

# Plan de cours



Numéro du cours

Pondération

Unités

**420-FAH-LI**

**1-2-2**

**1,66**

---

## Bloc 2

2022-06-20 – 2022-10-21

Titre du cours

Département

**Développement informatique**

**Informatique**



**Cégep Limoilou**

Campus de Québec

Programme

**AEC Développement d'applications  
sécuritaires**

*Cohorte 3*

Enseignant(e)s

**Eric Martel**

*Contact : Teams (préfééré) ou MIO*

## 1. Modalités du cours

Le cours sera offert	En présence
Sur quelle plateforme de vidéoconférence les cours seront-ils donnés ?	Teams (si besoin)
Quels sont les applications ou logiciels à installer et à utiliser pour suivre le cours ?	Teams + voir section 6
À quel endroit retrouve-t-on la documentation relative aux cours (les notes de cours, capsules, exercices) ?	Teams
À quel endroit seront partagées les consignes pour les travaux et les évaluations?	Teams
Comment et quand rejoindre l'enseignant ?	Teams (préféré) / MIO  Disponibilités : L'enseignant est complètement disponible lors des séances prévues à l'horaire et se rend également disponible autant que possible à l'extérieur des heures de cours.

Veuillez utiliser les liens suivants afin d'accéder aux présentations et aux procédures concernant l'utilisation de chacun des outils

<a href="#">Compte utilisateur du cégep</a>	<a href="#">Installer Zoom</a>	<a href="#">Installer Teams</a>
<a href="#">Office 365</a>	<a href="#">Antidote Web</a>	<a href="#">Citrix : Accès aux laboratoires informatiques du cégep et accès aux machines virtuelles</a>
<a href="#">Guide de l'étudiant</a>	<a href="#">Étudier à distance</a>	<a href="#">Liste complète des procédures</a>

## **2. Présentation du cours**

### **2.1. Place et rôle du cours dans le programme**

Le cours **420-FAH-LI : Développement informatique** situé en deuxième bloc de l'AEC en Développement d'applications sécuritaires permet d'acquérir des connaissances complémentaires à la programmation. Ce cours combine des éléments réinvestis de la compétence 00Q2 et le développement de la compétence DY63. Les éléments de la compétence 00Q2 – *Utiliser des langages de programmation* qui seront développés dans ce cours sont, entre autres, le débogage du code, l'application de tests fonctionnels et la décomposition du problème. Au niveau de la DY63 - *Exploiter des principes de la programmation orientée objet*, l'analyse, la modélisation et la validation du code orienté objet seront traités ainsi que l'application de procédures liées à la gestion des versions de programmes. Ces éléments de compétences seront combinés afin de vous amener à développer de bonnes pratiques de programmation. Ils apprendront ainsi à créer des diagrammes de classes, des cas d'utilisation et à en tirer des jeux d'essais. Ils feront toutes ces étapes en appliquant des bonnes pratiques liées à la gestion des versions.

Ce cours est un complément aux cours de programmation orientée objets. Les notions apprises dans ce cours seront importantes pour faciliter la compréhension et le développement en équipe de programmes orientés objets valides et bien conçus. Les éléments abordés dans ce cours vous aideront à mieux comprendre les concepts de l'orienté objet en modélisant. De plus, la production de code incluant des tests et le travail en équipe avec la gestion des versions du code source seront abordés. Ce cours permet donc de soutenir le développement de la compétence DY63 et il doit être utilisé pour appuyer et faciliter les apprentissages faits dans le cours de programmation orientée objet II qui se donne en parallèle.

Ce cours se donne parallèlement à :

- **420-FAG-LI - Programmation Web 2** : Programmation d'une application Web interactive;
- **420-FAE-LI - Bases de données 2** : Création et utilisation des requêtes complexes, des fonctions, procédures et déclencheurs utilisant plusieurs tables afin d'exploiter une base de données de façon sécuritaire tout en assurant l'intégrité des données. Il permet d'apprendre et d'appliquer des concepts de base au niveau de l'administration d'une base de données (sauvegarde, réplication, gestion des utilisateurs);
- **420-FAF-LI – Programmation orientée objet 2** : Mise en place des concepts de programmation orientée objet avancés (héritage, polymorphisme, etc.), familiarisation à diverses structures de données en mémoire et à la persistance des données.

