|  |  |
| --- | --- |
|  | **420-2N6-EM** |
| **HIVER 2023** |
| **Département d’informatique** |

**Plan de cours**

|  |  |
| --- | --- |
| COURS : | **Programmation 2** |

|  |  |  |
| --- | --- | --- |
| PROGRAMME : | 420.B0 | Techniques de l’informatique  420.BB Réseautique |

|  |  |  |
| --- | --- | --- |
| DISCIPLINE : | 420 | Techniques de l’informatique |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PONDÉRATION : | *Théorie :* | 2 | *Pratique :* | 4 | *Étude personnelle :* | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| Professeur du cours | Bureau |  poste |  courriel |
| Pierre-Paul Gallant | E-211 |  | pierre-paul.gallant@cegepmontpetit.ca |

Période de disponibilité aux étudiants

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | LUNDI | MARDI | MERCREDI | JEUDI | VENDREDI |
| Avant-midi |  |  |  | 10h-11h |  |
| Après-midi | 15h-16h | 13h-15h |  |  |  |
| Autre |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Coordonnateurs du département | Bureau |  Poste |  Courriel |
| Dominic Proulx | E-210 | 2061 | coordination.info@cegepmontpetit.ca |
| Vincent Duval | E-210 | 6451 |  |

# Place du cours dans la formation de l’étudiant

|  |
| --- |
| Le cours *420-2N6-EM* *Programmation 2* se situe à la 2e session. Ce cours suit le premier cours de programmation orientée objet *420-1N6-EM* et servira de base au reste de la formation dans le but d’être en mesure de produire des applications orientées objet.  Le cours préalable *420-1N6-EM* Programmation 1, suivi à la première session, a permis à l’étudiant de s’initier à une méthodologie de résolution de problèmes et de se familiariser avec la programmation. Le cours *420-2N6-EM* Programmation 2 permettra d’exploiter davantage la programmation, de créer des interfaces graphiques et d’approfondir la programmation orientée objet.  Les concepts vus dans ce cours seront réinvestis dans le cours *420-2W5-EM* Programmation Web serveur et *420-3N5-EM* Programmation 3 du profil Programmation.  Ce plan de cours doit être conservé par l’étudiant tout au long de ses études, car il sera utile au moment de l’activité d’intégration.  **420-2N6-EM**  Programmation 2  **420-3N5-EM**  Programmation 3  Session 1  Session 3  Session 2  **420-2W5-EM**  Programmation  Web serveur  **420-1N6-EM**  Programmation 1 |

# Compétence(S) du portrait du diplômé

|  |
| --- |
| Programmer des applications (profil programmation)  Programmer des utilitaires (profil réseau) |

# compétence(S) MINISTÉRIELle(S)

|  |  |
| --- | --- |
| **00Q6** | Exploiter les principes de la programmation orientée objet |

# Objectif terminal de cours

|  |
| --- |
| À la fin du cours, l’étudiant sera en mesure de programmer une application à l’aide d’un langage de programmation orienté objet. |

# Orientations pédagogiques

|  |
| --- |
| La répartition des activités d’apprentissage a été réfléchie afin de permettre à l’étudiant d’apprendre et de produire des applications à l’aide de la programmation orientée objet.  Chaque rencontre de 3h aura une partie théorique et une partie pratique.  Partie théorique  La partie théorique introduit les concepts requis pour la réalisation d’exercices et des travaux pratiques. Par des exercices dirigés, l’étudiant apprendra les bases de la programmation et à résoudre des problématiques à l’aide de la programmation orientée objet.  Le professeur explique brièvement un concept de programmation objet à l'aide de démonstrations. L’étudiant complète ensuite des exercices sur la notion présentée. Les exercices simples doivent parfois être remis à la fin du cours.  Partie pratique  La période pratique consiste en une série d’exercices plus complexes en lien avec les concepts vus dans la période théorique et durant les cours précédents. Le professeur sera disponible pour répondra aux questions des étudiants. Le travail doit être remis à la fin du cours. Les étudiants qui n'ont pas terminé les exercices doivent les compléter pendant les heures de travail personnel et effectuer une remise finale lorsqu’ils sont complétés.  Évaluation  Les travaux pratiques devront être remis via la plateforme d’apprentissage en ligne avant l’heure indiquée. Un retard engendrera une pénalité de 10% par jours de retard. Maximum 3 jours après quoi une note de 0 sera attribuée si aucune justification n’est donnée.  Les examens sont réalisés à l'ordinateur et démontrent la maîtrise des concepts et l’habileté à bien programmer. |

