

---

|                               |  |  |  |
|-------------------------------|--|--|--|
| <b>TITRE DU COURS :</b>       | Environnement graphique client/serveur     |  |  |
| <b>NUMÉRO DU COURS :</b>      | 420 4F5 MO                                 |  |  |
| <b>SESSION :</b>              | Hiver 2026                                 |  |  |
| <b>pondération du cours :</b> | 2-3-2                                      |  |  |
| <b>DÉPARTEMENT :</b>          | Techniques de l'informatique, 420.B0       |  |  |
| <b>PROGRAMME :</b>            | Développement d'applications informatiques |  |  |
| <b>AXE DE FORMATION :</b>     | Applications natives                       |  |  |

|                         | <b>Nom</b>       | <b>Local</b> | <b>Téléphone</b>      | <b>Courriel</b>                     |
|-------------------------|------------------|--------------|-----------------------|-------------------------------------|
| <b>PROFESSEURS :</b>    | Mathieu Bergeron | B3320        | (450) 975-6100 p.6727 | mathieu.bergeron@cmontmorency.qc.ca |
| <b>COORDONNATEURS :</b> | Abdelhabib Yahia | B3322        | (450) 975-6100 p.7424 | abdelhabib.yahia@cmontmorency.qc.ca |
|                         | Antoine Tohmé    | B3322        | (450) 975-6100 p.7512 | antoine.tohme@cmontmorency.qc.ca    |

---

|                                 |                                |
|---------------------------------|--------------------------------|
| <b>PLATEFORMES PÉDAGOGIQUES</b> | Moodle, ntro.ca, aiguilleur.ca |
|---------------------------------|--------------------------------|

Ce plan de cours est sujet à toutes les clauses contenues dans le document décrivant la politique institutionnelle d'évaluation des apprentissages (PIEA) du collège. Il est aussi sujet à toutes les règles et procédures pour les personnes étudiant en techniques de l'informatique. Veuillez-vous référer à ces documents pour plus d'information sur chacune des clauses spécifiques.

Département d'informatique: [https://ntro.ca/4f5/presentation/regles\\_departement\\_informatique.pdf](https://ntro.ca/4f5/presentation/regles_departement_informatique.pdf)

PIEA : <https://www.cmontmorency.qc.ca/piea>

## **1. DESCRIPTION DU COURS**

L'étudiante ou l'étudiant du aura à développer une application native qui utilise une base de données et qui offre un service d'échange de données. Il analysera le projet et préparera son environnement de développement. Il créera la base de données et programmera l'application avec son service d'échange de données installé sur un serveur. Par la suite, il procédera au contrôle de la qualité de l'application et préparera son déploiement.

## **2. OBJECTIF INTÉGRATEUR**

Développer des applications natives avec bases de données et services d'échanges de données.

## **3. OBJECTIFS MINISTÉRIELS**

00SS (éléments 1 à 8) : Effectuer le développement d'applications natives avec bases de données

00SV (éléments 1,4,5,6 et 8) : Effectuer le développement de services d'échange de données

## **4. OBJECTIFS D'APPRENTISSAGE**

1. Analyser le projet et préparer l'environnement de développement
2. Programmer l'application et le service d'échange de données
3. Contrôler la qualité de l'application, du service et de son déploiement

## **5. ATTITUDES PROFESSIONNELLES**

1. Rigueur (AP1 – Niveau 2)
2. Collaboration (AP3 – Niveau 2)

## **6. VOLETS TRANSVERSAUX**

1. Compétence langagière (VT1 – Niveau 2)
2. Compétence numérique (VT2 – Niveau 2)

## **7. PRÉALABLES**

| <i>Spécialisation Développement d'applications informatiques</i>                          |  |
|---|--|
| <b>Ce cours est préalable absolu au présent cours</b><br>420 3D4 MO Interface utilisateur | <b>Ce cours est préalable relatif au présent cours</b><br>420 3E5 MO Interface Web |
| <b>Le présent cours est préalable absolu aux cours ci-après</b>                           |  |
| 420 5J6 MO Applications Mobiles   |  |
| 420 6A6 MO Stage en développement d'applications  |  |
| 420 6B6 MO Stage : interactions professionnelles en développement d'application           |  |

## **8. INTÉGRATION DES APPRENTISSAGES**

L'étudiant devra réaliser une application native mettant en œuvre une interface graphique complexe en interaction avec une base de données selon les spécifications fonctionnelles fournies et qui offre un service d'échange de données.

L'étudiant sera évalué à partir des critères de performance suivants :

1. Détermination correcte des tâches à effectuer
2. Installation correcte des logiciels et des bibliothèques
3. Création ou adaptation correcte de la base de données
4. Respect du modèle de données
5. Choix et utilisation appropriée des éléments graphiques pour l'affichage et la saisie
6. Programmation correcte des interactions entre l'interface graphique et l'utilisatrice ou l'utilisateur
7. Respect des formats d'échange de données
8. Programmation correcte de l'envoi des données en sortie et/ou de la réception des données en entrée
9. Application rigoureuse des plans de tests
10. Déploiement ou installation corrects de l'application

## **9. OBJECTIFS DÉTAILLÉS**

Pour des raisons pédagogiques, les points suivants seront traités progressivement au cours de la session et non de façon séquentielle.

| Objectif d'apprentissage   | Savoirs essentiels  | Importance |
|--|---|------------|
| 1. Analyser le projet et préparer l'environnement de développement (00SS : éléments 1-3 et 00SV : élément 1)               | 1.1 Analyse des documents de conception<br>1.2 Détermination des tâches à effectuer<br>1.3 Collaboration et communication dans les tâches à effectuer (AP3-N2)<br>1.4 Installation et configuration de la plateforme de développement et du système de gestion de base de données<br>1.5 Création d'un projet de développement et installation des bibliothèques<br>1.6 Installation et configuration du système de gestion de versions<br>1.7 Importation du projet et du code source du système de gestion de versions vers l'environnement de développement<br>1.8 Publications des changements vers le système de gestion de versions<br>1.9 Élaboration du modèle logique de données à partir du modèle conceptuel de données et des spécifications du document d'analyse<br>1.10 Création de la base de données locale ou distante<br>1.11 Insertion des données initiales et des données de tests  | 25%        |
| 2. Programmer l'application et le service d'échange de données (00SS : éléments 4-5 et 00SV : élément 4)                   | 2.1 Programmation de la couche présentation de l'application (VT2-N2) <ul style="list-style-type: none"> <li>Création de l'interface graphiques</li> <li>Application des techniques d'affichage dynamique et adaptatif</li> <li>Intégration des images dans l'interface graphiques</li> <li>Programmation des interactions entre l'interface graphique et l'utilisateur</li> </ul> 2.2 Programmation de la logique applicative de l'application (VT2-N2) <ul style="list-style-type: none"> <li>Mécanismes d'authentification et d'autorisation</li> <li>Programmation des spécifications fonctionnelles et de la logique d'affaire de l'application selon le document d'analyse</li> <li>Requêtes à la base de données</li> <li>Technique d'internationalisation</li> <li>Techniques de programmation sécurisée</li> </ul> 2.3 Programmation du service d'échange de données (VT2-N2) <ul style="list-style-type: none"> <li>Format d'échanges de données (p.ex. XML ou JSON)</li> <li>Programmation de la réception de données d'entrée</li> <li>Programmation de l'envoi de données de sortie</li> </ul> | 45%        |
| 3. Contrôler la qualité de l'application, du service et son déploiement (00SS : éléments 6-8 et 00SV : éléments 5, 6 et 8) | 3.1 Normes de programmation<br>3.2 Plans de tests de cas de tests sur l'application<br>3.3 Revues de code et de sécurité<br>3.4 Pertinence des correctifs (AP1-N1)<br>3.5 Respect des procédures de suivi des anomalies, des correctifs et de gestion de versions<br>3.6 Respect des documents de conception<br>3.7 Programmation d'une application client simple pour tester le services <ul style="list-style-type: none"> <li>Connexion au serveur</li> <li>Récupération du service à partir du serveur</li> <li>Utilisation du services</li> <li>Conversion des données fournies par le service en données exploitables par l'application client</li> </ul> 3.8 Préparation de l'application<br>3.9 Déploiement ou installation de l'application  | 30%        |

|  |  |
|--|--|
| 3.10 Détermination de l'information à rédiger                            |  |
| 3.11 Document de réalisation et d'exploitation de l'application (VT1-N2) |  |

## 10. DÉROULEMENT DU COURS

Note : le calendrier peut être sujet à changement.

| Étape  | Semaine | Objectifs           | Contenu  | Évaluations |
|--|---------|---------------------|--|-------------|
| 1  | 1       | 1.1-1.4, 1.6        | Module 1 : <ul style="list-style-type: none"> <li>Installation Eclipse, JDK, Gradle, Git, exécution des tutoriels</li> <li>Formation des équipes</li> <li>Création du projet Eclipse</li> </ul>  | • Atelier 1 |
|  | 2       | 1.5, 1.7-1.9 et 2.2 | Module 2 : <ul style="list-style-type: none"> <li>Patron MVC (1) : création d'un frontal de base (<i>frontend</i>)</li> <li>Document de conception <ul style="list-style-type: none"> <li>maquette d'une vue par personne</li> <li>spécification d'un modèle pour chaque vue</li> <li>plan de tests pour chaque vue</li> </ul> </li> </ul> | • Atelier 2 |
|  | 3       | 2.1                 | Module 3 : <ul style="list-style-type: none"> <li>Vue en JavaFx (1) : fxml de base, css, traductions</li> </ul>  | • Atelier 3 |
|  | 4       | 2.2                 | Module 4 : <ul style="list-style-type: none"> <li>Patron MVC (2) : événements et capteurs d'événements</li> </ul>  | • Atelier 4 |
|  | 5       | 1.10-1.11 et 2.2    | Module 5 : <ul style="list-style-type: none"> <li>Notion de modèle et observation d'un modèle</li> <li>Patron MVC (3) : création d'un dorsal de base (<i>backend</i>)</li> </ul>   | • Atelier 5 |
|  | 6       | 1, 2.1-2.2, 3       | Séance 6 : travail libre   | •           |
| <b>REMISE TP #1 : avant examen #1</b><br><b>EXAMEN#1 : semaine 6, séance 2 (en classe)</b> |         |                     |  |             |
| 2  | 7       | 2.1                 | Module 6 : <ul style="list-style-type: none"> <li>Vue en JavaFx (2) : contrôles personnalisés</li> </ul>   | • Atelier 6 |
|  | 8       | 2.3                 | Module 7 : <ul style="list-style-type: none"> <li>Vue en JavaFX (3) : animations, dessins 2D (canvas)</li> </ul>   | • Atelier 7 |
|  | 9       | 2.1                 | Module 8 : <ul style="list-style-type: none"> <li>Notion de message</li> <li>Patron MVC (4) : communication client/serveur</li> </ul>  | • Atelier 8 |
|  | 10      | 2.2                 | Module 9 : <ul style="list-style-type: none"> <li>Patron MVC (5) : modèles multiples</li> </ul>  | • Atelier 9 |
|  | 11      | 1-3                 | Séance 10 : travail libre  |             |
| <b>REMISE TP #2: avant examen #2</b><br><b>EXAMEN#2 : semaine 11, séance 2 (en classe)</b> |         |                     |  |             |

|                                      |       |     |                       |  |
|--------------------------------------|-------|-----|-----------------------|--|
| 3                                    | 12-15 | 1-3 | Travail sur le projet |  |
| REMISE PROJET : semaine 15, séance 2 |       |     |                       |  |

## **11. MÉTHODOLOGIE**

### **11.1. COURS MAGISTRAUX ET LABORATOIRES**

1. Ce cours est d'une durée de 75 périodes : 30 périodes de théorie (2 périodes par semaine) et 45 périodes de laboratoire (3 périodes par semaine) selon l'horaire en vigueur.
2. La présence au cours constitue une condition déterminante de la réussite. Elle permet de placer l'étudiante et l'étudiant en situation d'apprentissage supervisé par le professeur. La présence au cours est nécessaire à l'atteinte des objectifs, elle est donc obligatoire.
3. La ponctualité est de rigueur, le professeur peut refuser l'accès en classe à une personne qui s'y présente après un retard indu.
4. Une personne absente sans raison valable ne recevra pas d'aide supplémentaire de la part du professeur sur la matière manquée. Il est de la responsabilité de l'étudiante et de l'étudiant de voir à reprendre autrement les activités manquées.
5. Pendant la partie théorique, le professeur expliquera la matière du cours.
6. Pendant les laboratoires, les étudiantes et étudiants devront tester les notions vues pendant la partie théorique et/ou travailler sur leurs travaux.
7. Une étudiante et un étudiant ne peut utiliser que les logiciels enseignés dans le cadre de ce cours et doit respecter un climat propice au travail. Dans le cas contraire, le professeur peut lui demander de quitter le local et la personne sera considérée absente.

### **11.2. STRUCTURE DU COURS**

#### **TRAVAIL SUR UNE APPLICATION**

Les étudiants vont planter une application graphique (un jeu 2D simple), en commençant à neuf et en progressant tout le long de la session. Même si le travail est individuel à la base, les étudiants qui le désire auront l'opportunité de fonctionner en équipe et de mettre leur code en commun.

#### **ÉTAPES**

Le cours est divisé en 3 étapes. Les deux premières étapes sont divisées en modules. La troisième étape consiste en un projet.

#### **MODULES**

Un module comprend les activités suivantes, que l'étudiant.e **doit effectuer dans l'ordre** :

1. Un court exposé théorique

2. Un tutoriel préparant à accomplir l'atelier
3. Un atelier
4. Une auto-évaluation
5. Entre 1 et 3 fois par étape : une validation en classe de l'atelier

### **11.3. SYSTÈME DE BILLETTERIE**

Un système de prise de rendez-vous permettra de consigner les questions des étudiant.es. Durant les séances de travail en classe, l'enseignant.e répondra aux questions dans l'ordre d'entrée dans ce système.

### **11.4. ÉVALUATIONS**

#### **EXAMENS**

Les étapes 1 et 2 se terminent chacune par un examen (à effectuer en classe sur Moodle).

#### **ATELIERS**

Les ateliers sont à remettre tout au long de l'étape (5 ateliers pour l'étape 1 et 4 ateliers pour l'étape 2). L'évaluation des ateliers se fera de trois façons : i) via la remise du code sur GitLab, ii) via une auto-évaluation à remplir par l'étudiant et iii) via une validation en classe de type « revue de code ». Chaque étudiant sera convoqué entre 1 et 3 fois par étape pour une validation en classe.

#### **TRAVAUX PRATIQUES**

À la fin des étapes 1 et 2, chaque étudiant doit remettre une version fonctionnelle de son projet. L'enseignant va évaluer cette version afin de mesurer l'atteinte des objectifs de l'étape. La remise **doit être effectuée** via le même dépôt Git que l'étudiant va créer au début du cours (aucun autre type de remise sera acceptée).

#### **PROJET**

La troisième étape consiste en un projet où l'étudiant doit finaliser son projet. Les étudiants qui désirent travailler en équipe pourront intégrer leur code, tout en spécifiant clairement la contribution de chaque étudiant (l'évaluation demeure individuelle). La remise **doit être effectuée** via Git (aucun autre type de remise sera acceptée).

### **11.5. TRAVAUX ET EXAMENS**

1. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon l'article 7.4.2, pour les activités d'évaluations, une pénalité de 10 % par tranche de 24 heures de retard sera calculée à partir de l'heure prévue de la remise. Aucun retard ne sera accepté sans raison valable après quatre jours de retard, sans égard aux congés ou dépasse le moment où la rétroaction est fournie. Si c'est le cas, la note de 0 sera attribuée.

2. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon l'article 8.1, les délais de correction ne peuvent dépasser 10 jours ouvrables (15 jours si justification) suivant la date prévue de la remise des activités d'évaluation.
3. Lorsque le professeur doute du travail effectué par une étudiante ou un étudiant, elle peut avoir recours à une vérification orale ou écrite du niveau de connaissance de la personne concernée.
4. **Chaque personne est responsable de vérifier la qualité de ses fichiers et de ses imprimés remis dans le cadre d'une évaluation.**
5. Les évaluations sommatives seront conservées par le professeur pendant 6 mois et pourront être consultées sur demande.
6. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon l'article 7.2, dans le cas d'une absence à une évaluation sommative, la personne devra fournir, au plus tard dans les deux jours ouvrables suivant la date de l'évaluation, une justification pour motiver son absence auprès du professeur. Si la raison de l'absence n'est pas valable ou qu'elle n'est pas justifiée, celle-ci se verra attribuer la note zéro pour l'évaluation concernée.
7. Lors d'un examen, la personne doit se présenter au local désigné par le Collège, et réaliser l'examen sous la surveillance d'un membre du personnel.
8. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon l'article 7.4.1, la personne en retard à une évaluation sommative peut être admise en classe ou laboratoire, mais ne peut demander une prolongation. Elle peut se voir refuser l'accès à l'évaluation si une personne a déjà terminé et quitté la salle. Le retard peut être considéré comme une absence non justifiée et la note de zéro peut être attribuée à l'évaluation.
9. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon l'article 5.4, 10 % de la note de chaque activité demandant de produire un document écrit fera référence à la qualité de la langue.

