



420.B0 — Techniques de l'informatique

Département : Informatique

[Lien vers les  
mesures d'aide du  
Cégep Garneau](#)

# **Plan de cours pour la session Automne 2025**

## **420-08C-FX Projet d'approfondissement en programmation**

### **0-3-3**

Professeur : Matthieu Hermet  
Bureaux : G-4842  
Téléphones : 418-688-8310, poste 3445  
Courriels : [mhermet@cegepgarneau.ca](mailto:mhermet@cegepgarneau.ca)

Professeur : Olivier Tremblay  
Bureaux : G-4843  
Téléphone : 418-688-8310, poste 3415  
Courriel : [otremblay@cegepgarneau.ca](mailto:otremblay@cegepgarneau.ca)

Coordination : Christopher Coulombe  
Bureau : G-4845  
Téléphone : 418-688-8310, poste 3240  
Courriel : [ccoulombe@cegepgarneau.ca](mailto:ccoulombe@cegepgarneau.ca)

# 1. Présentation du cours

## Description générale

Ce cours permet à l'étudiant.e de développer une application de son choix incluant du son, des images et de la vidéo. Il choisit les fonctionnalités à développer, les technologies utilisées ainsi que l'organisation du projet. Il met à profit les connaissances acquises jusqu'à présent dans son cheminement.

## Cours préalable(s) et cours nécessitant la réussite de celui-ci

Cours préalable :

- 420-14B-FX – Programmation orientée-objet

## Contribution au projet éducatif du Cégep

Ce cours s'inscrit dans le cadre du projet éducatif du Cégep Garneau. Les apprentissages que ce cours propose favorisent chez l'étudiant.e le développement d'habiletés intellectuelles, mais aussi de compétences techniques, de savoir-être et de qualités relationnelles qui lui seront utiles tout au long de sa vie.

# 2. Compétence(s)

## Compétence(s) à atteindre dans le cadre du cours

00SR - Effectuer le développement d'applications natives sans base de données (atteinte partielle)

Cette compétence est aussi vue dans le cours suivant :

- Programmation pour mobiles (420-14D-FX);
- Conception et qualité logicielle (420-14E-FX)

Éléments de la compétence et critères de performance :

1. Analyser le projet de développement de l'application.
  - 1.1 Analyse juste des documents de conception.
  - 1.2 Détermination correcte des tâches à effectuer.
2. Préparer l'environnement de développement informatique.
  - 2.1 Installation correcte des logiciels et des bibliothèques sur la plateforme hôte.
  - 2.2 Configuration appropriée du système de gestion de versions.
3. Générer ou programmer l'interface graphique.
  - 3.1 Choix et utilisation appropriés des éléments graphiques pour l'affichage et la saisie.
  - 3.2 Intégration correcte des images.
4. Programmer la logique applicative.
  - 4.1 Programmation correcte des interactions entre l'interface graphique et l'utilisatrice ou l'utilisateur.
  - 4.2 Intégration précise des sons et des vidéos.
7. Rédiger la documentation.
  - 7.1 Détermination correcte de l'information à rédiger.
  - 7.2 Notation claire du travail effectué.

### **Principaux objets d'apprentissage et contexte de réalisation**

Les savoirs et les savoir-faire sont détaillés dans le déroulement du cours.

#### Savoir-être

Dans ce cours, l'étudiant.e devra démontrer son autonomie : Capacité à prendre en charge une situation donnée en mobilisant mes ressources de façon à poser des actions au moment opportun et dans un contexte déterminé.

#### Contexte de réalisation sur le marché du travail

- À l'aide d'images, de sons et de vidéos.
- À l'aide d'un compilateur conçu pour la plateforme cible, d'un compilateur croisé ou d'un interpréteur.
- À l'aide d'un émulateur sur la plateforme hôte.

- À l'aide de procédures de suivi des problèmes et de gestion des versions.
- À partir des documents de conception.
- Pour différentes plateformes cibles : tablettes, téléphones intelligents, ordinateurs de bureau, etc.
- Pour de nouvelles applications et des applications à modifier.

#### Contexte de réalisation de la formation

- À l'aide d'ordinateurs portables.
- En classe portable.
- Langues et environnements de développement permis : au choix de l'étudiant.e sous approbation du professeur. Doit être un langage orienté-objet et permettre la production d'un exécutable fenêtré (.exe) sous Windows.
- Gestion de projet par itération.

