

GROUPE	30 BOUCHARD, Lorne H.	bouchard.lorne_h@uqam.ca	(514) 987-3000 3085	PK-4440
Mercredi, de 9h00 à 12h00 (cours: local PK-4370) – Vendredi, de 9h00 à 12h00 (atelier: local PK-4360)				

**DESCRIPTION**

Ce cours s'adresse aux étudiants de sciences humaines et a pour objectif de leur permettre d'acquérir dans un laps de temps limité les concepts fondamentaux et les méthodes relatifs au travail d'analyse, de conception de systèmes et de réalisation selon différents paradigmes de programmation. Les cours et les travaux à réaliser seront adaptés chaque année en fonction de la composition du groupe et de l'orientation des projets de recherche des étudiants en privilégiant une approche par problèmes. Au terme de ce cours, les compétences suivantes auront été acquises, analyse, modélisation et conception de systèmes en objets (3 crédits), implantation au moyen de structures de données statiques et dynamiques et leurs algorithmes, vérification et validation de système (3 crédits), connaissance des principaux paradigmes de programmation et de leurs principaux domaines d'application (3 crédits). Une évaluation progressive viendra confirmer l'acquisition de chacune des compétences visées. Note: pour ces étudiants, une note minimale de B- est exigée pour l'admission définitive.

- 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 oeuvre d'applications 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
	Quatre travaux individuels	Petits travaux individuels portant sur des thèmes vus en classe. Les dates de remise sont indiquées dans le calendrier. Un travail remis en retard reçoit la note zéro à moins d'avoir fait l'objet d'une entente préalable avec le professeur.	40%
	Deux rapports avec présentation	Ce travail consiste à étudier et à présenter une réalisation informatique faite l'une avec le langage Prolog, l'autre avec le langage Lisp. La présentation devra comporter une partie démonstration.	40%
	Participation active en classe	Les discussions concernant l'avancement des travaux d'équipe sont un moyen privilégié d'échanger de l'information et de compléter la matière couverte par le cours.	20%

L'évaluation est progressive. Une note minimale de B- est exigée pour l'admission définitive au programme de doctorat.

**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, au calcul symbolique notamment. Lisp est un langage polyvalent qui est à la base un exemple de langage fonctionnel simple et puissant. 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.

1. La programmation fonctionnelle: **Lisp**
  - introduction: historique et concepts de base,
  - aspects syntaxiques et sémantiques des langages applicatifs,
  - structures de listes et récursivité,
  - heuristiques pour la définition des fonctions récursives,
  - fonctions d'ordre supérieur,
  - boucles,
  - les entrées-sorties,
  - le mécanisme de l'évaluation.

2. 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.

CALENDRIER	Période	Contenu	Lecture et laboratoire
	1	Notions de base Valeurs et variables Procédures et fonctions Langage des organigrammes Listes et vecteurs Tableaux indicés et associatifs Procédures récursives	Notes du professeur
	2	<b>Paradigme de la programmation fonctionnelle</b> Introduction: historique et concepts de base Fonctions et listes	<b>David S. Touretzky, <u>Common Lisp : A Gentle Introduction to Symbolic Computing</u></b> Chapitres 1 et 2
	3	Formes et évaluation Conditionnelles Variables et effets de bord Énoncé du premier travail	Chapitres 3, 4 et 5
	4	Listes comme structure de données- Récursivité Heuristique pour la définition de fonctions récursives	Chapitres 6 et 8
	5	Programmation applicative Opérations sur les listes Fonctions d'ordre supérieur Remise du premier travail Énoncé du second travail	Chapitres 7 et 8
	6	Entrée-sortie- Affectation	Chapitres 9 et 10
	7	Boucles Remise du second travail	Chapitre 11
	8	Présentations et démonstrations	
	9	<b>Paradigme de la programmation logique</b> Introduction: historique et concepts de base Les faits et le principe d'unification Les règles et la résolution	<b>Michael A. Covington, Donald Nute, and Andre Vellino, <u>Prolog Programming in Depth</u></b> Chapitre 1
	10	Énoncé du troisième travail Les listes et la récursivité La résolution et le contrôle de la résolution	Chapitres 3 et 4
	11	Arithmétique et opérateurs Les termes et leur traitement	Chapitre 3
	12	Remise du troisième travail Énoncé du quatrième travail Entrées-sorties et meta-logique	Chapitres 2, 5 et 6

Période	Contenu	Lecture et laboratoire
13	Exemples d'application au traitement du langage naturel	Chapitre 12
14	Exemples d'application à la recherche de solutions Remise du quatrième travail	Chapitre 8
15	Présentations et démonstrations	

## RÉFÉRENCES

- VR D. S. Touretzky (1990) – *Common LISP : A Gentle Introduction to Symbolic Programming* – Addison-Wesley/Benjamin Cummings, (1990). 487 p. – <http://www.cs.cmu.edu/~dst/LispBook/book.pdf>  
Téléchargeable gratuitement
- VR M.A. Convington, D. Nute, & A. Vellino – *Prolog Programming in Depth* – Prentice-Hall, (1997) 516 p.
- VC D. P. Friedman & M. Felleisen – *The Little LISPer, (Trade Edition)* – The MIT Press, (1987) 186 p.
- VC P. Graham – *ANSI Common Lisp* – Prentice-Hall, (1996) 432 p.
- VC P. Graham – *On Lisp : Advanced Techniques for Common Lisp* – Prentice-Hall, (1994) 413 p. – <http://lib1.store.vip.sc5.yahoo.com/lib/paulgraham/onlisp.pdf>  
Téléchargeable gratuitement.
- VC P. Norvig – *Paradigms of Artificial Intelligence Programming : Case Studies in Common Lisp* – Morgan Kaufmann/Elsevier, (1992) xxviii + 946 p.
- VC F.C.N. Pereira & S. M. Shieber – *Prolog and Natural Language Analysis* – Stanford CA (1987) Center for the Study of Language and Information, 266 p.
- VC I. Bratko – *PROLOG Programming for Artificial Intelligence (2e édition) (1990)* – Addison-Wesley, 597 p.
- VC L. Sterling, E. Shapiro – *The Art of Prolog, (2e édition) (1986)* – The MIT Press, 509 p.
- VC W.F. Clocksin, C.S. Mellish – *Programming in Prolog, (4e édition) (1994)* – Springer-Verlag, 281p.
- LR <http://www.lispworks.com/downloads/>  
LispWorks Personal Edition
- LR <http://www.swi-prolog.org/download.html>  
SWI Prolog

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

C : complémentaire – O : obligatoire – R : recommandé