## **12. INTÉGRITÉ INTELLECTUELLE**

1. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon les articles 6.1.1, 10% peuvent être accordés par évaluation pour des fautes méthodologiques (référencement documentaire ou de présentation).
2. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon les articles 6.1.2, pour toute infraction constatée, soit **plagiat, tricherie, fraude, tentative ou collaboration** à l'un ou l'autre de ces événements entraînent la note zéro pour l'activité d'évaluation en question et un rapport d'événement sera remis au comité départemental de plagiat, qui communiquera avec vous pour une rencontre.
3. En **cas de récidive** dans le même cours, la personne fautive se verra attribuer la **mention échec** pour le cours concerné. Un nouveau rapport d'événement doit être fait et remis.
4. Comme mentionné dans la politique institutionnelle d'évaluation des apprentissages selon les articles 6.1.3, la personne qui a reçu une sanction pour une infraction et qui présume une irrégularité ou une injustice peut faire appel de la décision et dispose de cinq jours ouvrables suivant l'avis de manquement

pour déposer au Service des programmes et de l'enseignement un formulaire de demande dans lequel elle présente un argumentaire justifiant de façon explicite le motif de sa demande d'appel.

### **13. ACTIVITÉS D'ÉVALUATION FORMATIVES**

L'évaluation formative permet d'obtenir un état des lieux au cœur d'une séquence d'apprentissage qui pourra permettre, le cas échéant, des réajustements. Elle peut aussi offrir l'occasion de prévenir des échecs en mettant en relief les signaux de risques pour ainsi poser une action préventive.

| Catégorie | Activité              | Portera sur           | Semaines |
|-----------|-----------------------|-----------------------|----------|
| Tutoriels | Étape 1 : 5 tutoriels | Contenu de la semaine | 1-6      |
|           | Étape 2 : 4 tutoriels |                       | 7-11     |

### **14. ACTIVITÉS D'ÉVALUATION SOMMATIVES**

Les activités d'évaluation pourraient être modifiées durant la session

| Catégorie      | Évaluation            | Description  | Savoirs essentiels      | Principaux critères d'évaluation | Date de l'évaluation ou de la remise   | Pondération |
|----------------|-----------------------|--|-------------------------|----------------------------------|--|-------------|
| Examens (40%)  | Examen 1              | <i>patron MVC, frontal/dorsal, client/serveur, vues en JavaFx, capteurs d'événements, notions de modèle, graphe de tâches</i>                                      | Objectifs 1, 2.1-2.2, 3 | Critères 3, 4 et 7               | Semaine 6, séance 2                    | 20%         |
|                | Examen 2              | <i>vues avancées en JavaFx, contrôles personnalisés, animations, dessins 2d, tailles élastiques, messages, communication client/serveur, collection de modèles</i> | Objectifs 2-3           | Critères 5, 6, 8, 9 et 10        | Semaine 11, séance 2                   | 20%         |
| Ateliers (30%) | Étape 1<br>5 ateliers | Remise du code, auto-évaluation et validation en classe  | Objectifs 1, 2.1-2.2, 3 | Critères 1, 2, 3, 4 et 7         | Première séance de la semaine suivante | 16.66%      |
|                | Étape 2<br>4 ateliers |  | Objectifs 2-3           | Critères 5, 6, 8, 9 et 10        |  | 13.33%      |

### **Intégration des apprentissages**

|           |      |   |                         |                           |                      |    |
|-----------|------|---|-------------------------|---------------------------|----------------------|----|
| Tps (12%) | TP 1 | <i>maquette (document de conception), vue personnalisée, navigation d'une vue à l'autre, afficher d'un modèle personnalisée en mode texte, ajout de données au modèle, respect de la méthodologie du cours et qualité du code</i> | Objectifs 1, 2.1-2.2, 3 | Critères 1, 2, 3, 4 et 7  | Semaine 6, séance 2  | 6% |
|           | TP 2 | <i>affichage du modèle avec des fragments personnalisés, affichage d'un objet 2d en mouvement, communication client/serveur, respect de la méthodologie du cours et qualité</i>   | Objectifs 2-3           | Critères 5, 6, 8, 9 et 10 | Semaine 11, séance 2 | 6% |

|                         |               |  |                 |                 |                      |     |
|-------------------------|---------------|--|-----------------|-----------------|----------------------|-----|
|                         |               | <i>du code</i>   |                 |                 |                      |     |
| <b>Projet<br/>(18%)</b> | <i>Projet</i> | <i>Ajout de plusieurs fonctionnalités client/serveur, affichage peaufiné avec du CSS visible, respect de la méthodologie du cours et qualité du code</i> | Objectifs 1,2,3 | Critères 1 à 10 | Semaine 15, séance 2 | 18% |

## 15. MATÉRIEL REQUIS

- Du matériel pour écrire à la main (p.ex. crayon/papier ou tablette/stylet)
- Des écouteurs pour visionner les vidéos du cours
- Un compte sur [GitLab.com](https://gitlab.com)
- Un disque dur externe clairement identifié.