

**Sigle : INF4083 Gr. 20**  
**Titre : Langages de programmation**  
**Session : Automne 2026 Horaire et local**  
**Professeur : Taleb, Mohamed**

## 1. Description du cours paraissant à l'annuaire :

### Objectifs

Au terme de ce cours l'étudiant(e) aura approfondi sa connaissance des concepts fondamentaux des langages de programmation, et aura acquis une vue d'ensemble des différents paradigmes de programmation : procédurale, orientée-objet, fonctionnelle, logique, concurrente. Il/elle aura acquis les principales techniques de la programmation fonctionnelle et les fondements de la programmation logique.

### Contenu

Fondements théoriques : fonctions calculables, thèse de Church-Turing, Lambda-calcul. Paradigmes de programmation : procédural, fonctionnel, orienté objet, parallèle et logique. Syntaxe et sémantique des langages, introduction à la compilation. Portée des variables, gestion de la mémoire, types, relation de sous-types, typage statique et dynamique. Techniques de programmation fonctionnelle : récursivité et récursivité terminale, fonctions d'ordre supérieur. Principes de la programmation Logique : faits et règles, clauses de Horn. Unification et résolution. Négation et coupures. Ce cours comporte des séances obligatoires de travaux dirigés (TD).

Descriptif – Annuaire

## 2. Objectifs spécifiques du cours :

Ce cours vise à donner aux étudiants une compréhension approfondie des langages de programmation, autant du point de vue théorique que pratique. Ceci leur permettra de faire un meilleur usage des langages qu'ils connaissent et de pouvoir facilement transférer leurs compétences à de nouveaux langages.

D'un point de vue théorique, ce cours a pour but de comprendre les différences sémantiques entre différents langages de programmation, et de développer des programmes dans différents langages de programmation.

D'un point de vue pratique, à travers l'apprentissage du langage multi-paradigme, les étudiants se familiariseront avec les concepts de base qui définissent la syntaxe et la sémantique des langages de programmation, ainsi que différentes techniques de programmation avancée issues de la programmation orientée-objet et fonctionnelle : introspection, récursivité terminale, map-reduce-filter, fonctions anonymes, compréhensions de listes, fermetures, décorateurs, etc.

Trois séances de cours sont à la disposition du professeur en NON-PRÉSENTIEL.

## 3. Stratégies pédagogiques :

Le cours se donne sous forme magistrale de trois (3) heures par semaine, complétés par 8 séances de TD, pour une durée de quinze (15) semaines. Les cours magistraux et les TD se feront en présentiel. Mais toutefois, trois (03) séances de cours en non présentiel peuvent être utilisées.

Plusieurs devoirs seront aussi proposés, qui font partie intégrante de l'apprentissage.

Les étudiant(e)s qui s'inscrivent à ce cours doivent s'assurer qu'ils ont accès à : un ordinateur ; une connexion Internet ; une webcam ; un microphone ; la suite Office 365 (les étudiant(e)s ont un accès gratuit à la suite Office 365 : <https://uqo.ca/sti/outils-numeriques>).

Mode d'enseignement au trimestre d'automne 2026 : **PRÉSENTIEL**.

Trois séances de cours sont à la disposition du professeur en NON-PRÉSENTIEL.

## 4. Heures de disponibilité ou modalités pour rendez-vous :

Sur rendez-vous par courriel : [mohamed.taleb@uqo.ca](mailto:mohamed.taleb@uqo.ca) et/ou sur Zoom

