



Collège d'Alma

PLAN DE COURS

Session et année : Automne 2025

TITRE DU COURS : Programmation orientée objet 1

Programme : 420.B0 – Techniques de l'informatique

Numéro du cours : 420-3Q6-AA

Pondération :	2	3	2
---------------	---	---	---

Code(s) de compétence(s) : 00Q6 (Partielle)

Cours préalable(s) : 420-1Q2-AA – Programmation 1
ou
420-2Q2-AA – Programmation 2

Ce cours est préalable à : 420-5Q6-AA - Programmation orientée objet 2

Date de désinscription : 19 septembre 2025

Date limite d'abandon sans mention d'échec : 03 novembre 2025

Coordonnées de la personne enseignante :

À noter : Si deux personnes enseignantes, doubler au besoin les sections : coordonnées et disponibilité

Nom :	Lou Simard
Département :	Informatique
Téléphone :	418 668-2387, p. 465
Local :	P1-425
Adresse courriel :	lou.simard@collegealma.ca

Disponibilité de la personne enseignante :

En présence, sur rendez-vous et à distance en tout temps. Voir l'horaire sur disponible sur Teams.

LE PROGRAMME

Présentation du programme

Le programme Techniques de l'informatique vise à former des techniciennes ou techniciens en informatique qui exerceront leur profession dans les domaines du développement d'applications Web, mobile, jeux, etc. Ils participent à la conception d'applications fonctionnant sous différentes plateformes en effectuant le développement et leur maintenance.

Les intentions éducatives privilégiées au Collège d'Alma sont les suivantes :

- Capacité d'adaptation ;
- Perfectionnement constant des connaissances ;
- Capacité de raisonnement ;
- Application d'une démarche structurée ;
- Polyvalence ;
- Ouverture d'esprit.

À la fin de sa formation, le diplômé en Techniques de l'informatique au Collège d'Alma pourra :

- Développer des applications répondant aux besoins particuliers des utilisateurs ;
- Soutenir les utilisatrices et les utilisateurs ;

Déployer et gérer un réseau.

LE COURS

Présentation, place et rôle du cours dans le programme d'études

Placé à la troisième session, ce cours se veut une introduction à la programmation orientée objet. En particulier, le cours s'intéresse à la modélisation logicielle sous forme de classes. Il permet à l'étudiant d'analyser, de modéliser et de codifier selon l'approche objet des solutions à des problèmes simples et pertinents.

La compétence 00Q6 est vue partiellement dans ce cours. Les éléments de compétence visés dans ce cours sont :

- Analyser le problème
- Modéliser les classes
- Produire les algorithmes pour les méthodes
- Programmer des classes
- Documenter la programmation
- Appliquer la procédure liée à la gestion des versions de programmes

Les autres éléments de compétence seront abordés à la quatrième session dans le cours « 420-4P2-AA Programmation d'interfaces graphiques ».

Ce cours permettra à l'étudiant de :

- De favoriser son évolution et l'approfondissement de ses savoirs professionnels ;
- De lui permettre de comprendre les principes sous-jacents aux techniques et aux technologies utilisées ;
- De lui permettre de développer sa faculté d'Expression, sa créativité, son sens de l'initiative et son esprit d'entreprise.

Le travail pratique sera privilégié. L'étudiant aura à chaque semaine **deux périodes de cours** avec un **contenu théorique**. Ces périodes seront utilisées pour communiquer des informations générales, présenter certains contenus plus théoriques et faire des examens ou des exercices dirigés.

Trois autres périodes seront assignées en laboratoire pour effectuer les différents travaux requis.

L'étudiant devra de plus s'attendre à fournir trois (2) heures de travail personnel par semaine. Il aura accès au laboratoire pour lui permettre d'expérimenter les différents logiciels avec lesquels il devra se familiariser. À moins d'avis contraire, tous les **examens**, **exercices** et **travaux** se feront **individuellement**.

