

COORDONNATEUR	LAFORÉST, Louise	laforest.louise@uqam.ca	(514) 987-3000 7790	PK-4725
GROUPES	01 LÉVESQUE, Stéphane	stephane.levesque@videotron.ca	(514) 987-3000 0439	PK-4151
	Mardi, de 18h30 à 20h30 (cours) – Jeudi, de 18-30 à 20h40 (ateliers) – CEU Laval			
	20 BARASSI, Marc	barassi.marc@uqam.ca	(514) 987-3000 0439	PK-4151
	Mardi, de 18h00 à 21h00 (cours) – Jeudi, de 18h00 à 20h00 (ateliers)			
	21 LAFORÉST, Louise	laforest.louise@uqam.ca	(514) 987-3000 7790	PK-4725
Mardi et jeudi, de 10h30 à 12h00 (cours) – Mardi, de 13h00 à 15h00 (ateliers)				
40 LAFORÉST, Louise	laforest.louise@uqam.ca	(514) 987-3000 7790	PK-4725	
Jeudi, de 15h30 à 17h00 et vendredi, de 13h30 à 15h00 (cours) - Vendredi, de 15h15 à 17h15 (ateliers)				
42 LÉVESQUE, Stéphane	stephane.levesque@videotron.ca	(514) 987-3000 0439	PK-4151	
Jeudi, de 18h00 à 21h00 (cours) – Mardi, de 18h00 à 20h00 (ateliers)				

DESCRIPTION
<p>Acquérir une méthode de développement de solutions logicielles dans le cadre du paradigme orienté-objet: analyse du problème, conception simplifiée, codage et test d'une solution. Sensibiliser au développement de programmes de qualité: fiables, faciles à utiliser, à comprendre et à modifier.</p> <p>Introduction à un environnement de développement logiciel. Introduction aux algorithmes. Éléments de programmation de base: vocabulaire, syntaxe et sémantique, constantes, variables, types simples et composés (tableaux à une et deux dimensions), conversions de type, affectation, opérateurs et expressions, instructions, structures de contrôle (séquence, sélection, itération), instructions simples d'entrées-sorties, fichier texte. Introduction aux éléments de la programmation orientée-objet: classes, objets, méthodes et paramètres, variables de classe, d'instance et locale, portée et durée de vie des variables, constructeurs. Notion d'encapsulation. Introduction à l'utilisation de classes et de paquetages prédéfinis.</p> <p>Ce cours comporte une séance obligatoire de laboratoire (2 heures). Six de ces laboratoires seront évalués.</p>

OBJECTIF
Acquérir une méthode de développement de solutions logicielles dans le cadre du paradigme orienté-objet : analyse du problème, conception simplifiée, codage et test d'une solution. Sensibiliser au développement de programmes de qualité : fiables, faciles à utiliser, à comprendre et à modifier.

ÉVALUATION	Description sommaire	Date	Pondération
	Examen commun intra	Samedi 5 novembre 2005 de 14h00 à 17h00	25%
	Examen commun final	Samedi 17 décembre 2005 de 14h00 à 17h00	25%
	TP 1	*	10%
	TP 2	*	15%
	TP 3	*	15%
	Laboratoires	Énoncés fournis au laboratoire	10%

\* Les dates de remise sont spécifiques à chacun des enseignants.

L'étudiant doit obtenir une moyenne cumulée aux examens égale ou supérieure à 50% ainsi qu'une moyenne cumulée pour les travaux pratiques égale ou supérieure à 50%. Si ces seuils ne sont pas atteints, la mention échec sera automatiquement attribuée au cours. Les règlements de l'UQAM concernant le plagiat seront strictement appliqués. À moins d'avis contraire, les travaux pratiques sont individuels et en cas de doute sur l'originalité des travaux, un test oral pourra être exigé. En cas de plagiat, la sanction peut aller de la note zéro pour le travail ou l'examen jusqu'à l'exclusion de l'université. Les travaux doivent être remis aux dates spécifiées par chacun des enseignants. 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 disquettes. La perte d'une disquette ou de son contenu (à cause d'un virus ou de toute autre raison) n'est pas une raison valable pour remettre un travail en retard ou ne pas le remettre.

Les étudiants doivent consulter régulièrement le site Web des cours de programmation ainsi que le site Web de leur enseignant. On y trouve, entre autres, les énoncés des travaux pratiques, certains exemples et les questions de révision pour les examens.

**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.**

**Politique d'absence aux examens**

Un étudiant absent à un examen se verra normalement attribuer la note zéro pour cet examen. Cependant, si l'étudiant était dans l'impossibilité de se présenter à l'examen pour un motif valable, certains arrangements pourront être pris avec son enseignant. Pour ce faire, l'étudiant devra présenter à son enseignant l'un des formulaires prévus à cet effet accompagné des pièces justificatives appropriées (par ex., attestation d'un médecin que l'étudiant était dans l'impossibilité de se présenter à l'examen pour des raisons de santé, lettre de la Cour en cas de participation à un jury).

Une absence pour cause de conflit d'horaires d'examen n'est pas considérée comme un motif valable d'absence, à moins d'entente préalable avec la direction du programme et l'enseignant durant la période d'annulation des inscriptions avec remboursement : tel qu'indiqué dans le guide d'inscription des étudiants, il est de la responsabilité d'un étudiant de ne s'inscrire qu'à des cours qui ne sont pas en conflit d'horaire.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique et pour obtenir les formulaires appropriés, consultez le site web suivant :

<http://www.info.uqam.ca/enseignement/politiques/absence-examen>

## CONTENU

Voici un aperçu des thèmes que seront traités dans ce cours.

- Introduction aux algorithmes
  - Lecture, affichage, sélection, boucles
  - Pseudo-code
- De l'algorithme au programme
  - Édition, compilation, exécution
  - Environnement Java
  - Méthode simple de développement d'un logiciel
- Introduction à la programmation Java
  - Structure d'un programme simple en Java
  - Conventions d'écriture du code Java
  - Notion de variables, de types simples et de constantes
  - Affectation de valeurs
- Opérateurs arithmétiques, relationnels et logiques
  - Priorité des opérateurs
  - Type du résultat d'une expression
  - Transformation de type
- Entrées/sorties simples
  - Affichage de valeurs de types simples
  - Saisie de valeurs
  - Utilisation d'une classe maison
- Sélections
  - Instruction if-else
  - Sélections imbriquées
  - Choix multiples
- Répétitions
  - Instructions while, do-while, for
  - Comment choisir une boucle
- Introduction aux méthodes
  - Utilisation de méthodes prédéfinies
  - Conception de méthodes
  - Visibilité
- Introduction aux classes et objets
  - Concepts: classe, objet, instance

- Classe prédéfinie String
- Application d'une méthode à un objet
- Conception d'une classe simple avec ses méthodes et ses variables
- Constructeur
- ❑ Tableaux
  - Déclaration et manipulation
  - Tableaux à plus d'une dimension
- ❑ Traitement des exceptions
- ❑ Fichiers
  - Fichiers de texte: lecture et écriture de données

## RÉFÉRENCES

- VO DELANNOY, C – *Programmer en Java* – 3e édition, Eyrolles, 2004.
- UO <http://www.grosmax.uqam.ca/prog/>  
Site web du cours
- VC TASSO, A. – *Le livre de Java premier langage* – 2e édition, Eyrolles, 2002.
- VC BARNES, D.J., KÖLLING M – *Conception objet en Java avec BlueJ* – Pearson Education, 2003.

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

C : complémentaire – O : obligatoire – R : recommandé