

PLAN DE COURS

TITRE DU COURS : Introduction à la programmation

CODE DE COURS : 420-KB1-LG

PROGRAMME : 420.B0 Techniques de l'informatique

DISCIPLINE : Informatique

Enseignement régulier

Formation continue

Pondération : 4-4-3

Session et année : Automne 2018

Préalable(s) : Aucuns

Pavillon : Pavillon d'ordinaire

| NOM DU OU DES ENSEIGNANTS | COORDONNÉES (local, téléphone, courriel) |
|----------------------------|---|
| Joan Sébastien Morales | Bureau F320, poste 2783, joan.morales@clg.qc.ca |
| Alexandre André-Lespérance | Bureau S523, poste 2771, aalesperance@hotmail.com |
| Maxime Barakatt | Bureau F308, poste 2369, utilisez Colnet |
| Stéphane Chassé | Bureau F301, poste 2518, stefchasse@videotron.ca |
| Etienne Forest | Bureau F306, poste 2787, etienne.forest@gmail.com |

| NOM DU OU DES COORDONNATEURS | COORDONNÉES (local, téléphone, courriel) |
|------------------------------|---|
| Vincent Echelard | Bureau F316, poste 2781, vincent.echelard@clg.qc.ca |
| Joan Sébastien Morales | Bureau F320, Poste 2783, joan.morales@clg.qc.ca |

Présentation du cours

Ce cours est une introduction aux méthodes généralement utilisées en programmation et en algorithmie pour développer des applications simples. Le langage de programmation utilisé est le C# et l'environnement de développement intégré est Visual Studio. On y aborde, entre autres, le cycle de développement logiciel, les données, les opérations de base, les expressions, les structures algorithmiques et le découpage en fonctions. On insiste sur les tests, la qualité du code et le bon fonctionnement de l'application.

Contribution du cours au programme

Ce cours est le premier de l'axe Programmation, la branche fondamentale et porteuse de tous les cours de la formation spécifique. L'apport des notions du cours de mathématiques seront réinvesties dans ce cours, particulièrement concernant les notions d'opérateurs arithmétiques, des opérations logiques et de la représentation des nombres. L'étudiant ou l'étudiante devra aussi développer une méthode de travail mettant l'accent sur la qualité de la conception et de la codification, le bon fonctionnement de l'application en fonction des demandes, l'usage des tests et l'auto-documentation du code qui lui servira dans les cours suivants de l'axe programmation.

Présentation de la ou des compétences du devis ministériel développées dans ce cours

Le cours permet le développement complet de la compétence 00Q2 Utiliser des langages de programmation.

Objectif

Standard

| Énoncé de la compétence | Contexte de réalisation |
|---|---|
| 00Q2 Utiliser des langages de programmation | <ul style="list-style-type: none">• Pour des problèmes dont la solution est simple.• À l'aide d'algorithmes de base.• À l'aide d'un débogueur et d'un plan de tests fonctionnels. |

| Éléments de la compétence | Critères de performance |
|--|--|
| 1. Analyser le problème. | <ul style="list-style-type: none">• Décomposition correcte du problème.• Détermination correcte des données d'entrée, des données de sortie et de la nature des traitements.• Choix et adaptation appropriés de l'algorithme. |
| 2. Traduire l'algorithme dans le langage de programmation. | <ul style="list-style-type: none">• Choix approprié des instructions et des types de données élémentaires.• Découpage efficace du code informatique.• Organisation logique des instructions.• Respect de la syntaxe du langage. |

| | |
|---|---|
| 3. Déboguer le code. | <ul style="list-style-type: none"> • Code informatique conforme à l’algorithme. • Utilisation efficace du débogueur. • Repérage complet des erreurs. • Détermination judicieuse de stratégies de correction des erreurs. • Pertinence des correctifs. • Notation claire des solutions aux problèmes rencontrés. |
| 4. Appliquer le plan de tests fonctionnels. | <ul style="list-style-type: none"> • Manifestation d’attitudes et de comportements empreints de rigueur. • Repérage complet des erreurs de fonctionnement. • Pertinence des correctifs. • Fonctionnement correct du programme. • Notation claire de l’information relative aux tests et à leurs résultats. |

Présentation de l’objectif d’intégration

À partir d’une description des fonctionnalités, l’étudiant sera en mesure de créer une application simple, complète et robuste en utilisant **Visual Studio** et les bases de la programmation **C#**.

Objectifs d’apprentissage et contenu du cours

1. Développer des solutions informatiques
 - a. Introduction au cycle de développement logiciel (analyse, élaboration, codification, tests)
 - b. Les éléments de base d’un algorithme (intrants, traitement, extrant)
 - c. Les données (variable, constante littérale et constante symbolique)
 - d. Les opérations de base (lecture, écriture, affectation)
 - e. Les expressions (arithmétiques, relationnelles, logiques)
 - f. Les structures algorithmiques: séquence, alternative (complète et incomplète), répétitive (classique, boucle à compteur, boucle conditionnelle)
 - g. L’imbrication des structures algorithmiques
 - h. Le développement graduel
 - i. Le découpage en fonctions
 - j. Les mécanismes d’échange d’informations
 - k. Les jeux d’essais (valeurs limites, extrêmes et erronées)
2. Coder des algorithmes dans le langage de programmation **C#**
 - a. Les commentaires minimalistes, les entêtes de fichiers et les standards de documentation .NET (mettre l’emphase sur la clarté du code source)
 - b. L’utilisation de bibliothèques standards et d’espaces nommés (*using*)
 - c. La structure générale d’un programme et le programme principal
 - d. L’appel de fonctions
 - e. L’utilisation de fonctions mathématiques
 - f. Les *if*, *switch*, *while*, *do..while*, *for*
 - g. Les entrées et sorties à la console

- h. Les principaux types de données (nombres entiers, nombres à virgule flottante, Texte, booléens)
 - i. La variable (déclaration, portée, initialisation)
 - j. Les énumérations
 - k. Les opérateurs (priorités, associativités) pour enseigner une syntaxe portable
 - l. Les conversions implicites et explicites (*float.Parse, int.Parse*)
 - m. La conception et la programmation de méthodes de classe (appel, définition, entête, paramètres, variables locale et valeur de retour)
 - n. Les paramètres par valeur et par référence (*ref*)
 - o. La pile d'appels – démontrer les concepts sans tomber dans la mécanique fine
 - p. La gestion de la mémoire (pile et tas) – démontrer les concepts sans tomber dans la mécanique fine
 - q. Les types « valeur » et « référence »
 - r. L'instanciation et utilisation d'objets appartenant à des classes prédéfinies (par exemple, *Random*), accès aux membres, passage en paramètre d'un objet
 - s. Une introduction à la codification de classes : membres (attributs et méthodes), encapsulation (membres privé et publics), propriétés automatiques, accesseurs et mutateurs, constructeurs paramétriques, classes immuables
 - t. Une introduction aux tableaux 1D et 2D (*Array* 1D et 2D, valeur *null*, déclaration, allocation, initialisation, utilisation, passage en paramètre, débordements)
 - u. Une introduction aux algorithmes de tri (tri par sélection)
3. Utiliser les fonctionnalités de base de l'environnement de développement intégré Visual Studio
- a. La manipulation de solutions et de projets
 - b. L'édition du code
 - c. L'exécution des programmes
 - d. Le débogage (points arrêts, pas-à-pas, espions)
4. Démontrer de l'autonomie et de la rigueur dans une démarche structurée de résolution de problèmes
- a. Utilisation de l'aide en ligne
 - b. Respect des normes de programmation
 - c. Application méthodique des plans de tests fonctionnels

Présentation des évaluations, de la production finale d'intégration (PFI) et des conditions particulières de réussite du cours

Présentation des évaluations

| Évaluation | Date | Pondération |
|--|--|----------------------------------|
| 6 tests avec une partie théorique et une partie pratique | Tout au long de la session | De 5% à 8% pour un total de 40 % |
| Exercices de laboratoires | Tout au long de la session | 30% |
| Épreuve synthèse en 3 parties: examen théorique, examen pratique et travail synthèse | (15 ^e ou 16 ^e semaine) | 30% |
| | TOTAL: | 100 % |

