

PLAN DE COURS

Développement d'applications pour entreprises

Titre du cours

Techniques de l'informatique

Nom du ou des programme(s) ou de la composante de formation générale

Techniques de l'informatique

Discipline

420-413-MV

2-2-2

2,00

Numéro du cours

Pondération

Unités

Sara Boumehraz

S-012

sara.boumehraz@collegemv.qc.ca

Enseignant-e

Numéro de bureau

Poste téléphonique et courriel

Informatique

**Olivier Tardif
S-012**

Nom du département

Nom du coordonnateur ou de la
coordonnatrice du département

2023-2024

Hiver 2024

Année scolaire

Trimestre

PRÉSENTATION GÉNÉRALE DU COURS :

Ce cours a pour préalable le cours *Introduction aux bases de données*. Il est le troisième cours dans l'axe de formation *Génie logiciel*, qui contient aussi les cours *Introduction aux bases de données*, *Architecture de logiciel*. Il est également le deuxième cours de l'axe *Développement natif*, qui contient aussi les cours *Interfaces humain-machine*, *Développement de jeux vidéo* et *Développement d'applications mobiles*.

Dans ce cours, la personne étudiante doit utiliser un langage orienté objet ainsi que divers outils de développement afin de concevoir et développer des applications utilisant des bases de données. Ces applications doivent pouvoir être utilisées dans un contexte d'entreprise sur un ordinateur de bureau.

C'est l'occasion pour la personne étudiante de développer une application d'allure professionnelle qui devra améliorer la productivité de ses utilisateurs. Elle apprend comment développer des applications natives pleinement intégrées dans l'écosystème d'une plateforme. Elle développe en tandem une base de données et pourra ainsi acquérir une perspective renforcée sur les interconnexions entre les données et les choix d'architecture.

Le langage utilisé dans ce cours peut différer de celui utilisé dans le cours *Programmation orientée objet* (ex: ce cours utilise C# pour se servir de .NET, mais les étudiants sont familiers avec Java). Puisque ce cours en est un de 4e session, la personne étudiante est en mesure de démontrer une certaine autonomie dans l'apprentissage de nouveaux langages, mais le survol de la syntaxe de celui-ci reste un élément de contenu essentiel.

Le cours *Introduction aux bases de données* permet à la personne étudiant de se familiariser avec l'opération et l'administration d'une base de données, préparant l'étude de fonctionnalités plus avancées telles que vue dans ce cours.

Les acquis du cours *Architecture de logiciel* sont mis à profit dans ce cours pour concevoir des applications cohérentes qui répondent aux exigences du cahier de charge.

Le cours *Interfaces humain-machine*, qui est également donné à la 4e session se concentre sur l'apprentissage des notions de design communes à toutes les interfaces. Ces notions sont très présentes dans les guides de styles jumelés aux plateformes de développement natif.

ÉNONCÉ DE LA COMPÉTENCE :

La compétence visée dans le cours est la compétence **00SS** : Effectuer le développement d'applications natives avec base de données.

Les compétences sollicitées dans le cours sont

- La compétence **00Q6** : Exploiter les principes de la programmation orientée objet.
- La compétence **00Q7** : Exploiter un système de gestion de base de données.
- La compétence **00SY** : Collaborer à la conception d'applications.

CIBLE D'APPRENTISSAGE :

Développer et déployer une application native pour un ordinateur personnel utilisant une base de données et contrôlable entièrement à l'aide d'une interface graphique.

ÉTAPES PROGRESSIVES D'APPRENTISSAGE ET ÉVALUATIONS DES APPRENTISSAGES :

1. Contenus d'apprentissage

Il est important de noter que les éléments suivants sont cités à titre indicatif. Il se pourrait qu'il y ait certains aménagements dictés par le déroulement du cours.

