

Programmation II

Coordonnateur du cours

LAFORÉST, Louise
 laforest.louise@uqam.ca
 (514) 987-3000 #7790
 PK-4725

Groupes

20	MALENFANT, Bruno	malenfant.bruno@uqam.ca	(514) 987-3000 #3699	PK-4115
	Mardi, de 18h00 à 21h00 Salle (cours)		Jeudi, de 18h00 à 20h00 Salle PK-S1555 (atelier)	
30	MALENFANT, Bruno	malenfant.bruno@uqam.ca	(514) 987-3000 #3699	PK-4115
	Mercredi, de 14h00 à 17h00 Salle (cours)		Lundi, de 13h30 à 15h30 Salle PK-S1565 (atelier)	

Description du cours

Approfondir les concepts de la programmation orientée-objet. Approfondir les concepts de mise au point et de test de composants logiciels. Identification et définition des classes d'une solution logicielle. Relations entre les classes: composition et héritage. Classes abstraites et polymorphisme. Introduction à la notation UML. Algorithmes récursifs simples. Structures de données classiques: piles, files, listes et arbres binaires de recherche. Techniques classiques de recherche (séquentielle et binaire) et de tri. Introduction à la programmation des interfaces graphiques (GUI). Gestion des événements et des exceptions. Conception de paquetages. Introduction aux outils automatisés de validation. Ce cours comporte une séance obligatoire de laboratoire (2 heures). Six de ces laboratoires seront évalués.

Préalables académiques :
 INF1120 Programmation I

Objectifs du cours

- Approfondir les méthodes de conception et de programmation orientée-objet;
 - Spécialisation des classes;
 - Connaître les structures de données fondamentales et savoir les choisir et les utiliser;
 - Connaître les techniques de base de la recherche de données et des tris;
 - Connaître la conception d'une interface graphique (GUI) avec les composants de base;
 - Acquérir des connaissances de base en génie logiciel;
 - S'initier aux étapes de la réalisation d'un produit logiciel;
- Stratégies de conception, de mise au point et de tests;
- Documentation du logiciel;
- Approfondir le langage Java;
 - À la fin de la session, l'étudiant(e) devrait être en mesure d'élaborer un programme structuré et fonctionnel en utilisant les notions de génie logiciel étudiées au cours et en se servant des différentes structures de données fondamentales.

Contenu du cours

1. Retour sur le concept de classe

- Variable d'instance, de classe et valeur constante
- Accès aux composants d'une classe
- Notion d'encapsulation
- Méthode d'instance et de classe
- Constructeur

2. Notion d'héritage

- Contexte d'utilisation
- Classe Object
- Constructeur de la sous-classe et Super
- Héritage et méthodes : redéfinition, surcharge
- Contrôler l'héritage : Final
- Notion de polymorphisme
- Conversion de type entre les classes d'une hiérarchie

3. Junit

- Utilisation de Junit pour les tests

4. Introduction aux collections

- Notion de collection en Java
 - Notion de type abstrait (TDA)
 - Notion de structure de donnée
- a) Introduction à ArrayList
 - b) Notion d'interface
 - Notion de contrat
 - Utilisation vs implémentation
 - c) Introduction au TDA Pile
 - Interface des services sur Pile
 - Exemples d'utilisation
 - Implémentation avec ArrayList
 - d) Introduction au TDA File
 - Interface des services sur File
 - Exemples d'utilisation
 - Implémentation avec ArrayList
 - e) Notion de paquetage

5. Introduction à la récursivité

- Fonctionnement : cas de base et convergence
- Trace d'exécution
- Conception d'une méthode récursive
- Conversion d'une méthode récursive en une méthode itérative

6. Introduction aux listes chaînées

- Avantages
- Notion de Noeud ou Maillon de liste
- Notion de chaînage
- Nouvelles implémentations des TDA Pile et File avec liste chaînée
- Insertion et retrait dans une liste chaînée
- Notion de remorque de fin de liste
- Parcours d'une liste

- Liste ordonnée
- Notion de liste doublement chaînée

7. Introduction aux interfaces graphiques (Swing)

- Notion de container et de composants
- JFrame, JLabel, JTextField, JButton
- Gestion d'événements (Listener)
 - ActionEvent (souris et clavier)
 - ItemEvent
 - Gestionnaire d'événements et leur enregistrement
- Les GUI et les threads
- Les boîtes de dialogue (classe JOptionPane)
- JPanel, JTextArea, JRadioButton, JCheckBox
- Notion de Layout manager (BorderLayout, FlowLayout, GridLayout)
- Exemple d'un projet d'interface intégrant les notions de base

8. Techniques de recherche

- Recherche séquentielle non ordonnée et ordonnée
- Recherche binaire
- Notation grand-O
- Arbre binaire de recherche : Construction, insertion, recherche, parcours préfixe, postfixe et infixe.