DESCRIPTION DE LA COMPÉTENCE

Code et énoncé de la compétence	Contexte de réalisation
00Q6 - Exploiter les principes de la programmation orientée objet	<ul style="list-style-type: none"> • À partir d'un problème • À l'aide de règles de nomenclature et de codage
Éléments de la compétence	Critères de performance
1. Analyser le problème	<ul style="list-style-type: none"> • Détermination correcte des attributs et des méthodes des classes • Application judicieuse des principes d'encapsulation et d'héritage • Représentation graphique correcte des classes et de leurs relations • Respect des règles de nomenclature
2. Modéliser les classes	<ul style="list-style-type: none"> • Détermination adéquate des opérations nécessaires pour chaque méthode • Détermination correcte d'une séquence logique des opérations • Vérification appropriée du fonctionnement des algorithmes • Représentation correcte des algorithmes
3. Produire les algorithmes pour les méthodes	<ul style="list-style-type: none"> • Détermination adéquate des opérations nécessaires pour chaque méthode • Détermination correcte d'une séquence logique des opérations • Vérification appropriée du fonctionnement des algorithmes • Représentation correcte des algorithmes
4. Programmer des classes	<ul style="list-style-type: none"> • Choix approprié des instructions, des types de données élémentaires et des structures de données • Organisation logique des instructions • Programmation correcte des messages à afficher à l'utilisatrice ou à l'utilisateur • Intégration correcte des classes dans le programme • Fonctionnement correct du programme • Respect de la syntaxe du langage

	<ul style="list-style-type: none"> • Respect des règles de codage
5. Documenter la programmation	<ul style="list-style-type: none"> • Notation claire de commentaires dans le code informatique • Notation claire de la documentation d'aide à la programmation • Utilisation appropriée des générateurs de documentation
6. Appliquer la procédure liée à la gestion des versions de programmes	<ul style="list-style-type: none"> • Configuration correcte du système de gestion de versions. • Soumission méthodique du code modifié. • Gestion judicieuse des branches et des conflits.

VUE SYNOPTIQUE DU COURS

Éléments de compétence/Objectifs d'apprentissage	Contenu	Nombre d'heures
Analyser le problème	<ul style="list-style-type: none"> • Modélisation objet • Diagrammes de classes • Pseudocodes • Diagramme de séquence • Diagramme d'état/transition 	Environ 12 heures.
Modéliser les classes	<ul style="list-style-type: none"> • Structure interne d'un objet et attributs • Interface de programmation d'une classe et méthodes • Notion de membre de classe et membre d'instance • Modificateur d'accès aux membres • Espace de noms • Héritage et abstraction • Polymorphisme • Diagramme de classes • Convention de nomenclature 	Environ 12 heures.
Produire les algorithmes pour les méthodes	<ul style="list-style-type: none"> • Pseudocodes • Diagramme de séquence • Diagramme état/transition • Application de jeux d'essais 	Environ 12 heures.
Programmer des classes	<ul style="list-style-type: none"> • Langage de programmation orienté objet Java • Ramasse-miette • Syntaxe du langage • Mots clés réservés • Outils de test unitaires JUnit • Accesseurs et mutateurs • Normes de programmation • Bonnes pratiques de programmation 	Environ 15 heures.
Documenter la programmation	<ul style="list-style-type: none"> • Natures des commentaires • Balises de commentaires pour la génération de documentation 	Environ 3 heures.

	<ul style="list-style-type: none"> • Générateurs de documentation 	
Appliquer la procédure liée à la gestion des versions de programmes	<ul style="list-style-type: none"> • Gestionnaires de versions (Git) • Connection à un gestionnaire de version • Obtention des sources du dépôt • Gestion des conflits 	Environ 3 heures.

CALENDRIER

Semaine	Contenu	Activités d'apprentissage et évaluations formatives (F)	Évaluations sommatives	Travail à faire avant le prochain cours
1	<p>Présentation du plan de cours</p> <p>Présentation et préparation des outils de développement et de modélisation :</p> <ul style="list-style-type: none"> • IntelliJ IDEA pour la programmation • Draw.io pour la modélisation <p>Historique et présentation du langage Java 17</p> <p>Environnement de développement Java (JRE, JVM, JDK etc.)</p>			S'assurer que les outils de développement et de modélisation soient correctement configurés et fonctionnel.
2-3	<p>Différenciation entre la programmation orientée-objet et la programmation traditionnelle</p> <p>Éléments de l'environnement de programmation</p> <p>Historique et présentation du langage UML 2.0</p> <p>Le langage Java</p> <p>Historique et présentation de la conception orienté objet</p> <p>Terminologie associée à la programmation orientée objet :</p> <ul style="list-style-type: none"> • Objet, attributs, méthode • Classe d'objets, instance • Encapsulation, polymorphisme • Héritage <p>Terminologie associée à la modélisation orientée objet :</p>	Laboratoire 1	Laboratoire 2	<p>Terminer le laboratoire formatif 1.</p> <p>Terminer le laboratoire sommatif 2 (à la fin des 2 semaines).</p>