- **Survol rapide de la syntaxe du langage utilisé**
 - Syntaxe du langage C#
 - Les structures de contrôle en C#
 - Introduction aux classes et aux méthodes
 - La gestion des tableaux et les collections en C#
 - Introduction à la programmation orientée objet (POO) en C#
 - La gestion des fichiers en C#
- **Cadriciels (frameworks) de développement d'applications natives**
 - Introduction aux applications natives et aux cadriciels (frameworks)
 - Présentation et utilisation du cadriciel (framework) .NET / .NET
 - Cadriciels pour le développement des interfaces graphiques modernes
 - Présentation du (framework) WinUI
 - Implémentation de l'interface graphique avec WinUI.
 - Cadriciel d'intégration de la connectivité à la base de données
 - Introduction au cadriciel (framework) Entity
 - Configuration et utilisation d'Entity Framework.
- **Intégration d'une base de données**
 - Création d'une base de données à partir d'un besoin
 - Concepts importants pour travailler avec les bases de données
 - Indexation
 - Écriture de fonctions
 - Programmation de déclencheurs
 - Autres concepts
 - La connexion à une base de données à l'aide de bibliothèques
 - Introduction à l'ORM (Mapping objet-relationnel)
- **Contrôle de qualité**
 - Introduction aux tests unitaires en C#
 - Introduction au contrôle de qualité dans le développement de logiciels.
 - Les techniques de contrôle de qualité
 - La création de tests automatisée
 - La couverture du code
 - Le suivi de problèmes
 - La correction des erreurs
 - Les revues de code
- **Déploiement d'une application**
 - Création d'exécutables

- Empaquetage (packaging)
- Distribution sécuritaire (signature, checksum, etc.)
- **L'architecture et la gestion structure d'une application de bureau (Desktop)**
 - La planification d'un projet
 - L'architecture de l'application
 - La structure de l'application
 - Patron de conception

2. Activités d'apprentissage et d'enseignement

Le cours est donné en laboratoire et il s'échelonne sur quinze semaines à raison de deux rencontres par semaine.

À chaque rencontre, une majeure partie est consacrée à la présentation de la matière nécessaire pour la réalisation du projet en équipe, pour atteindre les différents objectifs.

Le reste de la période permet au professeur d'expliquer, de clarifier les concepts importants, de faire des démonstrations, et de solutionner les exercices réalisés.

De plus, l'étudiant doit consacrer au moins deux heures supplémentaires de travail personnel par semaine, pour compléter les devoirs, avancer et réaliser le projet.

3. Évaluation des apprentissages

➤ Évaluations formatives

Afin de fournir une rétroaction continue aux étudiants, chaque semaine, pendant les périodes pratiques, ou en dehors des séances de cours des exercices formatifs en lien avec la matière enseignée sont distribués et l'enseignant donne une rétroaction aux étudiants pendant ou après ces périodes.

De plus, le corrigé de ces exercices peut être expliqué en classe et distribué aux étudiants.

➤ Évaluations sommatives, critères d'évaluation et pondération

Tableau des évaluations

Type d'évaluation	Pondération
4 quiz (3% chaque)	12 %
5 devoirs (5% chaque)	25 %
Total des examens (présentiel)	37 %
Charte d'équipe	2 %
10 semainiers (1% chaque)	10 %
Démonstration à la fin d'une itération	2%
Présentation du projet final	3%
Démonstration de l'application	3%
Évaluation par les pairs	2%

Auto-évaluation	1%
Projet de session (Épreuve finale)	40 %
Total des travaux d'équipe / projet	63 %
Total des évaluations	100 %

Épreuve finale

- **Type d'évaluation**

À partir d'un cahier de charge détaillant la demande d'une entreprise fictive, la personne étudiante développe en équipe une application native ciblant des ordinateurs de bureau qui répondra aux besoins du client. La réalisation de l'épreuve s'échelonne sur plusieurs semaines.

L'application doit utiliser un langage orienté objet et doit intégrer l'utilisation d'une base de données relationnelle. L'application doit utiliser les librairies de la plateforme ciblée afin d'offrir une interface graphique d'apparence native. Finalement, elle doit être empaquetée afin d'être prête au déploiement. L'épreuve finale est divisée en trois volets.

Volet 1 : Conception d'une application native de bureau (10 %)

À partir d'un cahier de charge détaillant la demande d'une entreprise fictive, la personne étudiante avec son équipe doivent rédiger le document de conception (en équipe de quatre personnes étudiantes). Le projet est échelonné sur l'ensemble de la session et il est remis en plusieurs livrables. Le premier livrable constitue le volet 1 de l'épreuve finale et a une pondération de 10 %.

Volet 2 : Réalisation d'une application native de bureau (15 %)

À partir d'un document de conception, la personne étudiante doit réaliser, en équipe de quatre personnes étudiantes, une application native de bureau. Le projet est échelonné sur l'ensemble de la session et il est remis en plusieurs livrables (code, scripts, documents, l'exécutable, etc.). Le deuxième livrable constitue le volet 2 de l'épreuve finale et a une pondération de 15 %.