Ce cours a comme préalable relatif (à 50%) le cours **420-FAB-LI Programmation orientée objet I**.

### **2.2. Objectif terminal du cours**

À la fin de ce cours, vous aurez créé et utilisé des diagrammes de classe, aurez élaboré et mis en application des plans de test et ce, en utilisant un environnement de collaboration.

Pour ce faire, vous serez en mesure :

- d'assurer la validité du code;
- d'utiliser des outils collaboratifs;
- de modéliser graphiquement un programme orienté objet.

### 3. Organisation du cours

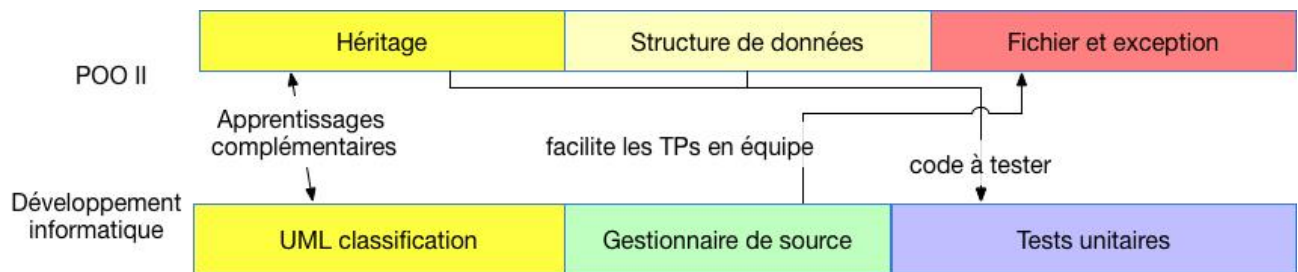
#### 3.1. Contenu et apprentissages-clés

Capacités essentielles à développer ou à maîtriser	Savoir-penser nécessaires	Savoir-agir nécessaires	Savoir-devenir nécessaires
<p>Capacité 1 :</p> <p>Assurer la validité du code</p>	<ul style="list-style-type: none"> <li>• Identifier les types de tests permettant de valider et tester du code ;</li> <li>• Décrire les tests à effectuer dans un français écrit adéquat (grammaire et orthographe) ;</li> <li>• Identifier la cause d'un problème ;</li> <li>• Distinguer l'importance relative des différents tests ;</li> <li>• Déterminer les outils et la procédure pour effectuer des diagnostics.</li> </ul>	<ul style="list-style-type: none"> <li>• Rédiger un plan de test ;</li> <li>• Concevoir des tests unitaires ;</li> <li>• Déterminer la couverture des tests présents ;</li> <li>• Produire des programmes validés ;</li> <li>• Trouver la cause d'un problème en utilisant les outils nécessaires (débugueur et traces) et y apporter les correctifs.</li> </ul>	<ul style="list-style-type: none"> <li>• Travailler de manière structurée, organisée, logique et méthodique ;</li> <li>• Développer un sens critique quant à son travail afin d'identifier les failles ;</li> <li>• Adopter des pratiques de travail d'équipe rigoureuses ;</li> <li>• Rédiger systématiquement de la documentation ;</li> <li>• Adopter une attitude analytique et critique.</li> </ul>
<p>Capacité 2 :</p> <p>Utiliser des outils collaboratifs</p>	<ul style="list-style-type: none"> <li>• Identifier des méthodes de répartition du travail ;</li> <li>• Reconnaître les bonnes pratiques en matière de messages dans la gestion des versions ;</li> <li>• Planifier ses intégrations pour limiter les conflits.</li> </ul>	<ul style="list-style-type: none"> <li>• Configurer le gestionnaire de versions ;</li> <li>• Gérer les conflits ;</li> <li>• Partager son travail avec une équipe ;</li> <li>• Écrire des messages dans la gestion des versions dans un français adéquat (grammaire et orthographe).</li> </ul>	
<p>Capacité 3 :</p> <p>Modéliser graphiquement un programme OO.</p>	<ul style="list-style-type: none"> <li>• Déterminer les objets nécessaires pour réaliser un programme ;</li> <li>• Déterminer les constituant des objets (attributs et méthodes) ;</li> <li>• Identifier les liens entre les objets (hiérarchisation, agrégation, composition, dépendance...).</li> </ul>	<ul style="list-style-type: none"> <li>• Utiliser l'outil de modélisation pour représenter les modèles objets ;</li> <li>• Générer des diagrammes à partir du code et l'inverse ;</li> <li>• Produire des diagrammes de classes et de cas d'utilisation.</li> </ul>	

