COORDONNATEUR		LAFOREST, Louise	laforest.louise@uqam.ca	(514) 987-3000 7790	PK-4725
GROUPES	10	LAFOREST, Louise	laforest.louise@uqam.ca	(514) 987-3000 7790	PK-4725
		Lundi 13h30 à 15h00 et de 15h15 à 16h45 (cours) – Mercredi, de 14h00 à 16h00 (ateliers)			
	30	DOUKOURE, Ismaël	doukoure.ismael@uqam.ca	(514) 987-3000 3699	PK-4115
		Mercredi, de 18h00 à 21h00 (cours) – Jeudi, de 16h00 à 18h00 et de 18h00 à 20h00 (ateliers)			
	40	LEFEBVRE, Bernard	lefebvre.bernard@uqam.ca	(514) 987-3000 3426	SH-5725
		Jeudi, de 9h00 à 10h30 et de 10h45 à 12h15 (cours) – Mercredi, de 9h00 à 11h00 (ateliers)			

DESCRIPTION

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 INF1120 Programmation I

OBJECTIFS

- 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.

ÉVALUATION

Description sommaire	Date	Pondération
Examen commun intra	Samedi 10 mars de 9h30 à 12h30	30%
Examen commun final	Samedi 28 avril de 9h30 à 12h30	25%
2 travaux pratiques : TP1 (15%) – TP2 (20%)	Les dates de remise sont spécifiques à chacun des groupes	35%
Quiz en classe	5 quiz (les 4 meilleurs seront retenus)	10%

Les dates de distribution des énoncés des travaux pratiques ainsi que les dates de remise sont spécifiques à chacun des groupes.

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

❏ Les règlements de l'UQAM concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants :

http://www.sciences.uqam.ca/etudiants/integrite-academique.html

http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html

- ❏ 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 plagiats 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.
- ❏ Une pénalité de 10 % par jour ouvrable sera appliquée aux travaux remis après les dates prévues. Après 5 jours ouvrables de retard, le travail sera considéré comme non remis entraînant la note 0 pour ce travail.
- ❏ Il est de la responsabilité de l'étudiant de faire des copies de sauvegarde de ses travaux sur au moins 2 disquettes. 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 : SH-4700 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**

CONTENU

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 (ADT)
- Notion de structure de donnée

a) Introduction à ArrayList

b) Notion d'interface
   - ✓ Notion de contrat
   - ✓ Utilisation vs implémentation

c) Introduction à Pile (ADT)
   - ✓ Interface des services sur Pile
   - ✓ Exemples d'utilisation
   - ✓ Implémentation avec ArrayList

d) Introduction à File (ADT)
   - ✓ Interface des services sur Pile
   - ✓ Exemples d'utilisation
   - ✓ Implémentation avec ArrayList

e) Notion de package

5. Introduction aux listes chaînées
   - Avantages
   - Notion de *Noeud* ou maillon de liste
   - Notion de chaînage
   - Nouvelles implémentations de ADT 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
6. Introduction à la récursitivé
   - Fonctionnement : cas de base et convergence
   - Trace d'exécution
   - Conception d'une méthode récursive
   - Conversion d'une méthode récursive en itératif
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 boite 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-0

- Arbre binaire de recherche
  - ✓ Insertion
  - ✓ Parcours; prefixe, infixe, postfixe

9. Techniques de tri
   - Tris simples ($O(n^2)$)
     - ✓ Sélection
     - ✓ Échanges (tri à bulles et tri roches)
     - ✓ Insertion
   - Tri rapide : QuickSort

RÉFÉRENCES

U R **http://www.grosmax.uqam.ca/prog/**
Site web du cours

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

V R DELANNOY, C. – *Programmer en Java* – **7e édition (Java6), Eyrolles, 2011.**

L R *Environnement de développement BlueJ (gratuit)* – **http://www.bluej.org**

L R *Environnement de développement Eclipse (gratuit)* – **http://www.eclipse.org**

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue – S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé