Cours 420-B43-IN

Programmation par objets avancée

Pondération : 3-3-3 ( 3 crédits – 4ème session )

Programme : *Techniques de l'informatique 420.A0*

Voie de spécialisation : *Informatique de gestion*

Préalables : Concepts de l'orienté-objet 420-B33-VM

Compétence visée : Appliquer une approche de développement par objets ( 16 T )

Produire une interface utilisateur ( 16 X )

Session : Automne 2012

*Professeur et coordonnées*

Nom : Éric Labonté

Courriel : elabonte@cvm.qc.ca

Local : A5.37

Tél. : ( 514 ) 982-3437 poste 7982

Site web : www.cvm.qc.ca/elabonte

Ce plan de cours a été adopté par le département d'informatique lors de sa réunion de juin 2010

**Présentation générale du cours**

###### Renseignements généraux

Le programme de technique de l’informatique vise à former des technicien(ne)s aptes à développer et implanter des applications, exploiter du matériel informatique, exploiter des bases de données, assurer du soutien technique et gérer des réseaux.

###### Brève description du cours

La programmation orientée-objet est une tendance de fond dans le monde informatique; la vitesse avec laquelle Java s’est imposé comme langage de programmation en est la preuve. Ce cours poursuivra l’apprentissage des concepts orientés objet par les étudiants et visera à les initier à la conception ainsi qu’à la programmation d’interfaces utilisateur à l’aide du langage Java. À titre de cours offert à des étudiants de 2ème année, il permet de donner un atout de plus dans l’approche généraliste de ce programme de techniques informatiques.

###### But du cours et lien avec le programme de formation

Ce cours vise à poursuivre l'apprentissage des concepts orientés objet et à initier les étudiants à la conception ainsi qu'à la programmation d'interfaces utilisateur à l'aide du langage Java.  
  
Ce cours est le deuxième d'une série de deux visant l'atteinte de la compétence 16T. À la fin de ce cours, les étudiants devront donc appliquer une approche de développement par objets afin de résoudre une situation problématique donnée. Des projets intégrants différentes technologies Java (Applets, dessin personnalisé, Accès aux bases de données Oracle, Multithreading, utilisation de librairies externes) permettront de relier l'ensemble des notions vues dans les deux cours.  
  
De plus, ce cours amorce l'atteinte de la compétence 16X devant permettre la production par les étudiants d'interfaces utilisateur. En particulier, les étudiants utiliseront différentes classes de composantes Java (JButton, JTextField, etc..) et pourront ainsi gérer les événements captés par les objets en question

###### Objectifs intégrateurs de ce cours

Au terme de ce cours, l’étudiant devra être capable de programmer un modèle objet et de greffer à ce modèle une ou des interfaces utilisateur permettant des interactions entre le modèle et les utilisateurs de l’application en respectant les règles de l’architecture MVC ( Model – View – Controller )

**Objectifs spécifiques :**

• S’approprier le langage de programmation Java.  
• Codifier des classes Java chacune servant de modèle à une série d’objets.  
• Valider le fonctionnement des classes grâce à des classes de tests.  
• Connaître les différentes fonctionnalités de l’environnement de programmation utilisé.  
• Générer une version exécutable des applications réalisées.  
• Programmer des interfaces utilisateur reliées aux modèles objet.  
• Appliquer différentes technologies Java (multithreading, protocole JDBC pour l’accès aux bases de données, utilisation de librairies externes ) afin de raffiner les programmes produits.

**Compétences ministérielles**

Appliquer une approche de développement par objet ( 016T )

* Créer un modèle objet
* Perfectionner le modèle objet
* Procéder à la codification d’une classe
* Valider le fonctionnement d’une classe
* Générer la version exécutable du programme

Produire une interface utilisateur ( 016X )

* Analyser les caractéristiques des utilisateurs et utilisatrices
* Choisir des périphériques d’entrée et de sortie
* Planification de l’organisation globale de l’interface
* Procéder à la programmation de l’interface utilisateur

###### Organisation des activités d'enseignement et d'apprentissage

Le cours se déroulera entièrement en laboratoire.  
  