### 3.2. Organisation du cours et méthodes pédagogiques

#### Structure du cours, déroulement de l'apprentissage et de l'enseignement, activités nécessaires

Le cours **420-FAH-LI : Développement informatique** permet d'acquérir des compétences complémentaires à la programmation. Ainsi, plusieurs liens seront faits avec le cours **420-FAF-LI : Programmation orientée objet II** qui sera donné en parallèle. Le diagramme suivant présente la séquence d'apprentissages qui se fera simultanément.



Le cours sera divisé en trois grandes étapes :

- Étape 1 : Modélisation
- Étape 2 : Gestionnaire de sources
- Étape 3 : Tests unitaires

Ainsi, le cours débutera par les notions de diagrammes de classe et de cas d'utilisation. L'apprentissage des notions de modélisation graphique (ex : UML) pourra aider les étudiantes et étudiants à consolider les notions d'héritages qui seront acquises au début du cours POO II.

Par la suite, les gestionnaires de sources seront abordés en utilisant un corrigé de travail du cours de POO II. Les étudiantes et étudiants pourront ainsi utiliser un exemple concret et le modifier en utilisant un gestionnaire de sources.

Le cours portera finalement sur la validation du code et la conception de tests unitaires. Pour ce faire, des corrigés fournis par l'enseignant de POO II pourront être utilisés afin de tester du code lié aux apprentissages de la session. De plus, reprendre du code familier devrait permettre à l'étudiant d'être plus rapidement efficace et ainsi de pouvoir se concentrer davantage sur la capacité qu'il doit développer.

Quelques rencontres sont à prévoir entre les enseignants des cours **420-FAH-LI : Développement informatique** et **420-FAF-LI : Programmation orientée objet II** pendant la session afin que les cours soient bien complémentaires et arrimés.