### 3. Déroulement du cours

#### **Stratégies ou méthodes pédagogiques utilisées**

Ce cours est un cours Projet, dans lequel l'étudiant.e doit réaliser un projet de développement d'une application, seul ou en équipe de deux, suivant 3 étapes :

1. dans un premier temps, définir un projet en respect des critères nécessaires et suffisants.
2. dans un deuxième temps, œuvrer à la réalisation échelonnée du projet durant 3 sprints de développement.
3. dans un dernier temps, porter la finition du projet au point de production d'un exécutable testable et produire la documentation requise.

L'utilisation d'outils de gestion de projet est centrale à la stratégie pédagogique du cours. Un outil de planification permettra le découpage et la planification de tâches suivant un calendrier validé par le professeur au fil des sprints. Un outil de contrôle de version permettra de suivre l'accomplissement itératif du travail d'implémentation.

À noter que le professeur guidera l'étudiant.e dans la gestion de son projet tout au long de la session. Le rôle du professeur n'est pas d'être une référence technique qui débogue les étudiant.es mais plutôt un conseiller pour la gestion de projet.

## Règles de vie dans le cadre des cours

### 1. Les comportements attendus en classe.

- Ponctualité et respect des échéanciers : arriver en retard à un cours en présentiel est plus dérangeant que lorsqu'il ne s'agit que de se joindre à une classe virtuelle.
- Écoute et participation active : autant à l'écran qu'en présentiel, il est important de s'impliquer pleinement dans les cours, sans partager notre attention vers d'autres éléments.
- Respect : que ce se soit dans des conversations entre personnes en présence ou dans les communications électroniques, il est essentiel de faire preuve de respect. Cela se démontre aussi en ayant un comportement qui favorise le maintien d'un climat de classe favorable à l'apprentissage. Cela doit également se manifester à l'extérieur de la salle de classe, lors de vos travaux notamment à la bibliothèque, à la cafétéria, dans la salle des pas perdus, etc.

### 2. La **Nétiquette Garneau** qui s'applique lors de l'usage de technologies de l'information et des communications.

Pour en savoir plus, voir la section suivante : [Outils numériques : la Nétiquette](#).

## Calendrier des activités

Le tableau ci-dessous présente les contenus et activités pédagogiques et d'apprentissage telles que réparties sur l'ensemble des cours de la session. Il vous est présenté à titre indicatif et peut faire l'objet de modifications durant la session conformément à l'article 6.4 de la [Politique institutionnelle d'évaluation de l'apprentissage](#).

Semaine	Contenus	Activités d'apprentissage	Évaluations sommatives
1-2	Planification du développement d'une application <ul style="list-style-type: none"> <li>• Objectifs et contraintes</li> <li>• Fonctionnalités générales</li> <li>• Notion de récit utilisateur.</li> </ul>	Choisir une application à développer.  Respecter les exigences techniques et fonctionnelles requises.	<b>Semaine 1 :</b> <b>Énoncé de la Planification du projet</b>  <b>Semaine 2 :</b> <b>Validation du sujet</b>

Semaine	Contenus	Activités d'apprentissage	Évaluations sommatives
		Créer un récit utilisateur permettant d'identifier les fonctionnalités nécessaires à la réalisation du projet.	
3	<p>Outil de gestion de projet</p> <p>Outil de dépôt de code distant</p> <p>Développement agile avec scrum :</p> <ul style="list-style-type: none"> <li>Élaboration du Backlog produit;</li> <li>Priorisation des récits utilisateurs;</li> <li>Notions de base de tâches et sprints : <ul style="list-style-type: none"> <li>Estimation de l'effort/tâche</li> <li>Hierarchisation et planification des Sprints.</li> </ul> </li> </ul>	<p>Utiliser un outil de gestion de projet.</p> <p>Installer les logiciels et librairies nécessaires.</p> <p>Produire un document d'installation et de configuration pour l'environnement de développement.</p> <p>Initialiser le dépôt distant et attribuer les droits requis.</p>	<p><b>Semaine 3 :</b> <b>Énoncé du Déroulement des Sprints</b></p> <p><b>Remise de la Planification du projet</b></p>
4-6	<p>Normes de commentaires du code</p> <p>Nomenclatures des classes et méthodes</p>	<p>Planifier l'ordre et la durée des tâches.</p> <p>Développer les fonctionnalités :</p> <ul style="list-style-type: none"> <li>Programmer l'application ;</li> <li>Gérer les erreurs et bien les communiquer à l'utilisateur ;</li> <li>Implémenter de la rétroaction pour l'utilisateur ;</li> <li>Choisir les contrôles appropriés ;</li> <li>Intégrer des images ;</li> <li>Intégrer des sons ;</li> </ul>	<p><b>Semaine 4 :</b> <b>Rencontre de planification du Sprint 1</b></p> <p><b>Semaine 5 :</b> <b>Suivi du Sprint 1</b></p> <p><b>Semaine 6 :</b> <b>Présentation du Sprint 1</b> <b>Planification du Sprint 2</b></p>

Semaine	Contenus	Activités d'apprentissage	Évaluations sommatives
		<ul style="list-style-type: none"> <li>Écrire des commentaires pertinents pour les entêtes de fonctions et classes en respectant les normes ;</li> <li>Écrire des commentaires précis pour du code plus complexe.</li> </ul> <p>Utiliser le dépôt distant.</p> <p>Mettre à jour l'outil de gestion de projet.</p> <p>Présenter les fonctionnalités développées dans le sprint.</p>	
7		<i>Idem au sprint 1</i>	Suivi du sprint 2
	Semaine de lecture		
8			Présentation du Sprint 2 Planification du Sprint 3
9 à 11		<i>Idem au sprint 1</i>	<u>Semaine 10</u> : Suivi du sprint 3  <u>Semaine 11</u> : Présentation du Sprint 3
12 à 14		<p>Finaliser le développement de l'application :</p> <ul style="list-style-type: none"> <li>Corriger les bogues ;</li> <li>Compléter les interfaces graphiques</li> </ul>	

Semaine	Contenus	Activités d'apprentissage	Évaluations sommatives
		<ul style="list-style-type: none"> <li>Gérer les erreurs et bien les communiquer à l'utilisateur ;</li> <li>Implémenter de la rétroaction pour l'utilisateur ;</li> <li>Écrire des commentaires pertinents pour les entêtes de fonctions et classes en respectant les normes ;</li> <li>Écrire des commentaires précis pour du code plus complexe.</li> </ul> <p>Produire un fichier exécutable sous Windows (.exe)</p> <p>Intégrer des ressources d'aide à l'utilisateur dans l'application (vidéo d'aide).</p> <p>Produire de la documentation d'aide au développement.</p> <p>Utiliser le dépôt distant.</p> <p>Mettre à jour l'outil de gestion de projet.</p>	
15			Présentations finales des projets et remise finale
16	Réservée en cas de besoin		



Cliquez ou appuyez ici pour entrer du texte.

## 4. Activités d'évaluation

### Évaluation formative

Dans un cours Projet, l'évaluation formative est fortement associée à l'évaluation sommative du fait de la complémentarité des parties successives.

Chaque évaluation fait l'objet d'une indication sur le travail à fournir en vue de l'évaluation suivante. Ces indications portent autant sur les jalons du projet dans son ensemble que sur les incomplétudes et échecs dans la remise courante, et ont ainsi une valeur formative, donnant à l'étudiant.e la possibilité de se reprendre et de revenir sur ses erreurs.

### Évaluation sommative

Le projet développé par les étudiant.es se déroule tout au long de la session. Le choix du sujet doit être approuvé par le professeur dès la deuxième semaine.

Les étudiant.es ont la responsabilité de choisir un sujet conforme aux requis énoncés dans un document initial de description du projet au plan sommatif. Les 3 sprints couvriront les phases de développement du projet. La remise finale portera sur la finition et sur la documentation du projet et ne devra pas inclure de fonctionnalités nouvelles.

Le tableau ci-dessous renferme la liste des activités d'évaluation sommative qui auront lieu au cours de la session ainsi que leur pondération.

Activités d'évaluation sommative	Contenu	Semaine de remise	Pondération
Présentation du projet	Description du projet. Langage et IDE. Récits utilisateurs. Liens du calendrier et du dépôt distant. Initialisation du calendrier en vue de l'évaluation au sprint 1.	3	10%

Activités d'évaluation sommative	Contenu	Semaine de remise	Pondération
<b>Sprint 1</b>	Complétude de la remise par rapport au calendrier. Vérification des fonctionnalités. Prévision des étapes suivantes.	6	15%
<b>Sprint 2</b>	Idem au sprint 1.	9	20%
<b>Sprint 3</b>	Idem au sprint 1.	12	20%
<b>Remise finale</b>	Finition du programme (documentation, commentaires, correction des bogues). Inclusion video. Validation (projet abouti).	15	35%

### Informations détaillées relatives aux activités d'évaluation sommative

#### Éléments du Projet.

**Planification du projet** : la planification du projet est un travail fondamental et doit faire l'objet d'une réflexion particulière. Pour rappel, ce travail doit porter principalement sur les aspects généraux suivants :

1. Récit Utilisateur
2. Description des Fonctionnalités
3. Environnement et Langage de développement

D'autres éléments possibles de planification seront présentés dans l'énoncé.