Présentation de la production finale d'intégration (PFI)

Pour ce cours, la production finale d'intégration contient les deux parties suivantes :

1. Un travail synthèse individuel portant sur l'intégration de toutes les notions vues durant la session. Ce travail permettra:
 - a. L'application de tout le cycle de développement
 - b. L'utilisation de tous les types d'expression et structures algorithmique
 - c. L'utilisation de structures de données simples comme les tableaux 1D et 2D
 - d. Le découpage de la solution en plusieurs fichiers sources et plusieurs fonctions
2. Une évaluation en classe. L'évaluation comportera un volet théorique et un volet pratique. Cette évaluation inclura, entre autres :
 - a. Les opérations de base
 - b. Les expressions arithmétiques, relationnelles et logiques
 - c. Les structures algorithmiques: séquence, alternative et répétitive
 - d. L'utilisation et la programmation de méthodes de classe

Critères d'évaluation de la production finale d'intégration

- Analyse correcte du problème
- Respect des consignes et implémentation des fonctionnalités demandées
- Choix des types et des structures de données appropriées
- Modularisation correct du code en fichiers et en fonctions
- Clarté du code et respect des normes de programmation
- Tests et robustesse de l'application

Présentation des conditions particulières de réussite du cours

En plus d'avoir cumulé 60 % et plus du total de toutes les évaluations (incluant la production finale d'intégration), la réussite de ce cours est sujette à une condition particulière: réussir la partie de la production finale d'intégration qui comporte l'évaluation supervisée en classe avec un minimum de 60 %. Ainsi, l'échec à cette partie de la production finale d'intégration équivaldra à une note maximale de 59 % pour ce cours.

Évaluation des travaux d'équipe d'envergure

Il n'y a aucuns travaux d'équipe dans ce cours.

Échéancier ou découpage du cours en séquence

| Semaine | Contenu |
|----------|--|
| 1-2 | Plan de cours Introduction Chapitre 1 - Les éléments algorithmiques TEST 1 (5%) |
| 3-4 | Chapitre 2 - Expressions relationnelles et alternatives TEST 2 (6%) |
| 5-6 | Chapitre 3- Expressions logiques et répétitives TEST 3 (7%) |
| 7-8-9 | Chapitre 4 - Les sous-programmes TEST 4 (8%) |
| 10 | Chapitre 5 - L'utilisation d'une bibliothèque graphique TEST 5 (6%) |
| 11-12-13 | Chapitre 6 – Tableaux et classes TEST 6 (8%) |
| 14-15 | Épreuve synthèse (30%) |

NB.: La structure hebdomadaire telle que montrée précédemment peut être modifiée en cours de session selon le rythme du groupe et imprévus.

Démarche pédagogique

Ce cours utilise une approche pédagogique qui se nomme la classe inversée. La classe inversée consiste à apprendre individuellement la théorie avant d'arriver en classe (souvent à l'aide d'un enregistrement vidéo de votre professeur) puis de faire des exercices sur cette théorie en classe.

La classe inversée comporte beaucoup d'avantages:

- vous pouvez visionner la capsule au moment où vous le voulez! Donc un moment où vous êtes peut-être plus en forme!
- vous pouvez mettre votre professeur "sur pause"!
- vous pouvez faire répéter votre professeur autant de fois que vous le voulez!
- ensuite, le professeur a plus de temps en classe pour aider les étudiants individuellement,
- et finalement, vous avez beaucoup plus de temps en classe pour vous pratiquer!

Par contre, le succès de cette approche repose sur votre participation! Vous n'aurez pas plus de travail qu'à l'habitude puisque le nombre d'heures de travail personnel à la maison (3 heures par semaine pour le cours de KB1) sera respecté. Vous devez seulement faire le travail avant de vous présenter en classe plutôt qu'après!