## 5. Plan détaillé du cours sur 15 semaines :

| Semaine | Thèmes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Dates         |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 1       | <b>Présentation du Plan de cours</b><br><br><b>Principaux paradigmes des langages de programmation</b> <ul style="list-style-type: none"> <li>• Historique</li> <li>• Importance des langages de programmation de haut niveau</li> <li>• Principaux paradigmes : langages impératifs/procéduraux, déclaratifs, fonctionnels et orientés objet</li> <li>• Compilation versus interprétation</li> <li>• Expression et arbre syntaxique : notation préfixe, suffixe et infixé</li> <li>• Tokens (expressions régulières) et notation BNF</li> <li>• Analyse syntactique</li> </ul> | 08 sept. 2026 |
| 2       | <b>Langages logiques avec PROLOG</b> <ul style="list-style-type: none"> <li>• Historique</li> <li>• Analyse sémantique</li> <li>• Types de données</li> <li>• Les faits et base de faits</li> <li>• Procédures</li> <li>• Les questions</li> <li>• Les règles</li> <li>• Les prédicats et formules</li> <li>• Le COUPE-CHOIX (!)</li> <li>• Les variables anonymes</li> <li>• Clauses de Horn</li> </ul> <b>Travail dirigé #1 (17 septembre 2026) : Les clauses</b>                                                                                                             | 15 sept. 2026 |
| 3       | <b>Langages logiques avec PROLOG (suite)</b> <ul style="list-style-type: none"> <li>• Requêtes avec valeurs et avec variables comme paramètres</li> <li>• Processus par déduction</li> <li>• Unification et résolution</li> <li>• Opérateurs arithmétiques</li> <li>• Opérateurs logiques et relationnels</li> <li>• Opérateurs et opérations booléens</li> </ul> <b>Travail dirigé #2 (24 septembre 2026) : Unification, les booléens</b>                                                                                                                                      | 22 sept. 2026 |
| 4       | <b>Langages logiques avec PROLOG (suite)</b> <ul style="list-style-type: none"> <li>• Qualificateurs : fail, forall</li> <li>• Le traitement des listes de la programmation logique</li> </ul> <b>Travail dirigé #3 (01 octobre 2026) : Les listes</b>                                                                                                                                                                                                                                                                                                                          | 29 sept. 2026 |
| 5       | <b>Les langages fonctionnels avec Common LISP</b><br><br><b>Éléments de base de la programmation fonctionnelle</b> <ul style="list-style-type: none"> <li>• Historique</li> <li>• Analyse sémantique</li> <li>• Types de données</li> <li>• Représentation des listes</li> <li>• Expressions et fonctions</li> <li>• Arité des fonctions</li> <li>• Évaluation des expressions</li> <li>• Les fonctions mathématiques prédéfinies</li> </ul>                                                                                                                                    | 06 oct. 2026  |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                        |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
|    | <ul style="list-style-type: none"> <li>• Les fonctions prédéfinies de construction de listes</li> <li>• Les fonctions prédéfinies d'accès aux listes</li> </ul> <p><b>Travail dirigé #4 (08 octobre 2026) :</b> Expressions booléennes, fonctions de construction de listes, fonctions d'accès aux listes</p>                                                                                                                                                    |                        |
| 7  | <b>Semaine d'études</b>                                                                                                                                                                                                                                                                                                                                                                                                                                          | 12 oct. – 16 oct. 2026 |
| 6  | <p><b>Les langages fonctionnels avec Common LISP (suite)</b></p> <p><b>Éléments de syntaxe</b></p> <ul style="list-style-type: none"> <li>• Flux de contrôles</li> <li>• Variables et liens</li> <li>• Contexte et lien enchâssé</li> <li>• Définition des fonctions</li> <li>• Fonction d'ordre supérieur</li> <li>• Composition des fonctions</li> </ul> <p><b>Travail dirigé #5 (22 octobre 2026) :</b> Traitement des listes et définition des fonctions</p> | 20 oct. 2026           |
| 8  | <p><b>Les langages fonctionnels avec Common LISP (suite)</b></p> <ul style="list-style-type: none"> <li>• Variables et affectations</li> <li>• Création de modification d'une liste</li> <li>• Récursivité</li> <li>• Structures, arbres, nombres, tri, et fouille</li> </ul> <p><b>Travail dirigé #6 (29 octobre 2026) :</b> Structure, arbre, nombres, tri, fouille</p>                                                                                        | 27 oct. 2026           |
| 9  | <b>Examen intra</b>                                                                                                                                                                                                                                                                                                                                                                                                                                              | 03 nov. 2026           |
| 10 | <p><b>Programmation orientée-objet avec Ruby</b></p> <ul style="list-style-type: none"> <li>• Historique</li> <li>• Analyse sémantique</li> <li>• Types de données</li> <li>• Variables</li> </ul>                                                                                                                                                                                                                                                               | 10 nov. 2026           |
| 11 | <p><b>Programmation orientée-objet avec Ruby (suite)</b></p> <ul style="list-style-type: none"> <li>• Chaînage d'instructions d'affectation</li> <li>• Instructions d'affectation parallèles</li> <li>• Tableaux réguliers et associatifs</li> </ul> <p><b>Travail dirigé #7 (12 novembre 2026) :</b> Tableaux réguliers, tableaux associatifs, fonctions</p>                                                                                                    | 17 nov. 2026           |
| 12 | <p><b>Programmation orientée-objet avec Ruby (suite)</b></p> <ul style="list-style-type: none"> <li>• Flux de contrôle</li> <li>• Entrées/Sorties Classes, objet et passage de messages</li> <li>• Modules</li> <li>• Les expressions régulières</li> <li>• Introspection</li> </ul> <p><b>Travail dirigé #8 (19 novembre 2026) :</b> Héritage, itérateur, expression régulière,</p>                                                                             | 24 nov. 2026           |