	<ul style="list-style-type: none"> • Acteur • Diagrammes • Membres • Portée <p>Notions de classes :</p> <ul style="list-style-type: none"> • Les membres d'instances • Les membres de classes • Le passage des paramètres par référence • L'autoréférence « this » • Les constructeurs • L'encapsulation des membres • La référence « null » • Les paquetages • Les accesseurs (« getter » et « setter ») • La méthode de comparaison « equals » <p>Les collections</p> <p>Les classes importantes du langage Java</p> <p>Les entrées et les sorties</p> <p>La méthode de sérialisation « toString »</p>			
4	Évaluation 1			
5-6	<p>L'héritage</p> <ul style="list-style-type: none"> • La dérivation • L'accès aux membres de la classe parent • La redéfinition des membres d'une classe • La dérivation en chaîne <p>La classe « Object »</p> <p>Les méthode « hashCode » et « equals »</p>	Laboratoire 3	Laboratoire 4	Terminer les laboratoire 3 et 4 avant l'examen 1.
7	Évaluation 2		Examen 1	
8	<p>Le polymorphisme</p> <ul style="list-style-type: none"> • Le type des instances de classe héritée 	Exercices Apprentissages actifs	Laboratoire 5	Terminer le laboratoire formatif 5 (à la fin des 2 semaines).

	<ul style="list-style-type: none"> • Les valeurs de retour covariantes • Le constructeur obligatoire • L'encapsulation et l'altération de l'accès aux membres • Les classes abstraites et les interfaces 	Travaux pratique		
9	La gestion des exceptions <ul style="list-style-type: none"> • L'exception explicite « Exception » • Le lancement des exceptions • La capture des exceptions • La gestion des exceptions • Les exceptions personnalisées 			
10	Les collections <ul style="list-style-type: none"> • Les listes • Les cartes • Les ensembles 		Laboratoire 6	Terminer le laboratoire sommatif 6.
11	Évaluation 3			
12	Les structures de données <ul style="list-style-type: none"> • Pile, file, liste et arbre • Performance et utilisation 		Laboratoire 7	Terminer le laboratoire sommatif 7.
13	Évaluation 4			

14	<p>Les tests</p> <ul style="list-style-type: none"> • Les assertions • Les tests unitaires avec JUnit <p>Documentation</p> <p>Gestion des versions</p>	Laboratoire 9		Terminer le laboratoire sommatif 9 (à la fin des 2 semaines).
15	Révision et ETC			

** L'horaire peut être sujet à changement

STRATÉGIES D'ENSEIGNEMENT ET D'APPRENTISSAGE

Présentations des différentes notions théoriques en classe.
Démonstration à faire en classe.
Exercices et laboratoires d'applications pratiques à faire en classe et à la maison.

ACTIVITÉS D'ÉVALUATION SOMMATIVE¹

Élément de la compétence ou compétence ciblée	Nature de l'activité/tâches	Modalités/consignes	Critères d'évaluation	Date	Pondération (%)
1. Analyser le problème 2. Modéliser les classes	Laboratoire 2 – Les base du langage Java	Individuellement, à l'aide d'un ordinateur de bureau.	<ul style="list-style-type: none"> Décomposition du problème en fonction des exigences de l'approche orientée objet Détermination correcte des données d'entrée, des données de sortie et de la nature des traitements Détermination juste des classes à modéliser Détermination correcte des algorithmes à produire 	Environ à la semaine 4	10%

¹ Article 6.5.1.1 de la PIEA L'évaluation sommative des apprentissages porte sur les compétences et les éléments de compétence précisés au plan de cours. Elle peut aussi considérer les buts généraux du programme d'études, le profil de sortie ou les compétences transversales liées à la formation. Toute activité d'évaluation sommative doit donc permettre de mesurer le niveau d'atteinte des compétences. Le calendrier du cours doit prévoir la remise des résultats d'au moins une évaluation sommative avant la date d'abandon.

Article 6.5.1.2 de la PIEA Si, en raison de la nature du cours, aucune évaluation sommative peut être remise avant la date limite d'abandon, une évaluation formative est nécessaire afin de permettre à l'étudiante ou l'étudiant de se situer face à sa réussite et aussi profiter des services offerts par le Collège, si nécessaire.