La majorité des cours consisteront en une présentation des éléments de programmation orientée objet à l’étude agrémentée de nombreux exercices à réaliser à l’ordinateur visant ainsi une participation active des étudiants.  
  
À d’autres moments, certains cours seront consacrés à la réalisation des travaux pratiques sommatifs ou du projet de l’épreuve certificative de ce cours. À l’aide des informations fournies par le professeur, l’étudiant pourra appliquer les concepts orientés objet afin de réaliser des projets dans un contexte le plus réaliste possible. Il passera à travers les différentes étapes de production d’un projet, de l’analyse du problème à la vérification de ses classes à l’aide de classes de tests lorsqu’une classe ou lorsque le projet au complet est achevé. De plus, il pourra concevoir des interfaces utilisateur reliées à son projet selon l’architecture vue en classe. Tous ces travaux permettront la maîtrise de l’environnement de travail utilisé.

Ces travaux pratiques sommatifs seront notés selon les critères suivants :

* Respect des concepts orientés objet
* Fonctionnalité de l’application
* Bonne ergonomie de l’interface utilisateur
* Disposition du code, présence de commentaires

**Ce qui est attendu de vous au niveau du comportement :**

* **Une présence active à chacun des cours :**
  + la présence aux cours est fortement encouragée; en fait elle s’est avérée directement proportionnelle à la note obtenue par le passé.
* Aucun travail autre que celui du cours ne sera toléré durant les périodes de cours
* Il est interdit de manger ou de boire dans les locaux informatiques
* Arriver quelques minutes avant le début du cours afin d’être en état d’apprentissage lors du début du cours

**Ce qui est attendu de vous au niveau de vos compétences :**

* Être capable de s’auto-évaluer et d’aller chercher de l’aide s’il y a problème
* Faire les travaux demandés et les remettre dans les délais prévus
* Manifester une attitude positive et agréable envers ses collègues et l’enseignant

###### Périodes de disponibilité

|  |  |
| --- | --- |
| **jour** | **heures** |
| lundi | 14h00 – 16h00 |
| mardi | 14h00 – 15h00 |
| mercredi | 15h00 – 16h00 |
| jeudi |  |
| vendredi | 11h00 – 12h00 |

###### Les activités d'évaluation

La note finale de l'étudiant sera calculée selon les normes énoncées ci-dessous :

|  |  |  |  |
| --- | --- | --- | --- |
| Évaluation | Nombre | Pondération Dates importantes | |
| Examens | 2 | 60% | Examen de mi-session : vendredi 18 janvier  Examen final : jeudi 7 février |
| Travaux | 2 | 15% |  |
| Contrôles ( mini-tests ) | 1 | 10% |  |
| Projet | 1 | 15% | Remise : vendredi 8 février |

**Remarques :**

Examens :

* Toute documentation sera permise lors des examens et des contrôles
* La présence aux contrôles et aux examens est obligatoire. Voir Annexe à la fin du document.

Travaux / projet :

* Les travaux et le projet seront remis de manière individuelle et doivent être réalisés de manière individuelle également ( travaux dont certaines parties ou le tout sont identiques se verront octroyer la note 0 )
* Les travaux devront être remis dans le module de remise de LÉA. Les modalités de remise des travaux seront expliquées en classe.

Double seuil :

Afin de réussir ce cours, l’étudiant devra obtenir :

• Une moyenne d’au moins 60% dans l’ensemble des évaluations

• Une moyenne d’au moins 60% dans l'activité de synthèse de ce cours ( voir page suivante ). Si l'étudiant n'obtient pas 60% dans l'activité de synthèse, il n'aura pas réussi le cours même si sa moyenne dans l'ensemble des évaluations est supérieure à 60%. La note de 55% sera attribuée à un étudiant dans cette situation improbable mais mathématiquement possible.