### 3.3. Calendrier des activités<sup>1</sup>

Semaine	Théorie	Exercices et évaluations
<b>01</b> <b>06-20 (4h)</b>	<ul style="list-style-type: none"> <li>UML               <ul style="list-style-type: none"> <li>Langage UML et diagrammes de classe</li> <li>Logiciels (Visio, Yed, draw.io, Visual Paradigm, etc.)</li> </ul> </li> </ul>	Exercices
<b>02</b> <b>06-27 (4h)</b>	<ul style="list-style-type: none"> <li>UML               <ul style="list-style-type: none"> <li>Création des classifications</li> <li>Dépendances entre les classes</li> </ul> </li> </ul>	TP1 (5 %)
<b>03</b> <b>07-04 (4h)</b>	<ul style="list-style-type: none"> <li>UML               <ul style="list-style-type: none"> <li>Faire le lien entre le code et le modèle UML</li> <li>Générer UML à partir de Visual Studio</li> </ul> </li> </ul>	TP1
<b>04</b> <b>07-11 (3h)</b>	<b>Examen 1 (15 %) – 14 juillet</b>	
<b>05</b> <b>07-18 (0h)</b>	<b>Vacances</b>	
<b>06</b> <b>07-25 (0h)</b>	<b>Vacances</b>	
<b>07</b> <b>08-01 (0h)</b>	<b>Vacances</b>	
<b>08</b> <b>08-08 (3h)</b>	<ul style="list-style-type: none"> <li>Systèmes de gestion de version (SGV)               <ul style="list-style-type: none"> <li>SVN, Git,</li> <li>Serveurs de version (GitLab, GitHub, BitBucket, etc.)</li> <li>Opérations de base (checkout/clone, update/pull, commit)</li> </ul> </li> </ul>	TP2 (8 %)
<b>09</b> <b>08-15 (3h)</b>	<ul style="list-style-type: none"> <li>SGV               <ul style="list-style-type: none"> <li>Gestion des conflits</li> <li>Branches et bifurcations</li> <li>Travail collaboratif</li> </ul> </li> </ul>	TP2
<b>10</b> <b>08-22 (3h)</b>	<b>Examen 2 (20 %) – 25 août</b>	
<b>11</b> <b>08-29 (3h)</b>	<ul style="list-style-type: none"> <li>Tests unitaires               <ul style="list-style-type: none"> <li>Invariants</li> <li>Éviter les débordements</li> <li>Théorie du contrat et assertions</li> <li>Plan de tests</li> <li>Couverture de code</li> </ul> </li> </ul>	TP3 (12 %)
<b>12</b> <b>09-05 (3h)</b>	<ul style="list-style-type: none"> <li>Tests unitaires               <ul style="list-style-type: none"> <li>Débogage avancé</li> <li>Test des éléments statiques</li> <li>Test des éléments non déterministes</li> </ul> </li> </ul>	TP3
<b>13</b> <b>09-12 (3h)</b>	<ul style="list-style-type: none"> <li>Tests unitaires               <ul style="list-style-type: none"> <li>Optimisation et loi de De Morgan</li> </ul> </li> </ul>	TP4 (15 %)
<b>14</b> <b>09-19 (3h)</b>	<ul style="list-style-type: none"> <li>Tests unitaires               <ul style="list-style-type: none"> <li>Tests impliquant des fichiers</li> </ul> </li> </ul>	TP4

<sup>1</sup> Cet échéancier est donné à titre indicatif. L'enseignant peut modifier l'ordre de présentation et les délais prévus pour chaque activité. Si un événement quelconque empêche la tenue d'un examen, celui-ci est automatiquement remis au cours suivant.

<b>15</b> <b>09-26 (3h)</b>	Révision	TP4 + Exercices
<b>16</b> <b>10-03 (3h)</b>	<b>Examen 3 (25%) – 10 octobre</b>	
<b>17</b> <b>10-10 (3h)</b>	Retour sur examen 3 + temps pour TP4	TP4

#### **4. Modalités d'évaluation des apprentissages**

Ce cours étant divisé en trois grandes parties du « coffre à outils », comprendra plusieurs évaluations de chaque portion. Les évaluations formatives pourront prendre différentes formes mais devront être faites de manière **individuelle** au moins une fois pour chaque grand bloc. Par exemple, les diagrammes de classe et de cas d'utilisation devront être remis pour commentaires, les jeux d'essais pourront être testés au niveau de la couverture de code (évaluation formative faite directement par l'étudiant). Au niveau du gestionnaire de sources, les configurations pourront être validées mais l'évaluation (autant formative que sommative) n'est pas un élément qui sera facile. C'est le type de chose qui peut être observée à travers le temps et l'utilisation.

##### **4.1. Liste des évaluations (formatives et sommatives)**

Les divers travaux qui sont utilisés pour l'évaluation sommative sont listés dans le tableau ci-après décrivant les éléments évalués dans chacun.

##### **Répartition des évaluations par défi et par capacité**

	TP 1	TP 2	TP 3	TP 4	Examen 1	Examen 2	Examen 3	Total
<b>Capacité 1</b>			✓	✓			✓	<b>40 %</b>
<b>Capacité 2</b>		✓	✓	✓		✓	✓	<b>30 %</b>
<b>Capacité 3</b>	✓	✓	✓	✓	✓	✓	✓	<b>30 %</b>
<b>Pondération</b>	<b>5 %</b>	<b>8 %</b>	<b>12 %</b>	<b>15 %</b>	<b>15 %</b>	<b>20 %</b>	<b>25 %</b>	<b>100 %</b>