Les étapes d'une classe inversée sont :

1. Visionnement de la capsule vidéo par l'étudiant avant d'arriver en classe.
2. Compléter un questionnaire ou un exercice d'autocorrection avant d'arriver en classe.
3. Au début du cours: Validation par l'enseignant des notions apprises à l'aide de question et/ou sondages.
4. Majeure partie du cours : exercices pratiques en classe

➤ N'oubliez pas d'apporter vos écouteurs si vous voulez écouter des capsules en laboratoire!

Politique départementale d'évaluation des apprentissages

Les éléments suivants présentent les règles et procédures relatives à l'évaluation des apprentissages du département d'informatique (420).

Présence et la participation aux cours

Conformément à la PIEA, la présence au cours est obligatoire. Le fait d'être absent ne peut pas, en lui seul, justifier le fait d'avoir manqué la ou les activités d'apprentissage qui ont eu lieu durant son absence. Il est de la responsabilité de l'étudiante ou de l'étudiant de prendre les mesures nécessaires pour rattraper son retard de façon autonome. L'enseignante ou l'enseignant n'est nullement tenu d'encadrer la reprise des activités manquées par l'étudiant.

La participation à des activités d'apprentissage en dehors du cadre de la classe

La participation à des activités d'apprentissage se tenant en dehors du cadre de la classe, mais liées aux objectifs d'un cours est obligatoire. Cela s'applique également au stage de fin d'études.

La présence aux évaluations sommatives et les modalités de reprise

La présence à une activité d'évaluation sommative est obligatoire. L'étudiante ou l'étudiant qui s'absente, sans motif exceptionnel et justifié, comme défini dans les règles et procédures élaborées par les assemblées départementales et inscrites au plan de cours, reçoit la note zéro (0).

C'est à l'étudiant qu'il revient de rencontrer son enseignante ou son enseignant pour lui faire part, dès son retour au Collège, des motifs de son absence et lui fournir une pièce justificative. Dans le cas d'un motif exceptionnel et justifié, selon la nature de l'activité d'évaluation sommative, l'enseignant pourrait proposer à l'étudiant une modalité de reprise ou une modification de barème.

Les motifs exceptionnels et justifiés sont les suivants :

- le décès d'un conjoint ou d'un membre de la famille proche, c'est-à-dire : conjointe ou conjoint, enfant, mère, père, belle-mère, beau-père, sœur, frère, belle-sœur, beau-frère, bru, gendre, grand-mère, grand-père ;
- la maladie, l'hospitalisation de l'étudiant ;
- l'assignation en cour de l'étudiant.

Cette liste n'est pas exhaustive et permet aux enseignants d'exercer leur jugement en fonction des situations.

La remise des travaux – Les pénalités pour retards et les modalités de reprise

Tout travail doit être remis à l'enseignante ou à l'enseignant à la date et selon les modalités indiquées au plan de cours. L'enseignant peut fixer des pénalités pour des retards. L'acceptation de travaux en retard après la remise des travaux corrigés ne devrait pas être permise, à moins que l'enseignant ne le juge à propos en raison d'un motif exceptionnel. Dans de tels cas, l'étudiante ou l'étudiant doit fournir une pièce justificative.

Lorsque des activités particulières reconnues par le Collège l'exigent (telles que celles du volet Sport-études ou des stages à l'étranger), les délais de remise des travaux pourront être adaptés à la situation.

Notez toutefois que l'enseignant est libre des modalités de pénalité pour un retard lors de la livraison d'un travail, en fonction du cours et du contexte.

L'évaluation de la qualité de la langue française

La langue française constitue l'un des principaux instruments du développement et de l'organisation de la pensée. La qualité du français revêt une telle importance que celle-ci constitue l'un des grands objectifs de tout apprentissage. La présente politique définit un cadre favorisant l'amélioration ou la consolidation de la qualité du français par l'étudiante ou par l'étudiant.

Un travail peut voir sa note varier jusqu'à 10 % en plus ou en moins en raison de l'excellence ou la déficience du français lorsqu'applicable.