###### Activités de synthèse ( Épreuve certificative )

Deux mesures permettront d’évaluer l’atteinte des éléments visés par les compétences et seront donc considérés comme activité synthèse de ce cours :

1. Un examen vérifiera la matière vue durant la session. Il combinera une partie écrite et une partie à réaliser à l’ordinateur consistant à programmer une application contenant un modèle objet et une interface utilisateur.
2. Un projet où l’étudiant devra :

* Programmer plusieurs modèles objet visant la complétion d’un projet respectant les principes fondamentaux de la programmation orientée objet (encapsulation, héritage, polymorphisme). Ce projet sera donc constitué d’instances de ces modèles communiquant entre eux à l’aide de méthodes.
* Perfectionner ces modèles afin de permettre une meilleure réutilisation des ressources.
* Codifier ces classes.
* Vérifier leur bon fonctionnement à l’aide de classes de tests créées par l’étudiant et par le professeur.
* Mettre en relation ces classes avec une interface utilisateur programmée par l’étudiant permettant de générer une version exécutable du projet stimulante.
* Utiliser des technologies Java afin de maximiser les fonctionnalités de l’application ( accès à une base de données Oracle ).

Contexte de réalisation :

Le projet sera fait individuellement par les étudiants en fin de session

Critères généraux d’évaluation :

L’évaluation du projet se fera en deux phases afin d’aider au maximum les étudiants à réussir l’ensemble de l’activité et avoir ainsi en mains un projet fonctionnel et stimulant à la fin de la session. L’évaluation du projet s’appuiera sur les points suivants :

* L’organisation du code respectant les principes orientés objet
* La fonctionnalité du programme en simulant son utilisation par un usager

**Calendrier des activités**

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Sujets abordés | Évaluations formatives  ( consulter le site web du cours ) | Évaluations  sommatives |
| 1 | * Introduction aux applets : classes Applet, JApplet, Graphics * Gestion des événements AWT pour les applets * Publication d'un applet sur un site web * Balise HTML APPLET * anticrénelage * dessin personnalisé : méthode paint, update, paintComponent * notion de zone de dessin * notions de base relatives aux Threads : le Timer * différences entre composantes Swing et AWT concernant l’application de la classe Graphics | * exercices en classe * exercices contenus dans les annexes 1C et 2 | * Contrôle 1 ( 10% ) * Travail pratique 1 ( 7% ) * Travail pratique 2   (8%)   * Examen de mi-session ( 25% ) |
| 2 | * Interfaces graphiques utilisant SWING * Classes JButton, JTextField, JPanel, JLabel, JComboBox, JCheckBox, JRadioButton, JTable, JOptionPane, JMenu, JTabbedPane, etc. * Layouts * Mécanismes de gestion des événements pour les interfaces graphiques utilisant SWING * Notion d’écouteurs, d’adaptateurs * Architecture MVC ( Model-View-Controller) * *Les objets JDialog* * *classes Runtime, Robot* * librairies POI | * exercices en classe * exercices contenus dans les annexes 3 à 10 |
| 3 | * Traitement des exceptions : classes Error, Exception * Hiérarchie des classes d’exceptions * Exceptions contrôlées et non-contrôlées * Classes d’exceptions créées par le programmeur * Mots-clés try, catch, finally | * Exercices en classe * Exercices contenus dans les annexes 11 à 15 | * Projet   ( 15% )   * Examen final ( 35% ) |
| 4 | * Accès aux fichiers texte ( package java.io ) * Classes Reader, Writer, InputStream, OutputStream * Accès aux bases de données Oracle   ( protocole JDBC )   * Package java.sql * *Accès à des fichiers XML* | * Exercices en classe * Exercices dans les annexes 16 à 20 |

**Environnement du cours**

**Environnement:**

* Système d’exploitation Windows 7
* JDeveloper 11g ( à télécharger sur K:/Logiciels Libres )

**Ordinateurs disponibles :**

Chaque élève doit disposer d'un micro-ordinateur pour toutes les périodes du cours.