**Projet** : Le projet doit satisfaire à certains critères de quantité et contenir au minimum un élément video sous forme d'aide à l'utilisateur, un élément média (image et/ou son), un aspect de personnalisation de l'interface par le choix d'options spécifiées par l'utilisateur dans un formulaire et consignée dans un fichier de préférences.. Au moins une classe du programme doit suivre le principe de composition.

L'étudiant.e doit ajouter un ou plusieurs éléments optionnels selon les besoins de son projet, qui devront être annoncés dans le document de planification du projet en première remise. Ces éléments peuvent être graphiques (dessin 2D, génération par le code, etc.) ou opératoires sur les données (Recherche, tri, versionnage culturel, etc.).

L'étudiant.e devra choisir un langage de programmation et un environnement de développement. Son projet pourra inclure des modules tierces et des ressources externes au besoin, sous réserve d'acceptation par le professeur.

Le développement lui-même se fera lors des 3 sprints de développement.

Un dernier sprint devra permettre la finition et la documentation du projet.

### **Modalités de rétroaction des évaluations**

Il y aura une rétroaction à la première remise dans un délai d'une semaine.

Avant chaque sprint (le calendrier des sprints sera fourni avec l'énoncé), l'équipe ou l'étudiant.e doit remettre un court document indiquant :

1. l'objectif du sprint et
2. les conditions satisfaisantes à sa complétion.

À noter que le premier document de planification du projet vaudra pour le projet en général ainsi que pour le premier sprint en particulier.

À la complétion de chaque sprint, l'équipe ou l'étudiant.e doit remettre un court document incluant :

1. Une rétrospective;
2. Un diagramme des classes produites;

S'en suivra une entrevue personnalisée entre le professeur et l'étudiant.e afin de fournir une rétroaction sur la qualité et la complétion du travail accompli par rapport au calendrier et sur la base de l'état du dépôt distant, ainsi qu'une validation et/ou ajustement du calendrier prévu pour la remise suivante.

À noter qu'une entrevue de mêlée pourra avoir lieu à chaque semaine pour faire le point sur le calendrier et sur la participation de tous les membres de l'équipe.

Il y aura une rétroaction à la remise finale dans un délai de deux semaines.

## 5. Encadrement

### Modalités d'encadrement, périodes de disponibilité et autres mesures d'aide

Le professeur est disponible à son bureau d'après son horaire sur le site du [Département des techniques de l'informatique](#). Le professeur est aussi joignable sur Teams. Il sera disponible aux périodes mentionnées dans son horaire et tentera de répondre aux messages dans la demi-journée. Si les questions de l'étudiant.e le nécessitent, une visio-conférence pourra être effectuée. Si une intervention ou interaction sur la machine de l'étudiant.e est nécessaire, le professeur pourra demander un partage d'écran à l'étudiant.e.

À noter que ce cours est un cours projet nécessitant des rencontres personnalisées régulièrement, qui pourront avoir lieu en présentiel ou via Teams au choix de l'étudiant.e.

### Mesures d'aide disponibles en plus de l'encadrement offert dans le cours

Pour en savoir plus sur les mesures d'aide et les services disponibles au cégep, les étudiant.es peuvent consulter l'outil interactif [Découvre quels services peuvent t'aider](#).

## 6. Matériel pédagogique, informatique et autre

### Matériel obligatoire

Aucun livre obligatoire.

### Matériel et support informatique

Ordinateur.

Connexion internet.

Pour du support technique, les étudiant.es peuvent consulter le site internet [Outils numériques pour votre session](#).

### **Matériel facultatif**

Aucun.

### **Protection du droit d'auteur, droit à la propriété intellectuelle, droit à l'image**

Tout matériel pédagogique produit dans le cadre de ce cours, incluant, mais non limitativement, toute vidéo préenregistrée ou en direct est protégé par le droit d'auteur, par le droit à la propriété intellectuelle ainsi que par le droit à l'image, et ce, sans égard au support utilisé. Il est strictement interdit de copier, de redistribuer, de reproduire, de republier, d'emmagasiner sur tout médium, de retransmettre ou de modifier ces documents. Toute contravention à ces conditions d'utilisation pourrait faire l'objet de sanction(s) de la part du Cégep.

