

PLAN DE COURS

Programmation en sciences

Titre du cours

Techniques de l'informatique

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

Informatique

Discipline

420-SN1-RE	1-2-3	2
------------	-------	---

Numéro du cours

1-2-3

Pondération

2

Unités

Maxime Fournier	S-016	maxime.fournier@cegepmv.ca
-----------------	-------	----------------------------

Enseignant-e

S-016

Numéro de bureau

maxime.fournier@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	Hiver
-----------	-------

Année scolaire

Hiver

Trimestre

PRÉSENTATION GÉNÉRALE DU COURS :

Le cours 420-SN1-RE Programmation en sciences est un cours lié à la discipline contributive informatique au programme Sciences de la nature.

Dans ce cours, la personne étudiante est initiée à l'utilisation d'un langage de programmation pour résoudre différents types de problèmes scientifiques en développant des applications en ligne de commande (i.e. sans interface graphique).

Les applications qu'elle apprend à développer lui permettent d'automatiser une suite de calculs, d'en tirer des conclusions et de représenter graphiquement les données pertinentes afin d'en permettre l'analyse par un être humain.

Certaines des applications développées dans ce cours peuvent utiliser des concepts de calcul afin de démontrer une application concrète de l'informatique au processus scientifique.

Les aptitudes de programmation développées dans ce cours pourront être réinvesties dans le cours 360-ESP-MV Projet d'intégration en sciences de la nature, selon les projets réalisés.

La compétence visée : Développer des programmes informatiques en vue d'automatiser la résolution de problèmes dans un contexte scientifique.

Vous devez vous procurer ce livre à la coop du cégep :

HUNEAULT, Sébastien, Olivier, ROUSSEAU et Charles, VARIN. *La programmation en sciences de la nature : Une introduction aux sciences numériques*, 1e éd., Montréal, Les éditions CEC, 364 p.

CIBLE D'APPRENTISSAGE :

Développer une application en ligne de commande permettant d'effectuer les calculs nécessaires pour résoudre un problème scientifique.

ÉTAPES PROGRESSIVES D'APPRENTISSAGE ET ÉVALUATIONS DES APPRENTISSAGES :

Tout au long de la session, l'apprentissage se fera de manière progressive, en alternant théorie, démonstrations et mise en pratique :

- **Présentation des concepts en classe** : chaque nouveau sujet sera introduit par une explication théorique, accompagnée de schémas ou démonstrations visuelles.
- **Exemples et démonstrations de code** : des cas concrets seront présentés en classe (live coding) afin d'illustrer les concepts abordés et de montrer leur application pratique.
- **Beaucoup de pratique** : l'accent sera mis sur les exercices de codage pour consolider les apprentissages. Les étudiants auront de nombreuses occasions de coder eux-mêmes en classe ou à la maison.
- **Évaluations sommatives** :
 1. Deux mini tests sur Moodle de 10% chacun.
 2. Un examen de mi-session 25%
 3. Un projet de 25%
 4. Un examen final en fin de session 30%

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
 - Paramètres (données) d'entrée
 - Découpage d'un problème en fonctions
 - **Mécanismes de communication de l'application** (entrée/sortie en mode console, etc.)
 - **Environnement de développement**
 - Utilisation des ressources de l'environnement de développement (ex : autocomplétion)
 - Compilation, exécution ou interprétation du code produit
 - Débogage avec la console
 - **Respect des normes** (*camelCase*, *snake_case*, espace blanc, etc.)
 - **Utilisation de bibliothèques**
 - Fonctions de la bibliothèque standard
 - Utilisation de fichiers structurés (ex. : CSV)
 - Production de graphiques

Stratégies pédagogiques :

- Enseignement magistral des concepts
- Démonstrations visuelles d'algorithmes
- Exemples de code
- Démonstrations de codage (*live coding*)
- Pratique et exercices de codage
- Utilisation de problèmes connexes aux contenus des cours du programme *Sciences de la nature*, par exemple :
 - ❖ Simulation de modèles de croissance de populations
 - ❖ Calculs de trajectoire
 - ❖ Visualisation et analyse de données statistiques
 - ❖ Dénombrement

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

Type d'épreuve	Pondération
Mini tests (2)	10% chaque
Projet	25%
Examen intra	25%
Examen final	30%

Mini-tests (10% chaque)

Conduits sur Moodle et réalisés individuellement, les mini-tests seront d'une durée d'une heure. Ils seront composés d'un mélange de revue de code, de questions théoriques à court développement et de choix de réponse.

Examen intra (25%)

Cet examen d'une durée de 2h est constituée d'une partie théorique d'une durée de 30-45 minutes et d'une partie pratique d'une durée de 1h15-1h30. Elle portera sur la première partie du cours: programmation générale en sciences de la nature.