# Planification du cours

Le détail de chaque semaine se trouve dans le semainier dans l’annexe, section 15.

|  |  |
| --- | --- |
| **OBJECTIF D’APPRENTISSAGE** | **CONTENU** |
| **PÉRIODE DES ACTIVITÉS : Semaine 1 à 15** | |
| 1. Produire les algorithmes appropriés à la résolution d’un problème | * Identification des données et des résultats attendus du problème * Utilisation des structures de base du langage * Documentation du code et utilisation de standards * Correction des erreurs de syntaxe et de logique * Utilisation approfondie du débogueur |
| **PÉRIODE DES ACTIVITÉS : Semaine 1 à 6** | |
| 1. Comprendre un problème à résoudre et en identifier les principaux éléments | * Transposition des notions du cours de programmation 1 vers Python * Principes du langage Python (indentation, typage dynamique) * Déclaration et utilisation des listes et dictionnaires * Déclaration et utilisation de fichiers Texte et de fichiers CSV * Utilisation des librairies standard |
| 1. Appliquer la procédure liée à la gestion des versions de programmes. | * Création d'un compte utilisateur. * Création d'un dépôt pour un projet. * Ajout et annulation de « commit » et description utile des modifications. * Stratégies de gestion de conflits |
| 1. Exploration des librairies et modules pour la réseautique | * Importation de librairies * Lire et comprendre la documentation * Mettre en pratique les éléments les plus utilisés * Implémentation d’une interface graphique |
| **PÉRIODE DES ACTIVITÉS : Semaines 7 à 13** | |
| 1. Modéliser des classes en définissant correctement leurs membres et méthodes et en utilisant les principes d’encapsulation et d’héritage | * Structure d’une classe * Champs : constants et variables membres * Modificateur d’accès des champs, propriétés et méthodes * Propriétés en lecture et/ou écriture * Constructeurs et autres méthodes propres à un objet * Surcharge des méthodes * Type par énumération * Imbrication d’objets * Interaction entre les objets * Introduction à l’héritage * Construction et initialisation des objets dérivés * Champ et méthodes statiques * Utilisation du polymorphisme |
| **PÉRIODE DES ACTIVITÉS : Semaine 14** | |
| 1. Produire des tests unitaires | * Création d’un test unitaire * Élaboration d’un jeu de tests * Développement piloté par les tests (Test-Driven Development (TDD)) |

# Modalités d’évaluation sommative

## Examens (50%)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mode d’évaluation et description de l’activité** | **Contexte de réalisation** | **Objectif(s) d’apprentissage** | **Critères d’évaluation** | **Échéance  (date)** | **Pondération (%)** |
| **Évaluation mi-session**  Examen portant sur l’utilisation des listes et des dictionnaires, l’utilisation de contrôles de flux, dans un contexte de création d’utilitaires | Individuel, en laboratoire informatique.  Avec la documentation donnée par le professeur. | 1,2,3,4 | Exactitude de la solution  Fonctionnalité du programme  Compréhension des concepts  Qualité du code | Semaine 7 | 25% |
| **Évaluation Terminale**  Examen final portant sur l’utilisation des classes, l’encapsulation et l’héritage. | 1,2,3,4,5,6 | Semaine 15 | 25% |
| L’évaluation de la langue française se fera dans les commentaires que vous aurez mis dans votre code. | | | | **TOTAL** | 50 % |

