

GROUPE	20 MEMMI, Daniel	memmi.daniel@uqam.ca	(514) 987-3000 7939	PK-4430
	Mardi, de 14h00 à 17h00			

**DESCRIPTION**

Ce cours s'adresse aux étudiants admis sous conditions qui doivent mettre à niveau leurs connaissances en informatique ; il fait partie de la suite DIC8001, DIC8002 et DIC8003. Le cours a pour objectif d'explorer les principaux paradigmes de programmation et leurs domaines d'application. Seront privilégiés la programmation fonctionnelle (e.g. Lisp), la programmation logique (e.g. Prolog), et leur application à la réalisation de prototypes. Des séances de laboratoire sont comprises pour toutes les parties du cours. Le cours et les travaux à réaliser seront adaptés en fonction de la composition du groupe et de l'orientation des projets de recherche des étudiants, en privilégiant une approche d'enseignement par problèmes.

Note: cours d'appoint, hors programme pour les étudiants du doctorat en informatique cognitive.

**OBJECTIF**

Ce cours sera une initiation à la programmation ou une révision des concepts de base, selon le niveau et les besoins des étudiants. Le contenu est basé principalement sur les langages Lisp et Prolog, exemples typiques de programmation fonctionnelle et logique, et langages favoris de l'Intelligence Artificielle. Mais l'accent sera mis davantage sur l'acquisition de bonnes pratiques de programmation plutôt que sur une étude exhaustive de ces langages particuliers. On essaiera en même temps d'illustrer les possibilités de chaque paradigme.

ÉVALUATION	Description sommaire	Date	Pondération
	Deux projets à programmer (un en Lisp et un en Prolog)		50%
	Examen final (notes et documents autorisés)		50%

**CONTENU**

A) 1re partie : Lisp

- **Notions de base** : algorithmes et structures de données, les variables et leur portée, notion de type, structures de données usuelles, itération et récursion
- **Programmation fonctionnelle** : bref historique, notion de fonction, variables et résultat, syntaxe et sémantique de Lisp, atomes et listes, fonctions et arguments, évaluation, primitives de listes
- **Fonctions en Lisp** : conditionnelles, appels récursifs, affectation et effets de bord, exemples simples de traitement de listes, conseils pour la programmation récursive
- **Traitement de listes** : définition récursive des listes, représentation interne, traitements récursifs, condition d'arrêt, récursion simple, double récursion
- **Fonctions d'ordre supérieur** : fonctions passées en argument, forme LAMBDA, fonctions MAP, utilité pour les traitements itératifs et exemples
- **Entrées-sorties et boucles** : lectures et impressions, primitives courantes, boucles PROG et variantes modernes, comparaison entre itération et récursion
- **Révision et projets** : revue des principales notions, discussion sur les projets

B) 2e partie : Prolog

- **Introduction à la programmation logique** : bref historique, syntaxe et sémantique, faits, règles et buts, démonstration logique et programmation
- **Fonctionnement de Prolog** : unification, processus de démonstration, retour arrière (backtrack), coupure et négation, formes d'itération et récursion
- **Traitement de listes** : structures, listes, traitements récursifs, exemples détaillés
- **Opérations et contrôle** : entrées-sorties usuelles, opérateurs, primitives de contrôle
- **Application aux systèmes experts** : rappel sur les systèmes experts, chaînage avant et chaînage arrière, réalisation d'un moteur d'inférence en chaînage avant
- **Application au langage naturel** : le problème de l'analyse de phrases, formalisme des Declarative Clause Grammars (DCG), exemple détaillé d'utilisation
- **Révision et projets** : revue des principales notions, discussion sur les projets
- **Examen final en salle** : l'examen portera sur tout le contenu du cours

**RÉFÉRENCES**

VR D.S. Touretzky – *Common Lisp : A Gentle Introduction to Symbolic Programming* – Addison-Wesley (1990).

VR P. Graham – *On Lisp: Advanced Techniques for Common Lisp* – Prentice-Hall (1994)

VR W.F. Clocksin & C.S. Mellish – *Programming in Prolog* – Springer-Verlag (1994)

---

<sup>VR</sup> M.A. Convington, D. Nute & A. Vellino – *Prolog Programming in Depth* – Prentice-Hall (1997)

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

C : complémentaire – O : obligatoire – R : recommandé