

Sigle : INF4083 Gr. 01

Titre : Langages de programmation

Session : Automne 2025 Horaire et local

Professeur : Davoust, Alan

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 : procédurale, fonctionnelle, orientée 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 leur compétences à de nouveaux langages.

D'un point de vue théorique, ce cours a pour but de relier les langages de programmation aux langages formels (langages réguliers et hors-contextes, lambda-calcul, logique du premier ordre).

D'un point de vue pratique, à travers l'apprentissage du langage multi-paradigme Python, 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.

Le cours inclut aussi une introduction à la programmation logique avec le langage Prolog.

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.

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>).

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

- Consultations : jeudi 13h-15h ou sur rendez-vous.
- Courriel : alan.davoust@uqo.ca

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	<p>Présentation du cours</p> <p>Introduction à la théorie des langages de programmation</p> <ul style="list-style-type: none"> • Historique • Architecture et langages « Von Neumann » • Principaux paradigmes de programmation: impérative, orientée-objet, fonctionnelle, parallèle, logique • Calculabilité, thèse de Church-Turing 	2 sept. 2025

2	<p>Syntaxe des langages, compilation et interprétation</p> <ul style="list-style-type: none"> • Processus de compilation et d'interprétation • Analyse lexicale et expressions régulières • Grammaires hors-contexte, forme BNF • Grammaires attribuées, analyse sémantique <p>TD #1 : 11/12 septembre 2025</p>	9 sept. 2025
3	<p>Lambda calcul</p> <ul style="list-style-type: none"> • Fonctions • Syntaxe : variable, abstraction fonctionnelle, application • Notion de variables libres et liées : alpha-conversion • Beta-réduction <p>TD #2 : 18/19 septembre 2025</p>	16 sept. 2025
4	<p>Lambda-calcul (suite)</p> <ul style="list-style-type: none"> • Implémentation des booléens • Entiers de Church <p>Notions de sémantique formelle</p> <ul style="list-style-type: none"> • Sémantique formelle du lambda-calcul 	23 sept. 2025
5	<p>Anatomie d'un langage de programmation : variables, expressions, types</p> <ul style="list-style-type: none"> • Expressions: notation préfixe, infixe, postfixe • Sémantique formelle des expressions • Systèmes de types • Typage statique, dynamique, sûr • Inférence de types <p>Gestion de la mémoire</p> <ul style="list-style-type: none"> • Portée des associations noms-valeurs • Pile et tas • Ramasse-miettes <p>TD #3 : 2/3 octobre 2025</p>	30 sept. 2025
6	<p>Anatomie d'un langage de programmation : classes</p> <ul style="list-style-type: none"> • Approche orientée-objet • Classes, objets, métaclasse • Héritage et polymorphisme <p>Anatomie d'un langage de programmation : fonctions</p> <ul style="list-style-type: none"> • Sémantique des fonctions • Gestion de la pile • Fonctions comme objets de première classe • Fonctions anonymes (lambda) • Fermetures 	7 oct. 2025
7	<p>Semaine d'études</p>	14 oct. 2025
8	<p>Examen de mi-session</p>	21 oct. 2025
9	<p>Programmation fonctionnelle</p> <ul style="list-style-type: none"> • Propriétés des langages fonctionnels • Manipulation de listes • Expressions conditionnelles • Compréhensions de listes <p>TD #4 : 30/31 octobre 2025</p>	28 oct. 2025

10	Programmation fonctionnelle (suite) <ul style="list-style-type: none"> • Récursivité • Récursivité terminale • Continuations <p>TD #5 : 6/7 novembre 2025</p>	4 nov. 2025
11	Programmation fonctionnelle (suite) <ul style="list-style-type: none"> • Fonctions d'ordre supérieur • Fonctions Map, reduce, filter • Évaluation paresseuse • Générateurs, coroutines <p>TD #6 : 13/14 novembre 2025</p>	11 nov. 2025
12	Programmation logique avec Prolog <ul style="list-style-type: none"> • Concepts et syntaxe de Prolog : termes, foncteurs, atomes, nombres • Faits et règles • Requêtes • Substitution et unification • Recherche de preuves • Arithmétique <p>TD #7 : 20/21 novembre 2025</p>	18 nov. 2025
13	Programmation logique avec Prolog <ul style="list-style-type: none"> • Séquences de requêtes • Conditions • Récursivité • Fonctions • Manipulation de listes • Coupures <p>TD #8 : 27/28 novembre 2025</p>	25 nov. 2025
14	Révisions pour l'examen final, exercices divers.	2 dec. 2025
15	Examen final	9 dec. 2025

6. Évaluation du cours :

- 3 devoirs : 30 %
- Examen de mi-session : 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

8. Principales références :

Volumes de référence pour le cours :

1. Michael L. Scott. Programming Language Pragmatics, Third Edition, Elsevier, 2009.
2. Allen B. Downey. Think Python 2nd edition, Green Tea Press, 2015, <https://greenteapress.com/wp/think-python-2e/>
3. Ivan Bratko. Prolog Programming for Artificial Intelligence, Fourth Edition, Addison Wesley, 2012.
4. Leon Sterling and Ehud Shapiro. The Art of Prolog: Advanced Programming Techniques, Second edition, MIT Press, 1994.
5. 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>