## 4.2. Capacités et aspects/critères d'évaluation

Capacités essentielles à évaluer sommativement	Aspects évalués et critères d'évaluation		Moyens (dispositifs) privilégiés
	Niveau optimal	Niveau minimal (seuil)	
Capacité 1 : Assurer la validité du code Pondération : 40%	<ul style="list-style-type: none"> <li>Rédiger un plan de tests <b><u>complet</u></b> et <b><u>rigoureux</u></b> dans un français adéquat (grammaire et orthographe) ;</li> <li>Concevoir des tests unitaires et <b><u>fonctionnels</u></b> permettant une <b><u>évaluation précise et détaillée du code</u></b> ;</li> <li>Utiliser <b><u>adéquatement</u></b> des outils afin de déterminer la couverture de code (ce qui a été testé) et <b><u>optimiser</u></b> la couverture, <b><u>de manière autonome</u></b> ;</li> <li>Déterminer <b><u>précisément, de manière autonome</u></b> la source d'un problème et y apporter un correctif <b><u>adéquat en tout temps</u></b> ;</li> <li>Identifier <b><u>correctement en tout temps et de manière autonome</u></b> les dysfonctionnements en utilisant un débogueur et des outils de traces.</li> </ul>	<ul style="list-style-type: none"> <li>Rédiger un plan de tests <b><u>pouvant comporter certains oublis mineurs dans un français adéquat</u></b> ;</li> <li>Concevoir des tests unitaires et <b><u>fonctionnels</u></b> permettant une <b><u>évaluation sommaire du code</u></b> ;</li> <li>Utiliser <b><u>adéquatement</u></b> des outils afin de déterminer la couverture de code (ce qui a été testé) mais <b><u>peut avoir des difficultés à maximiser la couverture</u></b> de code dans les tests ;</li> <li>Déterminer <b><u>précisément</u></b> la source d'un problème et y apporter un correctif <b><u>adéquat avec de l'aide occasionnelle</u></b> ;</li> <li>Identifier <b><u>correctement de manière autonome la plupart du temps, ou avec de l'aide occasionnelle</u></b>, les dysfonctionnements en utilisant un débogueur et des outils de traces.</li> </ul>	<ul style="list-style-type: none"> <li>Exercices formatifs – en environnement non contrôlé et à au moins 2 reprises</li> <li>Travaux pratiques - en environnement non contrôlé et à au moins 2 reprises</li> <li>Examens pratiques – en environnement contrôlé et à 2 reprises</li> </ul>
Capacité 2 : Utiliser des outils collaboratifs Pondération : 30 %	<ul style="list-style-type: none"> <li>Configurer <b><u>adéquatement et de manière autonome</u></b> le gestionnaire de versions afin de suivre et gérer l'évolution du code ;</li> <li>Effectuer des <b><u>opérations valides en tout temps</u></b> (ajout, modification, déplacement, suppression) lors de l'utilisation des outils collaboratifs ;</li> <li>Gérer <b><u>de manière autonome en tout temps</u></b> les conflits de versions (sans branches) ;</li> <li>Répartir <b><u>adéquatement</u></b> le travail pour favoriser l'avancement de l'équipe et <b><u>minimiser en tout temps ou la plupart du temps</u></b> les conflits de code ;</li> <li>Laisser des messages de validations <b><u>significatifs, complets, concis et clairs</u></b> dans un français adéquat (grammaire et orthographe, <b><u>en tout temps</u></b> ;</li> <li>Intégrer <b><u>efficacement</u></b> son travail <b><u>sans corrompre</u></b> le code déjà en place, <b><u>de manière autonome en tout temps</u></b>.</li> </ul>	<ul style="list-style-type: none"> <li>Configurer <b><u>adéquatement</u></b> le gestionnaire de versions afin de suivre et gérer l'évolution du code mais peut nécessiter de <b><u>l'aide occasionnelle</u></b> ;</li> <li>Effectuer des <b><u>opérations valides la plupart du temps</u></b> (ajout, modification, déplacement, suppression) lors de l'utilisation des outils collaboratifs ;</li> <li>Gérer <b><u>avec de l'aide</u></b> les conflits de versions (sans branches) ;</li> <li>Répartir <b><u>adéquatement</u></b> le travail pour favoriser l'avancement de l'équipe et <b><u>minimiser en tout temps ou la plupart du temps</u></b> les conflits de code ;</li> <li>Laisser des messages de validations <b><u>significatifs, complets, concis et clairs</u></b> dans un français adéquat (grammaire et orthographe, <b><u>la plupart du temps</u></b> ;</li> <li>Intégrer son travail <b><u>sans corrompre</u></b> le code déjà en place, <b><u>avec de l'aide occasionnelle</u></b>.</li> </ul>	<ul style="list-style-type: none"> <li>Exercices formatifs – en environnement non contrôlé et à au moins 2 reprises</li> <li>Travaux pratiques - en environnement non contrôlé et à au moins 2 reprises</li> <li>Examens pratiques – en environnement contrôlé et à 2 reprises</li> </ul>
Capacité 3 : Modéliser graphiquement un programme OO Pondération : 30%	<ul style="list-style-type: none"> <li>Produire <b><u>adéquatement</u></b> les diagrammes UML correspondant en <b><u>tous points</u></b> à du code fourni ;</li> <li>Transformer <b><u>sans erreurs ou oublis</u></b> du code existant en représentation graphique <b><u>complète de manière autonome</u></b> ;</li> <li>Générer <b><u>adéquatement</u></b> un squelette de code à partir d'un diagramme UML et produire des diagrammes <b><u>complets</u></b> répondant à <b><u>l'ensemble</u></b> des exigences du client.</li> </ul>	<ul style="list-style-type: none"> <li>Produire <b><u>adéquatement</u></b> les diagrammes UML correspondant <b><u>partiellement</u></b> à du code fourni ;</li> <li>Transformer du code existant en représentation graphique pouvant comporter <b><u>quelques lacunes de manière autonome ou avec de l'aide occasionnelle</u></b> ;</li> <li>Générer un squelette de code à partir d'un diagramme UML pouvant <b><u>comporter quelques lacunes</u></b> et produire des diagrammes (pouvant manquer de précision) répondant à <b><u>l'ensemble</u></b> des exigences du client.</li> </ul>	<ul style="list-style-type: none"> <li>Exercices formatifs – en environnement non contrôlé et à au moins 2 reprises</li> <li>Travaux pratiques - en environnement non contrôlé et à au moins 2 reprises</li> <li>Examens pratiques – en environnement contrôlé et à 2 reprises</li> </ul>