Volet 3 : Présentation et documentation d'une application native de bureau (15 %)

Suite à la conception et la réalisation de l'application native de bureau, la personne étudiante doit produire, en équipe de quatre personnes étudiantes, les différents documents du projet, un rapport final et les différents artéfacts. Le projet est échelonné sur l'ensemble de la session et il est remis en plusieurs livrables. Le livrable final constitue le volet 3 de l'épreuve finale et a une pondération de 15 %.

Veillez noter que l'échéancier suivant dépendra de l'avancement du cours. Il peut donc varier durant la session.

<u>CALENDRIER SYNTHÈSE</u>			
Semaine d'enseignement	Nature et date de remise des évaluations	Points alloués	Autres informations (s'il y a lieu)
Semaine 1	<ul style="list-style-type: none"> Présentation du plan du cours Configuration de l'environnement de développement Présentation de la charte d'équipe 		
Semaine 2	<ul style="list-style-type: none"> Introduction au langage C# Remise de la charte d'équipe Devoir 1 (résumé article : Gestion de projet) 	2% 5%	
Semaine 3	<ul style="list-style-type: none"> La gestion des fichiers La programmation orientée objet (POO) avec C# Présentation du projet de session + semainier 		
Semaine 4	<ul style="list-style-type: none"> Présentation des outils et gabarit pour commencer le projet Introduction aux applications natives et aux cadriciels (frameworks) <ul style="list-style-type: none"> Syntaxe Modèle Éléments Librairies Présentation et utilisation du cadriciels (framework) .NET pour le développement Installation des outils pour le développement avec .NET Structure d'une application .NET Quiz 1 (C#) Remise semainier 1 	3% 1%	
Semaine 5	<ul style="list-style-type: none"> Cadriciels pour l'interface graphique (framework UI) Devoir 2 (article : planification de projet agile) Remise semainier 2 	5% 1%	

CALENDRIER SYNTHÈSE

Semaine d'enseignement	Nature et date de remise des évaluations	Points alloués	Autres informations (s'il y a lieu)
Semaine 6	<ul style="list-style-type: none"> • Implémentation de l'interface graphique avec WinUI. • Patron de conception pour application de bureau • Quiz 2 (Cadriciels) • Remise semainier 3 	3% 1%	
Semaine 7	<ul style="list-style-type: none"> • Introduction au cadriciel (framework) Entity • Configuration et utilisation d'Entity Framework. • Retour sur les bases de données (SGBD) • Devoir 3 : (Synthèse d'article : gestion de risque) • Remise semainier 4 	5% 1%	
Semaine 8	<ul style="list-style-type: none"> • Semaine de relâche 		
Semaine 9	<ul style="list-style-type: none"> • Connexion aux bases de données par des librairies • Adéquation (Mapping) entre objets et bases de données (ORM) • Envoi de requêtes à travers la librairie • Devoir 4 : (Synthèse d'article : documentation d'un projet informatique) • Remise semainier 5 • Démonstration à la fin d'une itération 	5% 1% 2%	
Semaine 10	<ul style="list-style-type: none"> • La persistance des données • Travailler avec les bases de données : nouveaux concepts • Quiz 3 (DB et ORM) • Remise semainier 6 	3% 1%	
Semaine 11	<ul style="list-style-type: none"> • Le contrôle de qualité dans un projet informatique • Le suivi des problèmes de code et ses outils • Création des tests unitaires et les plans de tests • Devoir 5 (QA : Élaboration d'un plan de test) 	5%	

CALENDRIER SYNTHÈSE

Semaine d'enseignement	Nature et date de remise des évaluations	Points alloués	Autres informations (s'il y a lieu)
	<ul style="list-style-type: none"> • Remise semainier 7 	1%	
Semaine 12	<ul style="list-style-type: none"> • Retours sur les tests unitaires • Les tests unitaires avec C# • Techniques de contrôle de qualité • La couverture de test • Remise semainier 8 	1%	
Semaine 13	<ul style="list-style-type: none"> • Le plan de correction des erreurs • La revue de code • Le déploiement des applications • L'empaquetage (packaging) • Quiz 4 (tests unitaires) • Remise semainier 9 	3% 1%	
Semaine 14	<ul style="list-style-type: none"> • Présentation des projets finaux • Démonstrations des applications • Remise semainier 10 	3% 3% 1%	
Semaine 15	<ul style="list-style-type: none"> • Remise de l'évaluation par les pairs • Remise de l'auto-évaluation • Épreuve finale : Remise du projet final (le rapport final, le code, l'exécutable, les outils et les des différents documents) 	2% 1% 40%	

EXIGENCES PARTICULIÈRES DU COURS :

Locaux: laboratoires informatiques de développement d'application (S-021 et S-025).

