

GROUPE	30 LEFEBVRE, Bernard	lefebvre.bernard@uqam.ca	(514) 987-3000 1728	PK-4115
	Mercredi, de 14h00 à 17h00			

DESCRIPTION	<p>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. Haskell), la programmation logique (e.g. Prolog), et leur application à la réalisation de prototypes. 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.</p> <p>Note: cours d'appoint, hors programme pour les étudiants du doctorat en informatique cognitive.</p>
-------------	---

OBJECTIF	<p>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 Haskell 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.</p>
----------	--

ÉVALUATION	Description sommaire	Date	Pondération
	Projet en Haskell : Étude d'une réalisation Haskell dans un domaine de recherche : mise en oeuvre du logiciel, expérimentation et présentation écrite et orale des résultats de l'étude.		50%
	Projet en Prolog : Étude d'une réalisation Prolog dans un domaine de recherche : mise en oeuvre du logiciel, expérimentation et présentation écrite et orale des résultats de l'étude.		50%

CONTENU	<p>A) 1re partie : Haskell</p> <ul style="list-style-type: none"> • Programmation fonctionnelle : bref historique, notion de fonction, variables et résultat, syntaxe et sémantique d'Haskell, fonctions et arguments, évaluation. • Fonctions en Haskell : conditionnelles, appels récursifs, exemples simples de traitement, le curryage. • Traitement de listes : définition récursive des listes, traitements récursifs de listes, étude de fonctions prédéfinies. • Les types et les modules : classes de types, définition de nouveaux types, les modules. • Fonctions d'ordre supérieur : fonctions passées en argument, forme LAMBDA, fonctions map, fold, ... , utilité pour les traitements itératifs et exemples. • Sujets spéciaux : évaluation retardée, le mécanisme des listes en compréhension. • Entrées-sorties et monades : lectures et impressions par la monade IO, primitives courantes. • Révision et projets : revue des principales notions, discussion sur les projets. <p>B) 2e partie : Prolog</p> <ul style="list-style-type: none"> • 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.
---------	---

RÉFÉRENCES	<p>VR D.S. Touretzky – <i>Common Lisp : A Gentle Introduction to Symbolic Programming</i> – Addison-Wesley (1990)</p> <p>VR P. Graham – <i>On Lisp: Advanced Techniques for Common Lisp</i> – Prentice-Hall (1994)</p>
------------	--

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

^{VR} 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