

|   |                      |                          |                     |         |
|---|----------------------|--------------------------|---------------------|---------|
| COORDONNATEUR   | VILLEMAIRE, Roger    | villemaire.roger@uqam.ca | (514) 987-3000 6744 | PK-4615 |
| GROUPE  | 30 LEFEBVRE, Bernard | lefebvre.bernard@uqam.ca | (514) 987-3000 1728 | PK-4115 |
| Mercredi, de 13h30 à 16h30 (cours) – Jeudi, de 13h30 à 15h30 (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 intra                |      | 30%         |
|            | Travail pratique en Haskell |      | 20%         |
|            | Examen final                |      | 30%         |
|            | Travail pratique en Prolog  |      | 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 10% 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 règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, consultez le site suivant :

<http://www.sciences.uqam.ca/etudiants/integrite-academique.html>

#### Politique d'absence aux examens

**L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.**

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen – par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : SH-4700 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

## 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 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.

B) 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.

## RÉFÉRENCES

- VR THOMPSON, S. – *Haskell, The Craft of Functional Programming* – Addison-Wesley. 487 p., 1999
- VR CONVINGTON, M.A., NUTE, D. & VELLINO, A. – *Prolog Programming in Depth* – Prentice-Hall, 516p., 1997.
- VC BIRD, R. – *Introduction to Functional Programming using Haskell* – Prentice Hall Press, 460 p. (2nd edition).
- VC RABHI, F. LAPALME, G. Lapalme – *Algorithms: A functional Programming Approach* – Addison-Wesley, 235 p., 1999.
- VC FIELD, A.J., HARRISON, P.G. – *Functional Programming, International Computer Science* – Workingham G.B.: Addison-Wesley, 602 p., 1988.
- VC BIRD, R. WADLER, P. – *Introduction to Functional Programming, Computer Science* – New York: Prentice Hall, 293 p., 1988.
- VC BELLOT, Patrick – *Objectif Prolog* – Masson, 151 p., 1994.
- VC CLOCKSIN, W.F. – *Clause and Effect: Prolog Programming for the Working Programmer* – Springer, 143 p., 1997.
- VC BRATKO, I. – *PROLOG Programming for Artificial Intelligence* – International Computer Science, Workingham G.B.: Addison-Wesley, 597 p., 1990 (2nd edition).
- VC STERLING, L. SHAPIRO, E. – *The Art of Prolog* – Cambridge MA: The MIT Press, 560 p., 1994 (2nd edition).
- VC CLOCKSIN, W.F., MELLISH, C.S. Mellish – *Programming in Prolog* – Springer-Verlag, 281p., 1994 (4th edition).
- LC *The Haskell Platform*
- LC *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