Projet (25%)

Réalisé en équipe de 3 en classe et à l'extérieur des heures de cours. Le projet cible plusieurs problématiques et applications de la programmation scientifique dans différents domaines des sciences de la nature.

Étant donné qu'une partie de l'évaluation se déroule hors classe, une présentation orale individuelle est requise pour valider l'acquisition des compétences. La note obtenue lors de cet entretien servira à pondérer celle du travail d'équipe, confirmant ainsi la maîtrise personnelle du projet soumis.

Épreuve finale (30 %) :

Au cours d'une séance d'examen de 3 h dans un laboratoire d'informatique, la personne étudiante développe une application en ligne de commande pouvant résoudre un problème scientifique à partir d'un cahier de charge fourni par la personne enseignante. Elle doit être décomposée en quelques fonctions dont les signatures sont déjà fournies au départ de l'examen. La personne étudiante doit aussi écrire quelques signatures de fonctions visant à étendre la fonctionnalité de l'application, suivant la demande présentée dans le cahier de charge. L'exécution de l'application doit pouvoir suivre plusieurs chemins selon les entrées de l'utilisateur. Aucun interface graphique usager (GUI) n'est requis dans ce projet. Cette épreuve est réalisée de manière individuelle.

CALENDRIER SYNTHÈSE

Semaine d'enseignement	Activités d'apprentissage et évaluations	Points alloués
Semaine 1	Plan de cours Mise en contexte de la programmation (chap 1) Opérations arithmétiques (chap 2.2)	
Semaine 2	Variables (chap 2.3) Saisie de données (chap 2.4)	
Semaine 3	Fonctions (chap 2.5) Règle et convention (chap 2.1)	
Semaine 4	Traitement conditionnel (chap 3.1)	Mini-Tests 1 : 10%
Semaine 5	Boucle (chap 3.2)	
Semaine 6	Collections de données (chap 4.1)	
Semaine 7	Automatisation des opérations de tableaux (chap 4.2) Distribution de l'énoncé et de la correction de l'examen intra formatif	
Semaine 8		Examen Intra : 25%
Semaine 9	Lecture écriture de fichier (chap 4.3) Numpy (chap 4.4)	
Semaine 10	Bibliothèque (chap 5) Matplotlib (chap 6)	
Semaine 11	Statistiques et scipy Mathématiques et sympy	Mini-Tests 2 : 10%
Semaine 12	Applications en sciences	

Semaine 13		Projet En Classe : 25%
Semaine 14		Projet En Classe : 25%
Semaine 15		Examen final: 30%

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

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.

Seuil de réussite

Conformément à l'article 4.3.1 de la PIEA :

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.

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.

Conformément avec l’article 4.7.4 de la PIEA :

L’observation progressive des apprentissages Dans les cours où le travail étudiant doit être observé sur une base régulière afin que le personnel enseignant soit en mesure d’évaluer de manière sommative l’atteinte de la compétence, la présence peut constituer une condition de réussite. Il importe de préciser que cette évaluation doit être mise en œuvre à partir des balises indiquées dans le devis ministériel. Les situations qui se prêtent à l’activation de cet article sont les suivantes :

- Le cours prévoit des heures de travail en atelier dans lesquelles il faut manœuvrer de manière sécuritaire différents types d’appareillage, ce qui nécessite des observations répétées afin de garantir la sécurité sur tous les appareils.
- Le cours prévoit des heures de travail en atelier où la personne étudiante doit démontrer le développement progressif de ses habiletés menant à une production finale.
- Le devis ministériel prévoit de manière explicite la pratique régulière d’une activité pendant les heures contacts.
- Le personnel enseignant n’est pas en mesure d’évaluer, en assurant la sécurité de la personne étudiante, sa pratique d’une activité physique. Celle-ci doit avoir démontré, pour certaines activités physiques et ce, sur une base régulière pendant les heures contacts, sa capacité à pratiquer l’activité de manière sécuritaire. La personne étudiante qui s’absente de manière régulière se verra refuser l’accès aux évaluations sommatives pratiques de cette activité physique pour assurer sa sécurité. Elle obtiendra la note zéro pour cette épreuve. Le département ou le conseiller ou la conseillère pédagogique à la Formation continue aura alors la responsabilité d’approuver cette pratique pour chacun des cours, puis de faire approuver les paramètres d’application par la Commission des études et de s’assurer que ces paramètres sont clairement explicités à l’intérieur de chacun des plans de cours afin que les personnes étudiantes en soient bien informées.

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

HUNEULT, Sébastien, Olivier, ROUSSEAU et Charles, VARIN. *La programmation en sciences de la nature : Une introduction aux sciences numériques*, 1e éd., Montréal, Les éditions CEC, 364 p.

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>

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/>