## **7. Références**

### **Médiagraphie**

#### **Livres:**

Chacon, Scott, Pro Git, 2nd version, <https://git-scm.com/book/en/v2>

Cohn, Mike (2004) User Stories Applied : For Agile Software Development, Addison Wesley Professional, 304 pages

#### **Sites web:**

- Récits utilisateurs avec des exemples et un modèle : [Atlassian Agile Coach](#)
- Guide de Trello : <https://trello.com/guide>
- Référence officielle de Git : <https://git-scm.com/>
- Référence de Git : <https://www.atlassian.com/fr/git/tutorials/what-is-version-control>
- Référence GitLab : <https://docs.gitlab.com/>
- Diagramme de classe UML : <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes#L3>

## Politiques et règlements en vigueur

L'étudiant.e doit prendre connaissance de la **Politique institutionnelle d'évaluation des apprentissages (PIEA)** et du **Règlement de vie étudiante**, disponibles sur la page [Documents institutionnels](#) du site Web du Cégep Garneau.

## 8. Modalités d'application de la PIEA

Tout travail doit être remis au professeur selon les modalités indiquées au plan de cours ou sur l'énoncé du travail. Tout retard entraîne une pénalité de 10 % de la note maximale. Une pénalité équivalente est appliquée pour chaque période subséquente de 24 heures. Le travail peut aussi être refusé après un certain délai.

Après un maximum de 3 jours (72 heures) de retard, tout travail est refusé. Pour certains travaux, il est possible que le délai maximal soit moins de 3 jours (72 heures). Dans ce cas, le délai maximal doit être indiqué dans l'énoncé du travail.

En aucun cas un délai accordé pour la remise d'un travail ne peut avoir pour effet de prolonger le délai normal prévu pour la remise des notes finales (cinq jours ouvrables après la fin de la session).

### Pénalités pour les retards dans la remise des travaux

Conformément à la PIEA, tout retard entraîne une pénalité de 10 % de la note maximale par jour, et ce dès le moment prévu pour la remise du travail.

Après un maximum de 3 jours (72 heures) de retard, tout travail est refusé. Pour certains travaux, il est possible que le délai maximal soit moins de 3 jours (72 heures). Dans ce cas, le délai maximal doit être indiqué dans les consignes du travail.

### Précisions quant aux seuils de réussite

La note de passage est fixée à 60 % pour tous les cours. Cette note traduit l'atteinte minimale des objectifs et des exigences d'un cours.

De plus, pour les cours du programme Techniques de l'informatique, l'étudiant doit obtenir une moyenne minimale de 55 % dans les examens et une moyenne minimale de 55 % dans les travaux. Si un de ces deux seuils n'est pas respecté, la note maximale est de 55 %.

**Modalités de reprise telles que définies en vertu des articles 6.9 et 6.11 de la PIEA**

L'étudiant absent lors d'un examen pour un motif valable dispose d'un délai de 24 heures pour aviser son professeur.  
L'étudiant doit fournir une preuve motivant son absence.

**Plagiat et fraude scolaire tels que définis en vertu de l'article 6.16 de la PIEA**

L'étudiante ou l'étudiant doit prendre connaissance de l'article 6.16 de la [Politique institutionnelle d'évaluation des apprentissages](#) (PIEA) disponible sur le site Web du Cégep Garneau.

Les étudiantes et les étudiants sont invités à consulter la page [Le plagiat, c'est quoi?](#) du site Web du Cégep Garneau.

L'utilisation d'outils d'intelligence artificielle générative (ex. : ChatGPT, Copilot, Gemini, etc.) pour la réalisation des travaux ou examens de ce cours est strictement interdite, sauf indication contraire explicite du professeur. Toute utilisation, même partielle, sera considérée comme de la tricherie au sens de la Politique institutionnelle d'évaluation de l'apprentissage (PIEA) et pourra entraîner une note de zéro pour l'évaluation concernée, voire l'échec du cours, conformément aux dispositions de la PIEA.

**Autres précisions ou modalités**

Une étudiante ou un étudiant en retard à un examen ne sera admis que si aucun étudiant n'est sorti de la salle d'examen; dans un tel cas, l'étudiante ou l'étudiant ne disposera d'aucun temps supplémentaire. Lors des examens, aucune sortie temporaire ne sera tolérée.

En cas de fermeture imprévue du collège la journée d'un examen (grève, tempête, etc.), la reprise est systématiquement fixée au cours suivant selon l'horaire habituel.

Pour tout travail, une pénalité de 0,25 % par faute de français est appliquée jusqu'à concurrence d'un maximum de 20 %. Cependant, dans le cas d'une interface utilisateur (page Web, formulaire, etc.), la pénalité est de 1 % par faute. Ces pénalités ne s'appliquent pas lors des examens.

En cas de contradiction entre ces modalités et la PIEA, cette dernière a préséance.