### **4.3. Conditions de réussite du cours**

Pour réussir le cours, l'étudiante ou l'étudiant devra minimalement avoir atteint chaque capacité à 60%. Étant donné que l'étudiant doit réussir les 3 capacités, on doit lui permettre de le faire jusqu'à la fin de la session pour ces 3 capacités. Il faut donc prévoir évaluer en fin de session ces 3 capacités.

### **4.4. Modalités de reprise d'une évaluation sommative**

- Sur présentation d'une attestation écrite officielle, si vous êtes absent pour cause de force majeure (maladie, accident, décès d'un proche parent, subpoena, etc.) vous pourrez passer une évaluation à une date fixée conjointement avec l'enseignant concerné.
- Les voyages, vacances et activités de loisir ne constituent pas des cas de force majeure.
- L'absence à un cours ne relevant pas d'un cas de force majeure ne peut justifier la remise en retard d'un travail exigé par l'enseignant.

## **5. Coordonnées et disponibilités de l'enseignant**

**Enseignant :** Eric Martel

**Local :** Q3523

**Messagerie :** Teams (préféré et plus rapide) ou MIO (délai de 24h ouvrables)

## **6. Matériel**

### **Obligatoire**

- La méthode d'accès aux notes de cours ainsi que des capsules vidéo seront disponibles sur Teams.

### **Logiciels utilisés dans le cours**

- Visual Studio 2019
- SVN / TortoiseSVN
- Git / TortoiseGit
- Visio
- Visual Paradigm

### **Matériel utilisé dans le cours**

Vous pouvez utiliser :

- un ordinateur que vous possédez déjà (poste fixe ou portable) pourvu qu'il possède les caractéristiques minimales nécessaires;

- un des ordinateurs proposés dans le cadre de l'entente avec la Coop Zone;
- un autre modèle que vous achetez chez un autre fournisseur et qui possède les caractéristiques minimales nécessaires.

