

Sigle : INF1563 Gr. 01
Titre : Programmation I
Session : Automne 2026 **Horaire et local**
Professeurs : Davoust, Alan et Shangwe, Charmant

1. Description du cours paraissant à l'annuaire :

Objectifs

Au terme de ce cours, l'étudiant.e. sera en mesure de comprendre des problèmes simples, de formuler des solutions algorithmiques et de les implémenter dans un langage de programmation procédural.

Contenu

Introduction à la résolution de problèmes : analyse d'un problème, conception des solutions, codage des programmes dans un langage procédural (**Python**, C, etc.). Principes de la programmation : variables, constantes, expressions, instructions, types de données, structures de contrôle, procédures et fonctions. Bonnes pratiques de programmation : style et formatage, documentation. Introduction aux tests. Récursivité. Traitement des erreurs et gestion des exceptions. Ce cours comporte des séances obligatoires de travaux dirigés (TD).

[Descriptif – Annuaire](#)

2. Objectifs spécifiques du cours :

Ce cours vise à donner aux étudiant(e)s une connaissance des bases de la programmation impérative. Le cours couvre les huit unités suivantes, pour lesquelles on indique les compétences attendues en termes de savoir-faire :

- 1) Expressions Python
 - Écrire une expression pour implémenter un calcul numérique ou logique simple
 - Interpréter une expression numérique ou booléenne Python
- 2) Éléments de base de la programmation impérative : instructions, entrées/sorties
 - Décomposer un calcul en une série d'affectations utilisant des variables
 - Lire des données depuis le clavier, afficher des informations à l'écran
- 3) Algorithmique de base avec contrôle de flux
 - Écrire des algorithmes simples utilisant des structures conditionnelles et des boucles
 - Choisir les structures de contrôle de flux les plus appropriées pour un problème
 - Tracer un programme utilisant des structures de contrôle de flux
- 4) Fonctions
 - Utiliser correctement une fonction existante
 - Formuler les entrées et sorties d'un problème simple par un en-tête de fonction
 - Écrire le contenu d'une fonction en utilisant les paramètres et le type de retour
 - Écrire des tests unitaires pertinents pour une fonction simple
- 5) Récursivité
 - Comprendre la récursivité et formuler des algorithmes récursifs simples
- 6) Les chaînes de caractères, listes et tuples
 - Accéder aux éléments d'une liste ou d'un tuple, aux sous-chaînes d'une chaîne de caractères
 - Parcourir des listes à l'aide de boucles, implémenter des algorithmes pertinents aux collections (recherche, comptage, somme, etc.)
- 7) Les dictionnaires et ensembles
 - Manipuler un dictionnaire : insérer ou modifier une paire clé-valeur, accéder aux éléments, lister les appariements
 - Comprendre le fonctionnement des ensembles (*set*)
 - Manipuler un ensemble : insérer ou supprimer une valeur, vérifier l'appartenance
- 8) Techniques diverses
 - Manipulation de fichiers
 - Gestion d'exceptions
 - Utilisation de dates et heures
 - Notions de programmation orienté objet

3. Stratégies pédagogiques :

Les formules pédagogiques suivantes seront utilisées :

- Toutes les activités d'enseignement ont lieu en présentiel.
- Des lectures et capsules vidéo seront proposées afin de permettre aux étudiant.e.s de préparer les cours et d'apprendre à leur rythme
- Les séances de cours magistraux seront utilisées principalement pour de la programmation sous forme « tutoriel » ainsi que des exercices: il est donc fortement encouragé d'apporter, si possible, un ordinateur portable en classe.
- Des **séances de travaux dirigés** (2 h/semaine) viendront appuyer les séances de cours. Ces séances se feront dans un laboratoire de l'UQO et seront encadrées par un(e) assistant(e).

