COORDONNATEUR		VILLEMAIRE, Roger	villemaire.roger@uqam.ca	(514) 987-3000 6744	PK-4615
GROUPES	20	MEMMI, Daniel	memmi.daniel@uqam.ca	(514) 987-3000 7939	PK-4430
		Mardi, de 18h00 à 21h00 (cours) – Lundi, de 18h00 à 20h00 (ateliers)			
	30	VILLEMAIRE, Roger	villemaire.roger@uqam.ca	(514) 987-3000 6744	PK-4615
		Mercredi, de 13h30 à 15h00 et jeudi, de 8h30 à 10h00 (cours) - Jeudi, de 10h30 à 12h30 (ateliers)			

DESCRIPTION

Présenter les concepts fondamentaux de langages de programmation modernes. Comprendre les possibilités et limites des divers types de langages. Familiariser l'étudiant avec différents paradigmes de programmation et favoriser l'acquisition de nouvelles techniques et stratégies de programmation.

Étude des paradigmes de programmation fonctionnel et logique. Revue des principes de programmation fonctionnelle. Stratégies d'évaluation des arguments. Polymorphisme et déduction des types. Fonctions d'ordre supérieur. Efficacité et optimisation. Revue des principes de programmation logique. Forme clausale de la logique du premier ordre et clauses de Horn. Unification et résolution. Le problème de la négation. Applications.
Ce cours comporte une séance obligatoire de laboratoire (2 heures).

Préalables: INF1130 Mathématiques pour informaticien ou MAT2055 Logique et ensembles ; INF2120 Programmation II

OBJECTIFS

- **Que vise ce cours?** Ce cours vise à initier à la programmation fonctionnelle et logique. Il cherche à illustrer et à mettre en évidence les techniques de programmation propres à ces paradigmes. Il illustre leur potentiel et leur intérêt pour la mise en œuvre d'application complexes notamment dans le domaine de l'intelligence artificielle.
- Les compétences développées dans le cadre de ce cours vous rendront capable:
  - ❏ de connaître chacun des paradigmes,
  - ❏ d'apprécier les possibilités et les limites de chacun d'eux,
  - ❏ de développer des applications dans chacun des paradigmes.

ÉVALUATION

Description sommaire	Date	Pondération
Examen commun de programmation logique	Dimanche 28 octobre 2007 de 14h00 à 17h00	30%
Travail pratique en Prolog		20%
Examen commun de programmation fonctionnelle	Dimanche 16 décembre 2007 de 14h00 à 17h00	30%
Travail pratique en Haskell		20%

La moyenne des 2 examens doit être supérieure ou égale à 50% pour réussir le cours. L'utilisation de documents est permise aux examens. Les cartes d'étudiant seront contrôlées et exigées au début de chaque examen.

Les travaux pratiques doivent être réalisés par groupes de deux étudiants au plus. Une pénalité de 5% sera appliquée à la note pour chaque jour de retard. Les critères de correction seront distribués avec l'énoncé de chaque travail. La qualité du français constitue un critère d'évaluation pour un maximum de 10% de la note. Les cas de plagiat seront référés au comité de discipline.

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

Ce cours initie à des approches de programmation originales et différentes de la programmation impérative ou objet. Il met en évidence les caractéristiques, le potentiel de chacun de ces paradigmes au moyen d'une

introduction à des langages spécifiques et d'applications typiques de ces langages. Haskell permet de donner un exemple de langage fonctionnel puissant et moderne où les aspects liés aux types de données (vérification, polymorphisme, encapsulation) sont particulièrement développés. Prolog sert à illustrer le concept de programmation logique et les apports liés aux processus d'unification et de résolution

Caractérisation générale des paradigmes étudiés dans le cadre du cours: programmation fonctionnelle et programmation logique.

A) La programmation logique: Prolog

- introduction: historique et concepts de base,
- les faits et le principe d'unification,
- les règles et la résolution,
- les listes et la récursivité,
- exemples d'applications.

B) La programmation fonctionnelle: **Haskell**

- introduction: historique et concepts de base,
- aspects syntaxiques et sémantiques des langages applicatifs,
- récursivité et types récursifs,
- vérification de types, inférence de types, polymorphisme,
- fonctions d'ordre supérieur et curryage,
- types abstraits, encapsulation, modules et foncteurs,
- les entrées-sorties et les monades,
- le mécanisme de l'évaluation paresseuse.

RÉFÉRENCES

V R THOMPSON, S. – *Haskell, The Craft of Functional Programming* – **Addison-Wesley. 487 p., 1999**

V R CONVINGTON, M.A., NUTE, D. & VELLINO, A. – *Prolog Programmin in Depth* – **Prentice-Hall, 516p., 1997.**

V C BIRD, R. – *Introduction to Functional Programming using Haskell* – **Prentice Hall Press, 460 p. (2nd edition).**

V C RABHI, F. LAPALME, G. Lapalme – *Algorithms: A functional Programming Approach* – **Addison-Wesley, 235 p.,1999.**

V C FIELD, A.J., HARRISON, P.G. – *Functional Programming, International Computer Science* – **Workingham G.B.: Addison-Wesley, 602 p., 1988.**

V C BIRD, R. WADLER, P. – *Introduction to Functional Programming, Computer Science* – **New York: Prentice Hall, 293 p., 1988.**

V C BELLOT, Patrick – *Objectif Prolog* – **Masson, 151 p., 1994.**

V C CLOCKSIN, W.F. – *Clause and Effect: Prolog Programming for the Working Programmer* – **Springer, 143 p., 1997.**

V C BRATKO, I. – *PROLOG Programming for Artificial Intelligence* – **International Computer Science, Workingham G.B.: Addison-Wesley, 597 p., 1990 (2nd edition).**

V C STERLING, L. SHAPIRO, E. – *The Art of Prolog* – **Cambridge MA: The MIT Press, 560 p., 1994 (2nd edition).**

V C CLOCKSIN, W.F., MELLISH, C.S. Mellish – *Programming in Prolog* – **Springer-Verlag, 281p., 1994 (4th edition).**

L C *Hugs 98*

L C *Sicstus Prolog*

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

C : complémentaire – O : obligatoire – R : recommandé