## Travaux pratiques (50%)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mode d’évaluation et description de l’activité** | **Contexte de réalisation** | **Objectif(s) d’apprentissage** | **Critères d’évaluation** | **Échéance  (date)** | **Pondération (%)** |
| **Travail pratique 1**  Révision des fonctions, méthodes, contrôle de flux, listes et dictionnaires.  L’utilisation de la gestion de code source sera aussi évaluée. | Individuel, avec son matériel et ses notes de cours | 1,2,3,4 | Exactitude de la solution  Fonctionnalité du programme  Compréhension des concepts  Qualité du code | Semaine 6 | 25% |
| **Travail pratique 2**  Travail pratique portant sur l’utilisation des classes, l’encapsulation et l’héritage. Ainsi que le développement de tests unitaires. | 1,2,3,4,5 | Semaine 13 | 20% |
| **Travail pratique 3**  **Évaluation Terminale**  Travail pratique portant sur le développement de tests unitaires | 6 | Semaine 15 | 5% |
| L’évaluation de la langue française se fera dans les commentaires et lors des « commits » des travaux sur « github ». | | | | **TOTAL** | 50 % |

Les travaux pratiques ou les exercices des examens sont développés en Python dans l'environnement Visual Studio Code.

# Matériel requis obligatoire

* (aucun)

# Médiagraphie

**Documentation Python**

**https://docs.python.org/3/**

**Sites de référence**

[**https://www.w3schools.com/cs/**](https://www.w3schools.com/cs/)

# Conditions de réussite au cours

**1. Note de passage**

La note de passage du cours (PIEA, article 5.1m) est de 60 %.

***Double sanction****:* Pour réussir le cours, l'étudiant doit conserver un minimum de 60% à l’évaluation théorique (examens) et un minimum de 60% à l’évaluation pratique (TPs).

À défaut de répondre à ces exigences, l’étudiant obtient, à son bulletin, la note la plus basse enregistrée : celle des évaluations théoriques ou celle des évaluations pratiques du cours.

En aucun cas, un travail d'équipe n'assure la même note aux coéquipiers ; tout étudiant faisant partie d'une équipe est sujet à une évaluation personnelle sur le travail remis par l’équipe. La note du travail sera accordée en fonction des résultats obtenus lors de cette évaluation personnelle.

**2. Présence aux évaluations sommatives**

La présence aux activités d’évaluation sommative est obligatoire (PIEA 5.2.5.1). Les évaluations sommatives comprennent les TPs et les examens.

**3. Remise des travaux**

Les travaux exigés par un professeur doivent être remis à la date, au lieu et au moment fixés. Les pénalités entraînées par les retards sont établies selon les règles départementales (PIEA, article 5.2.5.2).

En cas de retard, les **pénalités départementales** sont :

* Dans un cas exceptionnel, un travail en retard pourra être accepté moyennant une pénalité de 10% par jour de retard.

**4. Présentation matérielle des travaux**

L’étudiant doit respecter les « *Normes de présentation matérielle des travaux écrits* » adoptées par le Cégep. Ces normes sont disponibles à l’adresse suivante : <http://rmsh.cegepmontpetit.ca/normes-de-presentation-materielle-des-travaux-ecrits-du-cegep/>.

Les **pénalités départementales** concernant le non-respect des normes de présentation matérielle des travaux (PIEA, article 5.3.2) sont :

* En cas de non-respect des normes de présentation, les travaux ne seront pas acceptés et seront retournés à l’étudiant pour révision. Le professeur peut ainsi retarder l’acceptation d’un travail jusqu’à ce qu’il soit conforme aux normes de présentation matérielle et les pénalités de remise des travaux s’appliquent.

**5. Qualité de la langue française**

L’évaluation de la qualité de la langue (PIEA, article 5.3.1) doit respecter les critères et les valeurs établis par le département.

Article 5.3.1 de la PIEA : « La maîtrise de la langue des étudiants est évaluée dans tous les cours où le français est la langue d’enseignement. » Au regard de l’importance d’une bonne maîtrise du français, nous vous invitons à consulter le site du Cégep Le français s’affiche ([www.cegepmontpetit.ca/lefrancais-saffiche](http://www.cegepmontpetit.ca/lefrancais-saffiche)).