			<ul style="list-style-type: none"> • Détermination correcte des attributs et des méthodes des classes • Application judicieuse des principes d'encapsulation et d'héritage • Représentation graphique correcte des classes et de leurs relations • Respect des règles de nomenclature 		
2. Modéliser les classes 3. Produire les algorithmes pour les méthodes 5. Programmer des classes	Laboratoire 4 – L'héritage	Individuellement, à l'aide d'un ordinateur de bureau.	<ul style="list-style-type: none"> • Détermination correcte des attributs et des méthodes des classes • Application judicieuse des principes d'encapsulation et d'héritage • Représentation graphique correcte des classes et de leurs relations • Respect des règles de nomenclature • Détermination adéquate des opérations nécessaires pour chaque méthode • Détermination correcte d'une séquence logique des opérations 	Semaine 8	15%

			<ul style="list-style-type: none"> • Vérification appropriée du fonctionnement des algorithmes • Représentation correcte des algorithmes • Choix approprié des instructions, des types de données élémentaires et des structures de données • Organisation logique des instructions • Programmation correcte des messages à afficher à l'utilisatrice ou à l'utilisateur • Intégration correcte des classes dans le programme • Fonctionnement correct du programme • Respect de la syntaxe du langage • Respect des règles de codage 		
1. Analyser le problème 2. Modéliser les classes 3. Produire les algorithmes pour les méthodes	Laboratoire 6	Individuellement, en classe , à l'aide d'un ordinateur de bureau.	<ul style="list-style-type: none"> • Décomposition du problème en fonction des exigences de l'approche orientée objet • Détermination correcte des données 	Semaine 10	20%

5. Programmer des classes			<p>d'entrée, des données de sortie et de la nature des traitements</p> <ul style="list-style-type: none"> • Détermination juste des classes à modéliser • Détermination correcte des algorithmes à produire • Détermination correcte des attributs et des méthodes des classes • Application judicieuse des principes d'encapsulation et d'héritage • Représentation graphique correcte des classes et de leurs relations • Respect des règles de nomenclature • Détermination adéquate des opérations nécessaires pour chaque méthode • Détermination correcte d'une séquence logique des opérations • Vérification appropriée du 	
---------------------------	--	--	---	--

			fonctionnement des algorithmes <ul style="list-style-type: none"> • Représentation correcte des algorithmes • Choix approprié des instructions, des types de données élémentaires et des structures de données 		
3. Produire les algorithmes pour les méthodes 5. Programmer des classes 6. Documenter la programmation	Laboratoire 6 – Les collections	Individuellement, à l’aide d’un ordinateur de bureau.	<ul style="list-style-type: none"> • Détermination correcte des attributs et des méthodes des classes • Application judicieuse des principes d’encapsulation et d’héritage • Représentation graphique correcte des classes et de leurs relations • Respect des règles de nomenclature • Choix approprié des instructions, des types de données élémentaires et des structures de données • Organisation logique des instructions • Programmation correcte des messages à afficher à l’utilisatrice ou à l’utilisateur 	Semaine 11	15%

			<ul style="list-style-type: none"> • Intégration correcte des classes dans le programme • Fonctionnement correct du programme • Respect de la syntaxe du langage • Respect des règles de codage • Notation claire de commentaires dans le code informatique • Notation claire de la documentation d'aide à la programmation • Utilisation appropriée des générateurs de documentation 		
Épreuve terminale de cours		Voir page suivante		Semaine 14-15	40%

ÉPREUVE TERMINALE DE COURS

Épreuve terminale de cours attestant de l'atteinte de la compétence

Objectif terminal de cours	Contexte de réalisation de la tâche
Appliquer une approche de développement par objet dans le cadre de la réalisation de programmes informatiques.	Modalités À partir du document d'analyse fournit et du code existant, analyser, déterminer, codifier, tester et documenter les modèles objets. Utiliser

Présentation de la tâche	les éléments associés à la programmation orientée objet et au langage de programmation Java pour mettre en œuvre les solutions proposées. Consignes <ul style="list-style-type: none">• Le travail se fait individuellement sous forme d’examen pratique.• L’examen se fait en classe pendant les heures de cours.• Les consignes de l’épreuve et l’énoncé spécifique des besoins seront fournis au moment de la réalisation.• L’enseignant est présent pour clarifier les consignes et l’énoncé et répondre aux questions des étudiants qui s’y rapportent. Moment et durée Entre la 13 et la 15 semaine, environ 6 heures.	
En tant que programmeur pour la compagnie TooSoft Technologies, vous devez réaliser des fonctionnalités pour un ou plusieurs logiciels à partir de l’énoncé des besoins.		
Critères d’évaluation et pondération		
Analyser le problème		Environ 5%
Modéliser les classes		Environ 20%
Produire les algorithmes pour les méthodes		Environ 20%
Programmer des classes		Environ 40%
Documenter la programmation		Environ 10%
Appliquer la procédure liée à la gestion des versions de programmes		Environ 5%
Pour un total de 40% de votre session.		