9. Techniques de tri

- Tris simples ($O(n^2)$)
 - Sélection
 - Échange (tri à bulles)
 - Insertion
- Tri rapide : QuickSort

Modalités d'évaluation

Description sommaire	Date	Pondération
Examen commun intra	Samedi, 29 octobre 9:30-12:30	30%
Examen commun final	Samedi, 17 décembre 9:30-12:30	25%
3 travaux pratiques : TP1 (11%) - TP2 (13%) - TP3 (13%)	Les dates de remise sont spécifiques à chacun des groupes	37%
Quiz en classe	4 quiz	8%

Règles concernant le seuil de passage

L'étudiant doit obtenir une moyenne cumulée aux examens égale ou supérieure à 50%, ainsi qu'une moyenne cumulée pour les travaux supérieure ou égale à 50%. Si ces seuils ne sont pas atteints, la mention échec sera automatiquement attribuée au cours.

Travaux pratiques

- À moins d'avis contraire, les travaux pratiques sont strictement individuels.
- En cas de doute sur l'originalité des travaux, un test oral pourra être exigé. Tous les cas de plagiat seront référés au comité de discipline de la Faculté. La sanction peut aller de la note 0 pour le travail ou pour l'examen jusqu'à l'exclusion de l'université.
- La politique concernant les retards pour les travaux pratiques est à la discrétion de chacun des enseignants.
- Il est de la responsabilité de l'étudiant de faire des copies de sauvegarde de ses travaux. Il est possible que le professeur demande à l'étudiant de lui transmettre certaines parties du travail suite à la remise.

Les étudiants doivent consulter régulièrement le site Web des cours de programmation.

Nous rappelons aux étudiants qu'ils doivent s'attendre à fournir une moyenne de 6 heures de travail personnel par semaine pour un cours de trois crédits (total de 90 heures).

Politique d'absence aux examens

L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen – par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : PK-3150 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs. Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

Intégrité académique

PLAGIAT Règlement no 18 sur les infractions de nature académique. (extraits)

Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constitue une infraction au sens de ce règlement.

La liste non limitative des infractions est définie comme suit :

- la substitution de personnes;
- l'utilisation totale ou partielle du texte d'autrui en la faisant passer pour sien ou sans indication de référence;
- la transmission d'un travail pour fins d'évaluation alors qu'il constitue essentiellement un travail qui a déjà été transmis pour fins d'évaluation académique à l'Université ou dans une autre institution d'enseignement, sauf avec l'accord préalable de l'enseignante, l'enseignant;
- l'obtention par vol, manoeuvre ou corruption de questions ou de réponses d'examen ou de tout autre document ou matériel non autorisés, ou encore d'une évaluation non méritée;
- la possession ou l'utilisation, avant ou pendant un examen, de tout document non autorisé;
- l'utilisation pendant un examen de la copie d'examen d'une autre personne;
- l'obtention de toute aide non autorisée, qu'elle soit collective ou individuelle;
- la falsification d'un document, notamment d'un document transmis par l'Université ou d'un document de l'Université transmis ou non à une tierce personne, quelles que soient les circonstances;
- la falsification de données de recherche dans un travail, notamment une thèse, un mémoire, un mémoire-crédation, un rapport de stage ou un rapport de recherche;
- Les sanctions reliées à ces infractions sont précisées à l'article 3 du Règlement no 18.

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html> et <http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

Médiagraphie

VR LEWIS & CHASE -- *Java Software Structures* -- **3e édition, Addison-Wesley, 2010.**

VR DELANNOY, C. -- *Programmer en Java - Java 5 et 6* -- **6e édition, Eyrolles, 2009.**

LR -- *Environnement de développement BlueJ* (gratuit) <http://www.bluej.org>

LR -- *Environnement de développement Eclipse* (gratuit) <http://www.eclipse.org>

LR -- *Environnement de développement NetBeans* (gratuit) https://netbeans.org/index_fr.html

A : article - C : comptes rendus - L : logiciel
S: Standard - U : uri - V : volume

C : complémentaire - O : Obligatoire - R : recommandé