseignante ou de l'enseignant.

Une récidive peut entraîner des mesures allant jusqu'au renvoi du Collège de l'étudiante ou de l'étudiant par la Direction des études. Pour en savoir plus sur la façon de citer ses sources dans un travail afin d'éviter le plagiat, consultez la [section suivante](#) du site internet de la [bibliothèque](#) du cégep Marie-Victorin.

---

## MÉDIAGRAPHIE :

### Livres de références

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>

Sierra, K., & Bates, B. (2022). *Head first Java: A brain-friendly guide* (3rd ed.). O'Reilly Media.

### Autres ressources

Bhargava, A. Y. (2016). *Grokking algorithms : An illustrated guide for programmers and other curious people*. Manning.

Cormen, T. H. (Éd.). (2009). *Introduction to algorithms* (3rd ed). MIT Press.

*Python 3.10.2 Documentation*. (s. d.). Consulté 15 février 2022, à l'adresse <https://docs.python.org/3/>

Sweigart, A. (2020). *Automate the boring stuff with Python : Practical programming for total beginners* (2nd edition). No Starch Press.

*W3Schools Free Online Web Tutorials*. (s. d.). Consulté 15 février 2022, à l'adresse <https://www.w3schools.com/>