

Département d'informatique et de recherche opérationnelle (DIRO)
Session d'automne 2025

IFT2035 (3 crédits)

CONCEPTS DES LANGAGES DE PROGRAMMATION

Dernière mise à jour le 30 août 2025

INFORMATIONS

PERSONNEL ENSEIGNANT

Mattéo Delabre

Chargé de cours

matteo.delabre
@umontreal.ca

Frédéric Lahaie-Bertrand

Auxiliaire

frderic.lahaie-bertrand
@umontreal.ca

Luchino Allix-Lastrego

Auxiliaire

luchino.allix-lastrego
@umontreal.ca

HORAIRE DU COURS

Cours	Mardi	15h30 – 17h00	B-2325 pav. 3200 Jean-Brillant
Cours	Mercredi	9h00 – 10h30	AA-1140 pav. André-Aisenstadt
Démo	Mercredi	10h30 – 12h30	AA-1140 pav. André-Aisenstadt

DATES IMPORTANTES

Premier cours	Mardi 2 septembre
Modification du choix de cours	Jusqu'au mercredi 17 septembre
Semaine de relâche	Du 20 au 26 octobre
Abandon avec frais	Jusqu'au vendredi 7 novembre
Dernier cours	Mardi 9 décembre

CONTENU DU COURS

OBJECTIFS D'APPRENTISSAGE

À ce stade de votre programme, vous êtes certainement familiers avec plusieurs langages de programmation. Il existe cependant [des centaines de langages de programmation différents](#). Lesquels choisir? Comment les apprendre? Qu'est-ce qui les distingue? Comment fonctionnent-ils? Pourriez-vous créer le vôtre? et pourquoi?

Tous ces langages peuvent être catégorisés selon différents *paradigmes*. Le premier objectif du cours est de vous familiariser avec cinq principaux paradigmes ([fonctionnel](#), [procédural](#), [logique](#), [objet](#) et [concurrent](#)). Nous illustrerons chacun d'entre eux en étudiant un langage *spécimen*. Bien que ce cours n'ait pour objectif de vous rendre expert dans aucun de ces langages, leur étude vous permettra d'appréhender des concepts qui se retrouveront dans la plupart de ceux que vous utiliserez à l'avenir.

Le second objectif du cours est de comprendre comment ces langages fonctionnent, comment ils sont définis et implantés. Notre étude des paradigmes de programmation sera parsemée d'interludes au cours desquels nous en explorerons les aspects théoriques.

À l'issue du cours, vous serez capables de:

- Apprendre à utiliser de nouveaux langages de programmation en peu de temps.
- Comparer les propriétés de différents langages de programmation.
- Identifier les paradigmes et les langages appropriés pour accomplir une tâche.
- Concevoir des langages simples en spécifiant leur syntaxe et leur sémantique.
- Réaliser des interprètes de base pour des langages simples.

CONTENU DÉTAILLÉ

1. **Programmation fonctionnelle** (*spécimen: OCaml*).
 λ -calcul, composition, fonctions d'ordre supérieur, types algébriques.
2. **Syntaxe et sémantique.**
BNF, arbre de syntaxe abstraite, analyse, règles sémantiques, interprètes.
3. **Fonctions et portées.**
Portée statique ou dynamique, fermetures, passage par référence, valeur, ou nom.
4. **Programmation procédurale** (*spécimen: Zig*).
État, séquençage, boucles, procédures, erreurs.
5. **Gestion de la mémoire.**
Pointeurs, pile d'appels, tas, allocation, structures, alignement.

6. Programmation logique (*spécimen: Prolog*).

Termes, unification, résolution, bases de données, contraintes.

7. Types avancés.

Typage statique et dynamique, sous-typage, polymorphisme paramétré.

8. Programmation objet (*spécimen: Io*).

Objets, méthodes, prototypes, héritage, classes, polymorphisme d'inclusion.

9. Métaprogrammation.

Macros, templates, génération de code.

10. Programmation concurrente (*spécimen: Erlang*).

Passage de messages ou mémoire partagée, processus.

LIVRES RECOMMANDÉS

Il n'y a pas de livre obligatoire ou de référence associé à ce cours. Les énoncés de tous les exercices et de tous les travaux seront fournis intégralement sur Studium.

Voici les livres recommandés pour chacun des cinq langages de programmation que nous utiliserons. Tous ces livres sont gratuitement accessibles en ligne.

- **OCaml:** «OCaml Programming», Clarkson (2025)
- **Zig:** «Introduction to Zig», Faria (2025)
- **Prolog:** «The Power of Prolog», Triska (2025)
- **Io:** Io Programming Guide (2021)
- **Erlang:** «Learn You Some Erlang», Hebert (2015)

Voici d'autres livres qui pourraient vous intéresser si vous cherchez à approfondir la matière du cours.