Les caractéristiques minimales recommandées pour un ordinateur portable avec un système d'exploitation PC (Windows 10) sont :

- une carte réseau WiFi;
- un processeur i5;
- une mémoire de 8 Go (16 Go recommandée);
- un disque dur SSD de 250 Go;
- un écran de 14 ou 15 pouces;
- une carte vidéo dédiée;
- une caméra web (intégrée ou externe);
- un micro (intégré ou externe);
- deux (2) ports USB.

Voici également les accessoires optionnels que nous vous recommandons de vous procurer :

- Une mallette de transport ou sac à dos conçu à cet effet;
- Un câble de sécurité;
- Un casque d'écoute avec micro;
- Une souris optique (avec ou sans fil);
- Une clé de mémoire USB (minimum 16 Go).

***Pour vous procurer un ordinateur dans le cadre de l'entente avec la coop Zone :***

Vous pouvez vous procurer l'ordinateur par un achat en ligne en cliquant sur le lien suivant :

<https://www.zone.coop/informatique/programmes-dacquisition-dordinateur/limoilou.html>

Vous pouvez aussi vous rendre aux différents points de vente de la coop. Pour les heures d'ouverture, nous vous invitons à consulter la page suivante: <https://www.zone.coop/la-cooperative/heures-d-ouverture.html>

## **7. Médiagraphie**

- Collins-Sussman, B., Fitzpatrick, B.W. et Pilato, C.M. (2022, 15 juin). Gestion de versions avec Subversion. <https://svnbook.red-bean.com/>
- Git – Documentation. (2022, 15 juin). Git. <https://git-scm.com/doc>
- Gervais, Luc (2020). *Apprendre la Programmation Orientée Objet avec le langage C# – 3<sup>e</sup> édition*. ENI.
- Osheroove, Roy (2013). *The Art of Unit Testing: with examples in C# – 2<sup>nd</sup> édition*. Manning.

## **8. Code de vie**

Les enseignantes et enseignants du Département d'informatique ont adopté les éléments suivants dans une optique de favoriser la réussite dans les cours.

### **Représentant de classe**

Un représentant de classe pourrait être désigné dans les deux premières semaines de la session afin de créer une synergie dans les groupes et d'avoir une personne identifiée pour aider à l'amélioration continue des cours. Deux rencontres pourraient être faites par session avec ces représentants afin de connaître les doléances, s'il y a lieu.

### **Présence active**

Nous considérons que la présence active en classe est essentielle à la réussite scolaire. En ce sens, le département s'attend à ce que les étudiants assistent à tous les cours et fassent tous les travaux formatifs demandés par leurs enseignants.

Par présence active, nous entendons plus qu'une simple présence physique en classe. La présence active comprend également la participation aux activités d'apprentissage (incluant les lectures préalables), la réalisation des travaux et exercices formatifs suggérés, la réalisation des travaux et exercices sommatifs demandés et enfin, l'implication personnelle dans les travaux d'équipes.

### **Absence**

Les enseignants prennent les présences à chacun des cours. Une étudiante ou un étudiant qui s'absente a la responsabilité de :

- Contacter son enseignant dans les plus brefs délais afin de déterminer une méthode de régulation de la situation avec de son enseignant (ex : travaux et lecture);
- Faire un suivi des mesures préconisées auprès de son enseignant (Vérifier les travaux, discussion avec son enseignant, etc.).

### **Plagiat**

Le plagiat est formellement interdit, sous quelque forme que ce soit.

Est notamment considéré comme du plagiat le fait de reproduire ou de réutiliser en tout ou en partie le travail d'un autre étudiant, avec ou sans son autorisation; dans le cas où il y aurait eu autorisation, l'étudiant plagié sera également considéré coupable de plagiat par complicité.