MÉDIAGRAPHIE

DELANNOY, Claude, *Programmer en Java 5e édition*, Eyrolles, 2008.

DEITEL, DEITEL, *Java comment programmer*, Les éditions Reynald Goulet, 2002.

DELANNOY, Claude, *Exercices en Java*, Eyrolles, 2003.

TASSO, Anne, *Le livre de Java premier langage*, 2003.

LEFRANÇOIS, Éric, *Manuel Java*, document .PDF, 2000.

LEFRANÇOIS, Éric, *Programmation objet avec Java*, document .PDF, 2002.

LEFRANÇOIS, Éric, *Java - Syntaxe de base Un abrégé*, document .PDF, 2002.

GARIN, Florent, *ANDROID Apprenez à développer efficacement pour le leader des OS mobiles*, 2012

MATÉRIEL OBLIGATOIRE

L'étudiant devra avoir accès à un ordinateur utilisant le système d'exploitation Windows 11.

EXIGENCES DU COURS

Règles d'évaluation propres au programme

Les REAPP sont disponible au département et en ligne.

Évaluation de la langue d'enseignement

Les enseignants tiennent compte du français écrit dans leur évaluation. Les fautes sont soulignées et une pénalité pouvant aller jusqu'à 20% de la note peut être appliquée.

Règles relatives au respect de l'intégrité intellectuelle

Le respect de l'intégrité intellectuelle est l'une des conditions essentielles à la réussite éducative de l'étudiante ou de l'étudiant. Tout plagiat ou toute tricherie, toute tentative de plagiat ou de tricherie, toute collaboration à un plagiat ou à une tricherie sont interdits et sanctionnés.

Le plagiat consiste à s'attribuer, entièrement ou en partie, la production d'autrui, quel qu'en soit le support (physique, numérique, etc.).

La tricherie se définit comme étant toute pratique malhonnête ou trompeuse se déroulant dans le cadre d'évaluations sommatives.

L'enseignante ou l'enseignant rencontre et informe la ou les personnes étudiantes soupçonnées de plagiat ou de tricherie, afin d'obtenir leur version des faits. À la suite de la rencontre, la personne enseignante informe la ou les personnes étudiantes de sa décision et, si la tricherie ou le plagiat sont avérés, elle avise de la sanction appliquée et des sanctions prévues en cas de récidive dans son cours.

S'il s'agit d'une première offense dans un cours, l'étudiante ou l'étudiant obtient la note zéro pour le travail en cause

« la personnes concernée » peut faire appel en demandant une révision de note.

(Article 6.17 de la Politique institutionnelle d'évaluation des apprentissages)

Règles de présentation matérielle

Les travaux devront être remis sur Teams.

Les travaux doivent respecter impérativement la nomenclature exigée.

Règles pour la remise en retard des travaux

Une pénalité sera imposée aux travaux remis en retard (10% par jour de retard), pouvant aller jusqu'à l'attribution de la note zéro (0) après cinq (5) jours ouvrables.

Règles de retard au cours

L'enseignant peut refuser à sa discrétion, l'entrée aux personnes étudiantes arrivant après l'heure de début du cours.

Règles de présence aux cours

Parce que le Collège reconnaît l'importance de la présence aux cours pour l'atteinte des compétences, la présence aux cours est obligatoire pour toutes les activités synchrones qu'elles soient en présence ou en ligne. L'étudiante ou l'étudiant qui a cumulé plus de 10 % de périodes d'absence dans un cours peut être exclu de ce cours.

Il revient à l'étudiante ou l'étudiant de prendre les mesures pour communiquer avec son enseignante ou son enseignant et lui expliquer les motifs de son absence, pièces justificatives à l'appui.