- «Programming Language Design Concepts», Watt (2004)
- «Structure and Interpretation of Computer Programs», Abelson et Sussman (2^e édition, 1996)
- «Seven Languages in Seven Weeks», Tate (2010)

ÉVALUATIONS

L'objectif des évaluations est de vérifier votre assimilation de la matière et de vous aider à cibler rapidement les notions que vous devriez retravailler. Tout travail remis ou examen passé recevra une **rétroaction dans les deux semaines qui suivent**.

TESTS

Il y a 5 tests au total, environ aux deux semaines. Les tests se déroulent en classe, sont individuels, et durent 15 minutes. Il n'y a aucune reprise pour les tests. La note des tests est la moyenne de tous les tests, en excluant la plus basse note.

DEVOIRS

Il y a 3 devoirs au total, qui portent sur les sujets suivants:

1. *Interprète pour un langage fonctionnel simple (en OCaml)*.
2. *Allocateur de mémoire (en Zig)*.
3. *Système d'inférence de types (en Prolog)*.

Les devoirs se font en équipe de deux. Les remises se font exclusivement sur Studium. **Aucun devoir remis en retard ne sera accepté**. La note des devoirs est la moyenne de tous les devoirs.

EXAMENS

Le premier examen intra dure 1 heure, le second dure 2 heures et l'examen final dure 3 heures. Les examens sont récapitulatifs. Ils sont individuels et ont lieu en personne, sur papier, sans recours à aucun document, appareil ou moyen de communication.

Intra 1	Mercredi 8 octobre	10h30 – 11h30	AA-1140
Intra 2	Mercredi 12 novembre	10h30 – 12h30	AA-1140
Final	Mercredi 10 décembre	9h30 – 12h30	<i>À venir.</i>

Toute absence à un examen doit être motivée au plus tôt (voir [sections 9.7 à 9.9 du règlement des études](#)). En cas d'absence motivée à un ou plusieurs examens intra, la note moyenne des autres examens sera reportée. En cas d'absence motivée à l'examen final, il y aura une reprise. Toute absence non motivée engendrera une note de zéro.

BARÈME

Tests	10%
Devoirs	15%
Intra 1	12,5%
Intra 2	25%
Final	37,5%

SITUATIONS DE HANDICAP ET ACCOMMODEMENTS

Si vous vivez une situation de handicap, des accommodements sont disponibles pour vous soutenir. Pour y accéder, vous devez **obligatoirement** être inscrit auprès du *Soutien aux personnes étudiantes en situation de handicap* (SPÉSH), et ce, **au plus tard le mercredi 17 septembre**. Il est également recommandé d'informer le chargé de cours par courriel de vos besoins avant cette date afin que nous puissions convenir d'accommodements si nécessaire. Consultez le [site des Services à la vie étudiante pour plus de détail](#).

INTÉGRITÉ

PLAGIAT ET FRAUDE

Les manquements à l'intégrité peuvent prendre plusieurs formes, par exemple, la triche pendant un examen, la soumission du même travail dans plusieurs cours, l'invention de faits ou de sources d'information ou l'utilisation d'aide extérieure non-autorisée pour réaliser un travail. Les sanctions pour de tels manquements s'échelonnent jusqu'au renvoi ou à l'exclusion de l'université. Consultez le [site Intégrité](#) pour plus d'information.

INTELLIGENCE ARTIFICIELLE GÉNÉRATIVE

Dans ce cours, les outils d'IA générative (IAg) peuvent être employés en soutien aux activités d'apprentissage des personnes participantes, par exemple:

- pour aider à comprendre des concepts du cours;
- pour générer des questions afin d'étudier pour les examens;
- pour analyser et reformuler les énoncés des devoirs;
- pour écrire une partie limitée (moins de 50%) du code remis pour un devoir;
- pour effectuer une révision linguistique des textes produits.

L'emploi de ces outils est toutefois interdit pour l'écriture des rapports remis avec les devoirs. La transparence est de mise: l'utilisation du [Formulaire de déclaration de l'utilisation de l'intelligence artificielle générative](#) est recommandée. Il est recommandé de consigner les conversations avec l'IAg. [Le site web des bibliothèques offre des repères à ce sujet](#). Il est de la responsabilité de la personne étudiante d'assurer l'exactitude du matériel utilisé, généré par ces outils. Lors des travaux en groupe, tous les membres du groupe sont tenus responsables du contenu remis.