La politique du cégep<sup>2</sup> stipule qu'une étudiante ou un étudiant reconnu coupable de plagiat se voit octroyer la note de zéro lors d'un premier acte, puis la mention « échec » au cours lors d'un deuxième acte. Advenant un troisième acte de plagiat, la sanction pourra aller jusqu'au renvoi.

Ressources, conseils et aide-mémoire : <https://www.cegeplimoilou.ca/etudiants/carrefour-de-l-information/bibliotheques/guides/presenter-un-travail-ecrit/antiplagiat/>

---

<sup>2</sup> [https://www.cegeplimoilou.ca/media/1291293/b\\_07-politique-institutionnelle-d-evaluation-des-apprentissages-piea\\_fev2016.pdf](https://www.cegeplimoilou.ca/media/1291293/b_07-politique-institutionnelle-d-evaluation-des-apprentissages-piea_fev2016.pdf), Annexe 1

## 9. Modalités départementales d'évaluation des apprentissages<sup>3</sup>

En conformité avec la politique institutionnelle d'évaluation des apprentissages (P.I.É.A) version 2014, le département d'informatique apporte les précisions suivantes :

<p>Article 6.6</p> <p><i>L'évaluation de l'expression et de la communication en français</i></p>	<p>Les exigences minimales de qualité des écrits produits par l'étudiante ou l'étudiant sont établies selon les aspects et les critères suivants, permettant essentiellement une communication suffisamment claire pour que l'enseignante ou l'enseignant puisse poser un jugement sur la matière faisant l'objet de l'évaluation :</p> <ul style="list-style-type: none"> <li>• L'organisation et l'expression des idées sont suffisamment logiques et cohérentes pour permettre la compréhension aisée de l'écrit;</li> <li>• La syntaxe et la ponctuation rendent la lecture aisée malgré des faiblesses évidentes ;</li> <li>• Les fautes d'orthographe (d'usage et grammaticale) sont fréquentes, mais ne nuisent pas à la compréhension de l'écrit ;</li> <li>• Le vocabulaire, notamment la terminologie propre au domaine, est sommaire, mais utilisé de manière juste et adéquate.</li> </ul> <p>Toute évaluation sommative peut être refusée par l'enseignante ou l'enseignant si elle ne répond pas aux exigences minimales de qualité de la langue exprimées précédemment. L'enseignante ou l'enseignant accorde à l'étudiante ou l'étudiant un délai de reprise de 24 h pour la reformulation du texte, délai dont le début est à la convenance des deux parties.</p>
<p>Article 12.1</p> <p><i>L'équivalence de l'évaluation des apprentissages</i></p>	<p>Lorsqu'un cours est donné par plusieurs enseignants, un comité de cours est formé afin d'établir le plan de cours, la valeur attribuée à chaque objet d'évaluation, les tâches servant à l'évaluation sommative et leur contexte de réalisation et des grilles d'évaluations afin d'assurer une uniformité dans la correction. Ce comité doit se réunir au besoin pendant la session afin de s'assurer de l'uniformité du contenu enseigné et des modalités de correction.</p>
<p>Article 12.3</p> <p><i>La remise en retard de réalisations servant à l'évaluation sommative des apprentissages</i></p>	<p>À l'exception des cas de force majeure, une remise en retard des travaux entraîne automatiquement une pénalité de 10 % par jour de retard.</p> <p><i>Une fois qu'une réalisation servant à l'évaluation est corrigée et remise aux étudiants, l'étudiant qui n'a pas encore remis son travail se voit attribuer la note zéro à cette activité d'évaluation.</i></p>
<p>Article 13.1</p> <p><i>Le réexamen d'un résultat attribué pour une tâche</i></p>	<p>L'étudiant qui veut faire une demande de révision de notes partielle doit d'abord s'adresser à son enseignant ou enseignante. S'il est insatisfait du résultat de sa démarche, il s'adresse alors au coordonnateur du département. Celui-ci dirigera un comité formé de l'enseignant concerné et d'un enseignant qui possède les compétences à juger les motifs du litige. Le résultat devra être communiqué à l'étudiant dans les 10 jours ouvrables suivant sa demande.</p>

<sup>3</sup> Adopté par le département d'informatique, le 14 mars 2016