La **procédure départementale** d’évaluation de la qualité du français est :

* Dans les activités sommatives, 10% des points peuvent être alloués à l’évaluation de la langue française. Au moins un des éléments suivants sera évalué :
  + Code de la langue (syntaxe, grammaires, orthographe…)
  + Vocabulaire et terminologie appropriée au travail

# Modalités de participation au cours

Il est de la responsabilité de l’étudiant d’être présent à tous ses cours et de participer activement aux activités d’apprentissage prévues par l’enseignant, conformément à l’article 6.3.4 de la politique institutionnelle d’évaluation des apprentissages (PIEA).

Dans le cas d’absence non motivée, il est de la responsabilité de l'étudiant de s’approprier les connaissances vues lors de son absence avant la tenue du prochain cours. L’enseignant n’est pas tenu de répondre aux questions des étudiants absents sans motif valable.

# RÈGLES DÉPARTementales

### Présence aux cours (article 5.3.4 PIEA)

Comme le stipule l’article 5.3.4 de la PIEA, la présence au cours\* est une preuve d’engagement de l’étudiant dans ses études. Le professeur doit consigner les absences dans le système électronique de gestion des absences ou sur un registre que l’étudiant pourra consulter. En cas d'absences répétées de l'étudiant, la procédure suivante s'applique :

Les absences non motivées seront comptabilisées et un premier avis écrit sera envoyé à l’étudiant par MIO à la suite d’absences équivalant à une semaine de cours (1/15). Il précisera les dates des absences, le nombre d'heures d'absences compilé ainsi que la possibilité d’exclusion qui pourrait entraîner un échec au cours si le pourcentage d’absences franchit le 2e seuil.

À la suite des absences équivalant à deux semaines du nombre d'heures du cours (2/15), un avis d'exclusion sera acheminé à l’étudiant et à la Direction des études qui en conservera le registre. La sanction pour cause d'absences se traduira par la note cumulée au moment de l'exclusion.

(\*) Pour être considéré comme présent à un cours, il faut être actif c’est-à-dire participer aux activités d’apprentissage, avoir complété la partie exigée des exercices/travaux et avoir fait une remise de ces exercices/travaux à la fin de chacune des rencontres.

### Travaux d’équipe

Lors de travaux d’équipe en classe, la présence est obligatoire sous peine d’avoir la note zéro (0) pour ce travail.

**Règles d’utilisation des laboratoires**

Les règles concernant l’utilisation des laboratoires du département Informatique sont affichées dans les laboratoires du département et doivent être respectées.

### Plagiat (article 5.3.4 PIEA)

Pour des fins de clarification, les directives suivantes sont applicables :

* Un premier plagiat sera sanctionné d'un 0 au travail. Si vous avez déjà eu un plagiat au collège, vous aurez une rencontre avec la direction.
* Un second plagiat entraîne la note de 0 au cours.

S'aider ou plagier :

* Vous avez le droit de travailler ensemble, de vous aider en expliquant des notions et autres. Parlez et expliquez à un autre étudiant est encouragé.
* Vous ne pouvez PAS échanger des fichiers, des parties de fichiers, vos clés USB ou disque externe etc. entre vous. Cela inclut recopier lettre par lettre le travail d'un autre.

Copier ou être copié :

* Dans un cas de plagiat, celui qui a partagé son travail est sanctionné au même titre que celui qui l'a copié.
* Protégez votre travail. Fermez votre session en quittant. Si vous partagez votre travail, ou le médium sur lequel il est enregistré, vous ne contrôlez pas ce que les autres vont en faire.

# Politiques et règles institutionnelles

Tout étudiant inscrit au cégep Édouard-Montpetit doit prendre connaissance du contenu de quelques politiques et règlements institutionnels et s’y conformer. Notamment, la *Politique institutionnelle d’évaluation des apprentissages* (PIEA), la*Politique institutionnelle de la langue française* (PILF), *la Politique pour un milieu d’études et de travail exempt de harcèlement et de violence* (PPMÉTEHV),les *Conditions d’admission et cheminement scolaire,* la *Procédure concernant le traitement des plaintes étudiantes dans le cadre des relations pédagogiques.*

Le texte intégral de ces politiques et règlements est accessible sur le site Web du Cégep à l’adresse suivante : [www.cegepmontpetit.ca/reglements-et-politiques](http://www.cegepmontpetit.ca/reglements-et-politiques). En cas de disparité entre des textes figurant ailleurs et le texte intégral, ce dernier est la seule version légale et appliquée.