|    |                                                   |              |
|----|---------------------------------------------------|--------------|
|    | modules, introspection                            |              |
| 13 | <b>Étude comparative des langages</b>             | 01 déc. 2026 |
| 14 | <b>Examen final</b>                               | 08 déc. 2026 |
| 15 | <b>Présentation des projets en NON-PRÉSENTIEL</b> | 15 déc. 2026 |

**6. Évaluation du cours :**

- Projet : 30 %
- Examen intra : 30 %
- Examen final : 40 %

**7. Politiques départementales et institutionnelles :**

- Politique du département d'informatique et d'ingénierie relative à la tenue des examens
- Note sur le plagiat et sur la fraude
- Politique relative à la qualité de l'expression française écrite chez les étudiants et les étudiantes de premier cycle à l'UQO
- Absence aux examens : cadre de gestion, demande de reprise d'examen (formulaire)

Tolérance **ZÉRO** en matière de violence à caractère sexuel.

Le Bureau d’intervention et de prévention en matière de harcèlement (BIPH) a pour mission d’accueillir, soutenir et guider toute personne vivant une situation de harcèlement, de discrimination ou de violence à caractère sexuel. Le BIPH oriente ses actions afin de prévenir les violences à caractère sexuel pour que nous puissions étudier, travailler et s’épanouir dans un milieu sain et sécuritaire.

Vous vivez ou êtes une personne témoin d’une situation de violence à caractère sexuel ? Vous êtes une personne membre de la communauté étudiante ou une personne membre du personnel, autant à Gatineau qu’à Ripon et St-Jérôme, l’équipe du BIPH est là pour vous, sans jugement et en toute confidentialité.

Ensemble, participons à une culture de respect.

Pour de plus amples renseignements consultez [UQO.ca/biph](https://uqo.ca/biph) ou écrivez-nous au [Biph@uqo.ca](mailto:Biph@uqo.ca)

**8. Principales références :**

**Volumes de référence pour le cours :**

1. Michael L. Scott. Programming Language Pragmatics, Third Edition, Elsevier, 2009.
2. Ivan Bratko. Prolog Programming for Artificial Intelligence, Fourth Edition, Addison Wesley, 2012.
3. Leon Sterling and Ehud Shapiro. The Art of Prolog: Advanced Programming Techniques, second edition, MIT Press, 1994.
4. Patrick Henry Winston and Berthold Klaus Paul Horn. LISP, 3<sup>rd</sup> Edition, 1989.
5. Sonja E, Keene. Object-Oriented Programming in Common Lisp, Addison Wesley, 1989.
6. Michael Fitzgerald. Learning Ruby, O’Reilly, 2007.
7. Jeri R. Hanly and Elliot B. Koffman. C Program Design for Engineers, Second Edition, Addison Wesley, 2001.

Le matériel du cours présenté en classe par le professeur est disponible sur <https://moodle.uqo.ca>.

**9. Page Web du cours :**

<https://moodle.uqo.ca>