**Matériel requis pour le cours**

**Livre obligatoire :**

Aucun matériel spécifique n'est requis pour ce cours

**Notes de cours / annexes :**

À télécharger sur le site web : [www.cvm.qc.ca/elabonte](http://www.cvm.qc.ca/elabonte)

**Médiagraphie**

**Livres / sites web**

livres populaires : Comment programmer en Java 2 ( 4ème édition ), Deitel & Deitel, éditions Reynald Goulet. ( 2002 )

Starting out with JAVA: From Control Structures through Objects (5eme édition ), Tony Gaddis, Editions Pearson, ISBN : 978-0-13-285583-9 (2010 )

sites WEB : en français :

- <http://bruce-eckel.developpez.com/livres/java/traduction/tij2/>

- [www.eteks.com](http://www.eteks.com) ( du C++ à Java )

en anglais :

- [www.java.sun.com](http://www.java.sun.com) ( référence en Java )

- [www.javaboutique.internet.com](http://www.javaboutique.internet.com)

- [www.unicode.org](http://www.unicode.org)

- [www.javaranch.com](http://www.javaranch.com)

autres livres :

* Frank Carrano, Imagine ! Java Programming Concepts in Context, 2010, 1008 pages, Pearson ISBN : 0131471066
* Frank Nielsen, A Concise and Practical Introduction to Programming Algorithms in Java, 2009, 280 pages, Springer, ISBN 184882338X

- John R. Hubbard, Programming With Java, 2ème édition, 2004, 202 pages, éd. McGraw Hill, ISBN 9780071420402

**De nombreux sites web complètent cette médiagraphie. Les adresses de ces sites se retrouvent sur le site web :** [www.cvm.qc.ca/elabonte](http://www.cvm.qc.ca/elabonte)

**Politique départementale**

**Département des techniques de l'informatique**

*Résumé des règles d’encadrement départementales relatives*

*à l’évaluation des apprentissages* (**adopté le 16 mars 2010**)

Ces règles précisent certaines modalités relatives à la *Politique d’évaluation des apprentissages* du cégep du Vieux Montréal. Nous invitons les élèves à la consulter :

* [*www.cvm.qc.ca/cegep/reglesPolitiques/Documents/PolitiqueEvaluationApprentissages.pdf*](http://www.cvm.qc.ca/cegep/reglesPolitiques/Documents/PolitiqueEvaluationApprentissages.pdf)

**MODALITÉS DE REPRISE.**

Si, pour des raisons exceptionnelles, un élève n’a pu se présenter à la date prévue pour un test ou un examen, le professeur peut l’autoriser à reprendre ce test ou cet examen si la raison de l’absence est consignée et agréée par écrit par le professeur; l’élève subit alors un examen dans les délais qui lui sont impartis **et selon les conditions fixées par le professeur**. Dans le cas où l’absence n’est pas agréée par le professeur, l’élève obtient la note zéro (0).

L’absence lors d’une épreuve synthèse de programme entraîne la note zéro (0), à moins que cette absence ne soit justifiée auprès de la coordination du département qui décidera avec les professeurs et les professionnels concernés des mesures à prendre. La réussite de l’épreuve synthèse est une condition d’obtention du diplôme d’études collégiales. Les consignes relatives à l’absence à l’épreuve synthèse doivent être communiquées à chaque session aux élèves concernés.

**ÉVALUATION.**

1. **Évaluation du français.**

La pondération liée à la qualité du français de tout travail écrit en français est de 10%. Les travaux sont corrigés à l’aide de la grille simplifiée. Chaque faute comptera pour 0.2 point.