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

- Heures de consultation : mercredi 13h-15h
- Possibilité de rencontres à d'autres horaires et éventuellement sur Zoom, à demander par courriel

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	Introduction - Présentation du cours, exigences, calendrier, ressources - Matériel et logiciel, Environnement de développement - Notion d’algorithme Notions de base : - Variables et expressions - Instructions : affectations, entrées et sorties de base TD 1 : les 10, 11, 14 septembre 2026.	9 sept. 2026
2	Contrôle de flux - Instructions conditionnelles - Blocs d'instructions TD 2 : les 17, 18, 21 septembre 2026	16 sept. 2026
3	Contrôle de flux (suite) Instructions répétitives TD 3 : les 24 et 28 septembre 2026 (25 Septembre : jour de la vérité et réconciliation)	23 sept. 2026
4	Contrôle de flux (suite) (livre chapitre 4) Algorithmique avec des boucles : compter, trouver, faire une somme, etc. TD 3/4 : les 1, 2 octobre 2026 (5 octobre : Élections provinciales)	30 sept. 2026
5	Les fonctions - Fonctions et procédures - Utiliser une fonction - Spécifier et implémenter une fonction TD 4 : le 9 octobre 2026	7 oct. 2026
6	Semaine d'études	14 oct. 2026
7	Examen Intra #1 (1h30) Correction de l’examen TD 5 : les 22, 23, 26 octobre 2026.	21 oct. 2026
8	Les fonctions (suite) - Paramètres optionnels - Tests unitaires Réursivité - Fonctions réursives TD 6 : le 29, 30 octobre et 2 novembre 2026	28 oct. 2026
9	Les listes et tuples - Accès aux éléments, indices, sous-listes - Parcours de listes, algorithmes - Listes vs. tuples TD 7 : le 5, 6 et 9 novembre 2026	4 nov. 2026
10	Les dictionnaires et ensembles - Dictionnaires - Ensembles TD 8 : le 12, 13, 16 novembre 2026	11 nov. 2026
11	Examen intra #2 (3h)	18 nov. 2026
12	Manipulation de fichiers texte - Ouverture, lecture-écriture et fermeture de fichier - Modes d'ouverture et gestionnaires de contexte Gestion des erreurs - Erreurs et Exceptions TD 9 : le 26, 27, 30 novembre 2026	25 nov. 2026
13	Brève présentation de la programmation orientée objet en python	2 dec. 2026

	- Classes, Objets et méthodes Utilisation de la bibliothèque datetime - Programmation avec date et heure TD 10 : le 3, 4, 7 Décembre	
14	Révisions et exercices divers	9 dec. 2026
15	Examen Final (3h)	16 dec. 2026

6. Évaluation du cours :

L'évaluation de ce cours se fait par compétences (en lien avec les unités listées plus haut).
Il y aura 3 examens, qui couvriront les unités suivantes :

- Examen intra 1** : Unités 1 à 3 (40% de pondération)
- Examen intra 2** : Unités 4 à 6 (40% de pondération)
- Examen final** : Unités 7 et 8 (20% de pondération)
- Les examens intra 2 et final incluront des exercices portant sur les unités évaluées précédemment, permettant de rattraper les notes des examens précédents.
- Deux devoirs seront proposés : ces devoirs donneront lieu à une évaluation purement formative (sans impact sur la note du cours)

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 ou écrivez-nous au Biph@uqo.ca

8. Principales références :

Référence principale : Les notes de cours, les notebooks et vidéos associées

Références recommandées : (disponibles en ligne)

- Swinnen, G. Apprendre à programmer avec Python 3. *Disponible en ligne:* https://inforef.be/swi/download/apprendre_python3_5.pdf
- Downey, A. (2016). Think Python 2nd edition. How to Think Like a Computer Scientist. Needham, Massachusetts: Green Tea Press. *Traduction française disponible sur la page Moodle du cours (pdf).*
- Al Sweigart. 2019. Automate the Boring Stuff with Python: Practical Programming for Total Beginner, 2nd Edition, No Starch Press.

9. Page Web du cours :

<http://moodle.uqo.ca>