Autres renseignements ou autres exigences particulières au cours

- Durant un examen, le professeur n'est en aucun cas disponible pour répondre aux questions sauf si ce sont des questions d'ordre physique (bris, problème réseau, etc.) ou de compréhension face à une question mal posée à l'examen. Il ne sert donc à rien d'appeler le professeur si votre question sert à obtenir une réponse ou même une piste à une réponse, le professeur ne vous la donnera pas.
- La durée d'un examen est celle prévu par le professeur et celle qui sera marquée également sur l'examen. Aucun délai de temps supplémentaire ne sera accordé à moins d'avis contraire.
- Les cellulaires doivent être placés en mode vibration.
- La nourriture et les jeux vidéo sont interdits dans les laboratoires.
- Par respect pour l'enseignant et les étudiants, on ne peut pas entrer dans un cours s'il est déjà commencé est que la porte est fermée.

Le plagiat et la tricherie

Extrait de la *Politique institutionnelle d'évaluation des apprentissages (PIEA)*

6.7 Le plagiat et la tricherie

Par la présente politique, le Collège déclare que le plagiat et la tricherie, sous toutes leurs formes, sont contraires aux valeurs qui orientent sa mission. Elles constituent des infractions graves qui, lorsque constatées, entraînent des pénalités.

6.7.1 Les définitions

« [L'] utilisation frauduleuse de l'œuvre d'autrui soit par emprunt, soit par imitation¹ » constitue un plagiat. L'étudiante ou l'étudiant qui permet que son travail soit copié en tout ou en partie est réputé aussi coupable que le plagiaire.

Le plagiat comprend :

- a) la présentation ou la remise du travail d'une autre personne comme étant le sien ;
- b) le copiage en tout ou en partie ou la traduction de l'œuvre d'autrui (publiée ou non publiée) sans en indiquer la source ;
- c) la paraphrase de l'œuvre d'autrui (publiée ou non publiée) sans en indiquer la source, à moins que cela ne constitue l'objet d'une activité d'évaluation formative ou sommative.

Toute autre forme de pratique malhonnête ou de dissimulation lors d'une évaluation sommative est considérée comme une tricherie.

La tricherie comprend, notamment :

- a) la falsification d'un rapport de laboratoire ;
- b) la préparation d'un travail pour quelqu'un d'autre ou la réalisation de son travail par quelqu'un d'autre ;
- c) le fait de prétendre avoir remis un travail qui en fait n'a jamais été remis ;
- d) la soumission de fausses informations, par exemple un faux certificat médical, afin d'obtenir un délai dans la remise d'un travail, dans le but d'éviter des sanctions ou d'obtenir un privilège ;
- e) le fait de demander à quelqu'un d'autre de signifier sa présence en classe alors que l'on est absent ;
- f) le fait d'attester la présence en classe d'une autre étudiante ou d'un autre étudiant absent ;
- g) l'obtention ou la tentative d'obtention, au cours de la réalisation d'une activité d'évaluation, d'une aide de quelque nature que ce soit sans y être autorisé ;
- h) l'aide ou l'assistance à une autre étudiante ou à un autre étudiant, au cours de la réalisation d'une activité d'évaluation, sans y être autorisé ;

- i) l'emploi ou la possession de tout matériel, de tout instrument ou de tout moyen pouvant servir au stockage, à la diffusion ou à la recherche d'informations sans y être autorisé ;
- j) le fait de réaliser toute forme d'évaluation à la place de quelqu'un d'autre ;
- k) le fait de confier la réalisation de ses propres évaluations à quelqu'un d'autre ;
- l) la communication, lors d'une activité d'évaluation en classe, avec quelqu'un d'autre, de quelque façon que ce soit, sans y être autorisé.

Le plagiat et la tricherie peuvent être décelés par l'enseignante ou par l'enseignant avant, pendant ou après la tenue de l'évaluation.

Médiagraphie

- Site Moodle du cours
- www.sfml-dev.org/index-fr.php
- Profdinfo.com
- C# 7.0 in a Nutshell: The Definitive Reference, Joseph Albahari et Ben Albahari, O'Reilly Media; 1 édition (28 octobre 2017), 1088 pages, 978-1491987650

Annexe I – Logigramme des préalables