**b) Plagiat et fraude.**

Le plagiat ou la fraude au cours d’un examen ou d’un travail entraîne automatiquement la note zéro (0) pour cet examen ou ce travail.

Un second cas de plagiat ou fraude entraîne l’échec du cours.

**c) Absence aux cours.**

L’absence au cours peut entraîner un échec lorsque la présence au cours est reconnue comme essentielle à l’atteinte d’objectifs précis prévus au plan cadre de ce cours et identifiés au plan de cours. Le cas échéant, toute absence non motivée par une raison de santé sera sanctionnée suivant les modalités prévues au plan de cours.

**d) Présence au cours.**

En général, et à moins d’indication contraire au plan de cours, il n’y a pas de pénalité pour une absence au cours. Cependant, le département d’informatique constate une forte corrélation entre la présence au cours et la réussite du cours. Dans cette optique, il encourage *fortement* la présence et la participation au cours. L’étudiant absent au cours se verra attribuer la note zéro pour sa participation lorsque celle-ci est obligatoire et prévue au plan de cours.

1. **Participation à certaines activités.**

L’élève absent au cours se verra attribuer zéro (0) pour sa participation lorsque celle-ci est obligatoire et prévue au plan de cours. Cependant, un maximum de 10% de la note finale peut être obtenu ainsi, suite à la participation à certaines activités pédagogiques liées aux objectifs du cours. Les modalités de la participation sont inscrites au plan de cours.

1. **Correction des travaux, des examens et des diverses épreuves.**

Le temps requis pour la correction des travaux de session et des projets peut dépasser le délai habituel de deux semaines, à condition d’être signalé au plan de cours. Le délai ne pourra être supérieur à quatre semaines.

1. **Note de passage.**

Les erreurs de mesure peuvent faire varier la note accordée pour une évaluation. En conséquence, un intervalle de confiance est précisé pour la note de passage. Afin de démarquer de façon significative l’échec de la réussite**,** **aucun professeur ne mettra une note finale entre 55 et 60**. Dans le cas où le résultat de l’élève est proche de la note de passage, cet intervalle permettra au professeur de disposer d’une marge de manœuvre afin de porter un jugement sur l’atteinte par l’élève des objectifs du cours avant de lui accorder ou non la note de passage.

Un cours peut comporter un double seuil de passage. Pour réussir ce cours, l’élève doit alors obtenir **60% à l’épreuve certificative du cours et 60% pour le total** de ses évaluations. Les modalités doivent être présentées dans le plan de cours.

**VOIES DE RECOURS DES ÉLÈVES EN MATIÈRE D’ÉVALUATION.**

* **L’article 7 de la *PEA* s’applique.**

1. **Modification de note pendant le cours**

Il s'agit d'une demande auprès du professeur, aussitôt après la remise du travail ou de l’examen corrigé, afin que la note obtenue soit modifiée. Après étude, le professeur maintient ou modifie la note.

1. **Modification de note après la réception du relevé de notes**

Première étape: la demande de **correction de note.**

Suite à la réception de son relevé de notes, l'élève peut compléter une demande de modification de note, dans laquelle il expose les motifs sérieux de sa demande. Celle-ci est acheminée à l’enseignant qui en prend alors connaissance et décide de maintenir ou de modifier la note finale.

Deuxième étape (*étape facultative*): La demande de **révision de note**.

Après avoir pris connaissance du résultat de sa demande de correction de note, l’élève qui se croit encore lésé par la note finale peut demander une révision de note selon le mécanisme prévu en précisant les motifs de sa demande. Un comité de révision de note est alors constitué. Ce comité de trois professeurs devra alors vérifier que la correction d'un travail ou d'un examen soit faite en appliquant rigoureusement le barème de correction. Les motifs d'une demande de révision de note portent donc sur :

* une mauvaise application du barème de correction
* une mauvaise interprétation de la réponse.