(Article 6.6 de la Politique institutionnelle d'évaluation des apprentissages)

S'il y a lieu, préciser les modalités particulières d'application de cette règle dans le respect des règles d'évaluation propres au programme.

Modalités de révision de note en cours de session

Suivant la transmission de la note de son évaluation, la personne étudiante a quatre jours ouvrables pour présenter une demande de révision de note. Pour ce faire, elle se procure un formulaire auprès des Services aux étudiants ou sur ColNET et transmet le formulaire complété ainsi que toute la documentation pertinente relativement à sa demande aux Services aux étudiants.

(Article 6.15.1 de la Politique institutionnelle d'évaluation des apprentissages)

Modalités de révision de la note finale

L'étudiante ou l'étudiant qui désire soumettre une demande de révision de sa note finale doit le faire dans les quatre jours ouvrables qui suivent l'émission du bulletin officiel par le Collège. Pour ce faire, elle ou il se procure le formulaire auprès des Services aux étudiants ou sur ColNET et transmet le formulaire complété ainsi que toute la documentation pertinente relativement à sa demande aux Services aux étudiants.
(Article 6.15.2 de la Politique institutionnelle d'évaluation des apprentissages)

INFORMATIONS RELATIVES AU VIVRE ENSEMBLE AU COLLÈGE D'ALMA

Le Collège d'Alma a adopté différents règlements et politiques qui encadrent le vivre ensemble au Collège. Vous en trouverez ici les principaux éléments. Pour une lecture plus approfondie des documents, référez-vous à chacun des documents, disponibles sur le site web du Collège : [Mon Collège numérique – Plateforme étudiants – Documents numériques](#)

Règles relatives aux conditions de vie

Il est interdit à une personne en état d'ébriété de se présenter sur les propriétés du collège. (Règle 3.3.1.2)

Il est interdit à une personne sous l'effet des drogues de se présenter sur les propriétés du collège. (Règle 3.3.2.2)

« Toute personne étudiante » qui contrevient à une de ces dispositions est passible des sanctions et mesures suivantes selon la situation : un avertissement, une expulsion, une suspension ou un renvoi. (Règle 6)

(Tiré du Règlement relatif aux conditions de vie du Collège)

Violence et harcèlement

Dans un esprit de prévention, toute « personne » devrait signaler à un membre du personnel ou à la Direction des ressources humaines et des affaires corporatives une situation de violence ou de harcèlement dont « elle » a été témoin.

Le Collège s'engage à :

- Ne tolérer aucune manifestation de violence ou de harcèlement en ses lieux;
- Prendre les moyens nécessaires pour prévenir et enrayer celle-ci;
- Favoriser un climat de confiance, de compréhension et de respect mutuel.

(Tiré de la Politique pour un Collège sans violence ni harcèlement)

Violences à caractère sexuel

Le Collège d'Alma s'engage à mettre en œuvre les mesures nécessaires pour prévenir et contrer les gestes de violence à caractère sexuel. La violence à caractère sexuel comprend toute forme de violence commise par le biais de pratiques sexuelles ou en ciblant la sexualité. (Politique visant à prévenir et à combattre les violences sexuelles)

Elle contrevient à la notion de consentement (accord explicite, libre et volontaire d'une personne de se livrer à une activité sexuelle). Le consentement peut être retiré en tout temps. Le consentement est invalide dans les énumérés à l'article 5.f de la Politique visant à prévenir et à combattre les violences sexuelles.

La Direction des ressources humaines et des affaires corporatives reçoit les plaintes et en assure le suivi rigoureux et efficace. Ce service peut être joint par les moyens suivants :

Site web du Collège : Mon collège numérique – plateforme étudiants – icône jeveuxenparler

Courriel : jeveuxenparler@collegealma.ca

Téléphone : 418 668-2387, poste 216

Local : P2-683 Direction des ressources humaines et des affaires corporatives (heures d'ouverture du Collège)

Le Collège s'engage à interdire :

- toute forme de violence à caractère sexuel à l'égard d'un membre de la communauté collégiale;
- toute forme de représailles à l'égard de l'auteur d'une plainte, d'un signalement ou d'une dénonciation;
- toute relation intime (amoureuse ou sexuelle) entre un étudiant ou une étudiante du Collège et un membre du personnel qui se trouve ou pourrait se retrouver dans une relation d'autorité, pédagogique ou d'aide.

(Tiré des principes directeurs de la Politique visant à prévenir et à combattre les violences sexuelles)