RÈGLES INSTITUTIONNELLES ET DÉPARTEMENTALES :

Usage du cellulaire et appareils électroniques en classe

Dans les lieux d'enseignement, l'utilisation d'ordinateurs portables et d'appareils électroniques (téléphones cellulaires, téléavertisseurs, lecteurs audionumériques, agendas électroniques, caméras numériques, assistants numériques personnels etc.) est interdite. Tout contrevenant pourra être expulsé sans préavis. Ces appareils doivent être rangés hors de vue pour toute la durée des séances de cours.

Enregistrement vocal ou vidéo

Par ailleurs, les usagers de tels appareils doivent respecter l'intégrité physique et morale des personnes. En conséquence, en tout temps et en tous lieux, il est formellement interdit d'enregistrer, de photographier ou de filmer sans le consentement des individus concernés.

Les modalités d'application de la Politique institutionnelle d'évaluation de l'apprentissage (PIEA) sont rendues disponibles aux étudiants et il appartient à ceux-ci d'en prendre connaissance.

Les articles ici-bas qui font l'objet de modalités particulières d'application font référence à ceux de la PIEA en vigueur disponible sur le portail du Cégep Marin-Victorin.

Présence aux évaluations sommatives

Conformément à l'article 4.4.1 de la PIEA :

La présence à une évaluation sommative est obligatoire. L'étudiant qui s'absente, sans motif grave à l'appui, reçoit la note zéro. C'est à l'étudiant qu'il revient d'aviser son professeur des motifs de son absence dans le plus bref délai et de lui fournir, s'il y a lieu, une pièce justificative. Seul un motif grave (ex. mortalité, accident ou maladie) peut être reconnu comme valable par le professeur. Dans un tel cas, selon la nature de l'évaluation, le professeur proposera à l'étudiant une modalité de récupération.

Lors d'un examen, l'étudiant doit se présenter au moment et à l'endroit prévus. S'il arrive en retard et qu'un autre étudiant a déjà terminé et quitté la salle, l'accès lui est refusé, à moins que la nature de l'évaluation le permette.

Remise des travaux

Conformément avec l'article 4.4.2 de la PIEA :

Dans le cas d'un travail, le professeur détermine les modalités de remise, à savoir le lieu et le support (version électronique, version imprimée ou document original). Tout travail qui ne respecte pas ces modalités pourra être refusé. Le professeur détermine également la date et le moment de la remise du travail. L'étudiant qui remet son travail en retard se verra, sauf dans des situations jugées exceptionnelles par le professeur, attribuer une pénalité de 10% de la pondération prévue au départ de ce travail, par jour ouvrable, à compter du jour et de l'heure de la remise du travail.

Par ailleurs, un travail qui n'est pas remis à temps peut être refusé à compter du moment où le professeur utilisera le contenu de ce travail dans le cadre de son cours, ou qu'il sera requis pour poursuivre un travail en équipe. Une telle condition pédagogique doit être indiquée à l'avance aux étudiants, avec les consignes du travail.

Tout travail remis au professeur après que les étudiants ont reçu leurs travaux corrigés est refusé. Seul le professeur, s'il le juge à propos, peut proposer un autre travail et accorder un délai.

Dans tous les cas où le type de travail le permet, l'étudiant doit conserver un brouillon, un fichier électronique ou une photocopie de son travail.

Correction du Français

Conformément avec l'article 4.6.2 de la PIEA :

Dans les productions écrites (examens, travaux, projets), la correction du français est obligatoire et elle constitue une pénalité jusqu'à concurrence de 10% de la note. Pour établir cette pénalité, les productions écrites sont corrigées à l'aide d'une grille à échelle descriptive, selon le type de travail exigé.