# le centre de services adaptés – pour les étudiants en situation de handicap

Les étudiants ayant un diagnostic d’un professionnel (limitations motrices, neurologiques, organiques, sensorielles, troubles d’apprentissage, de santé mentale, trouble du spectre de l’autisme ou autres) ou ayant une condition médicale temporaire peuvent faire une demande pour obtenir des mesures adaptées.

Pour avoir accès à ce service, faites parvenir votre diagnostic soit par mio à "Service, CSA" ou par courriel à [servicesaadaptes@cegepmontpetit.ca](mailto:servicesaadaptes@cegepmontpetit.ca).

Si vous avez déjà un plan de mesures adaptées avec le CSA, vous êtes invités à communiquer avec votre professeur dès le début de la session afin de discuter avec lui des mesures d’accommodement déterminées par le CSA.

# ANNEXE - GRILLE D’ÉVALUATION DU FRANÇAIS ÉCRIT

**MA RÉUSSITE AU CÉGEP**

Pour avoir un aperçu de l’ensemble des ressources et services disponibles, vous êtes invités à consulter le site web [https://mareussite.cegepmontpetit.ca/](about:blank).

**GRILLE D’ÉVALUATION DU FRANÇAIS ÉCRIT**

|  |  |
| --- | --- |
| **Qualité du français écrit** | **/10** |
| Clarté du propos. | /3 |
| Emploi des termes techniques. | /4 |
| Respect du code orthographique et grammatical (les accords). | /3 |

\*Cette grille d’évaluation a été créée avec la collaboration de M. Jean-Sébastien Ménard, enseignant de littérature

# ANnEXE – semainier du cours

|  |  |  |
| --- | --- | --- |
| **Semaine** | **Rencontre** | **Matière** |
| S1 | R1 | Intro Python |
|  |  | Indentation pour les blocs de code |
|  |  | Types des variables |
|  |  | Opérateurs logiques et de comparaison |
|  |  | Conditionnel |
|  |  | Boucle while et boucle for |
|  | R2 | List |
|  |  | List et String |
|  |  | Les méthodes de List |
|  |  | Boucle for (pour passer au travers d’une collection) |
| S2 | R3 | GitHub |
|  |  | Approfondissement String |
|  |  | Introduction Package OS |
|  | R4 | Approfondissement Listes |
| S3 | R5 | Utilisation des fichiers de texte |
|  | R6 | Dictionnaire |
| S4 | R7 | Utilisation des fichiers csv |
|  | R8 | Synthèse de la matière des premières semaines, début Fonctions |
| S5 | R9 | Fonctions |
|  | R10 | Fonctions et modules |
| S6 | R11 | TP 1 |
|  | R12 | TP 1 (25%) |
| S7 | R13 | Formatif |
|  | R14 | Intra (25%) |
| S8 | R15 | Utiliser les classes |
|  |  | Création des classes |
|  | R16 | Création de classes |
|  |  | Création de méthodes dans la classe |
|  |  | Variable de classe |
| S9 | R17 | Class and static methods using decorator |
|  | R18 | Héritage – Création de sous-classe |
| S10 | R19 | Héritage – Création de sous-classe |
|  | R20 | Modélisation d’une typologie de réseau |
| S11 | R21 | Modélisation d’une typologie de réseau |
|  | R22 | TP 2 |
| S12 | R23 | TP 2 (20%) |
|  | R24 | Méthodes spéciales (dunder) |
| S13 | R25 | Property Decorators: Getters, Setters and Deleters |
|  | R26 | Tests unitaires |
| S14 | R27 | Tests unitaires(suite) |
|  | R28 | TP3 (5%) |
| S15 | R29 | Formatif |
|  | R30 | Examen Final (25%) |