Présence en classe

Conformément à l'article 4.7.1 de la PIEA, il appartient à l'étudiant :

- De fournir les efforts nécessaires pour atteindre les objectifs du cours.
- D'être présent, à l'heure, à toutes les périodes de cours prévues à son horaire et d'y participer activement. À défaut d'être présent, il doit récupérer par lui-même les apprentissages manqués.
- De respecter l'horaire prévu de même que le temps de pause. L'étudiant qui ne respecte pas ces exigences pourra subir les sanctions prévues au Règlement relatif aux conditions de vie au Cégep Marie Victorin (Règlement numéro 9).
- De respecter les délais de remise des travaux ou, si cela est impossible, il a la responsabilité d'entrer en contact avec son professeur dans les meilleurs délais (l'article 4.4.2 de la PIEA).

Il appartient aussi à l'étudiant:

- De respecter toutes les autres règles prévues à la politique relative à l'utilisation des technologies de l'information et de la communication
- D'utiliser un langage approprié, courtois et professionnel dans ses communications numériques

De plus, il est strictement interdit de boire ou manger dans les laboratoires.

Plagiat ou fraude

Selon l'article 4.9.1 de la PIEA :

Le plagiat se définit comme l'acte de faire passer pour sien un contenu ou une production d'autrui sans en identifier la source. Commet un plagiat l'étudiant qui par exemple :

- Recopie un extrait d'un texte sans utiliser les normes de citation.
- S'approprie l'idée ou le texte d'un auteur en le paraphrasant incorrectement ou en omettant d'utiliser les normes de citation.
- Utilise un concept, une image ou une musique sans en indiquer la source.

La fraude se définit comme l'acte de tromper dans le but d'en tirer un avantage personnel. Commet une fraude l'étudiant qui par exemple :

- Utilise un autre matériel que celui qui est autorisé, incluant le matériel qu'il a produit dans une évaluation pour un autre cours.
- Copie le travail ou les réponses d'examen d'une autre personne.
- Aide une autre personne à copier.
- Participe au vol, à la falsification de données, de document ou de matériel reliés à une évaluation ou à la justification d'une absence lors d'une évaluation (par exemple, n papier de médecin).
- Utilise de l'aide non permise pour réaliser un travail.

Tout plagiat, toute tentative de plagiat, toute collaboration à un plagiat entraîne la note zéro « 0 » pour l'évaluation en cause et doit faire l'objet d'un rapport écrit au Service des programmes et du développement pédagogique de la part de l'enseignante ou de l'enseignant.

Une récidive peut entraîner des mesures allant jusqu'au renvoi du Collège de l'étudiante ou de l'étudiant par la Direction des études. Pour en savoir plus sur la façon de citer ses sources dans un travail afin d'éviter le plagiat, consultez la [section suivante](#) du site internet de la [bibliothèque](#) du cégep Marie-Victorin.

MÉDIAGRAPHIE :

- Beaulieu, A. (2020). *Learning SQL : Generate, manipulate, and retrieve data* (Third edition). O'Reilly.
- Beck, K. (2003). *Test-driven development : By example*. Addison-Wesley.
- Beighley, L. (2007). *Head first SQL*. O'Reilly Media.
- *C# docs—Get started, tutorials, reference*. (s. d.). Microsoft Documentation. Consulté 15 février 2022, à l'adresse <https://docs.microsoft.com/en-us/dotnet/csharp/>
- DeBarros, A. (2018). *Practical SQL : A beginner's guide to storytelling with data*. No Starch Press.
- Freeman, E., Robson, E., Sierra, K., & Bates, B. (2021). *Head first design patterns : Building extensible & maintainable object-oriented software* (2nd edition). O'Reilly.
- *GNOME Developer Documentation*. (s. d.). GNOME Development. Consulté 15 février 2022, à l'adresse <https://developer.gnome.org/documentation/>
- *Java Documentation*. (s. d.). Oracle Help Center. Consulté 15 février 2022, à l'adresse <https://docs.oracle.com/en/java/index.html>
- McLaughlin, B., Pollice, G., & West, D. (2007). *Head first object-oriented analysis and design* (1st ed). O'Reilly.
- *.NET documentation*. (s. d.). Microsoft Documentation. Consulté 15 février 2022, à l'adresse <https://docs.microsoft.com/en-us/dotnet/>
- Obe, R. O., & Hsu, L. S. (2017). *PostgreSQL : Up and running: a practical guide to the advanced open source database* (Third edition). O'Reilly Media, Inc.
- Schonig, H.-J. (2020). *Mastering PostgreSQL 13 : Build, administer, and maintain database applications efficiently with PostgreSQL 13*.
- Weisfeld, M. A. (2013). *The object-oriented thought process* (Fourth edition). Addison-Wesley.