

Sigle : INF1563 Gr. 01

Titre : Programmation I

Session : Hiver 2025 Horaire et local

Professeur : Davoust, Alan

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 six unités suivantes, pour lesquelles on indique les compétences attendues en termes de savoir-faire :

- 1) Éléments de base de la programmation impérative avec Python : expressions, fonctions, entrées/sorties
 - Écrire une expression pour implémenter un calcul numérique ou logique simple
 - Décomposer un calcul en une série d'affectations utilisant des variables
 - Lire des données depuis le clavier, afficher des informations à l'écran
- 2) 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
- 3) 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
- 4) Récursivité
 - Comprendre la récursivité et formuler des algorithmes récursifs simples
- 5) Les chaînes de caractères et collections
 - 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.)
 - Manipuler un dictionnaire : insérer ou modifier une paire clé-valeur, accéder aux éléments, lister les appariements
- 6) Techniques diverses :
 - Manipulation de fichiers
 - Gestion d'exceptions
 - Notions de programmation orienté objet
 - Utilisation de quelques bibliothèques

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 : Lundi 13h-15h, bureau B2026
- Possibilité de rencontres à d'autres horaires et éventuellement sur Zoom, RV par courriel : alan.davoust@uqo.ca

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	Introduction <ul style="list-style-type: none"> Présentation du cours, exigences, calendrier, ressources Matériel et logiciel, Environnement de développement Notion d'algorithme Notions de base : <ul style="list-style-type: none"> Variables et expressions Instructions : affectations, entrées et sorties de base TD 1 : le 17 janvier 2025. TD 1 : le 27 janvier 2025.	15 janv. 2025
2	Contrôle de flux <ul style="list-style-type: none"> Instructions conditionnelles Blocs d'instructions TD 2 : le 24 janvier 2025. TD 2 : le 27 janvier 2025.	22 janv. 2025
3	Contrôle de flux (suite) <ul style="list-style-type: none"> Instructions répétitives TD 3 : le 31 janvier 2025 TD 3 : le 3 février 2025.	29 janv. 2025
4	Contrôle de flux (suite) (livre chapitre 4) <ul style="list-style-type: none"> Algorithmique avec des boucles : compter, trouver, faire une somme, etc. TD 4 : le 7 février 2025. TD 4 : le 10 février 2025.	5 févr. 2025
5	Les fonctions <ul style="list-style-type: none"> Fonctions et procédures Utiliser une fonction Spécifier et implémenter une fonction TD 5 : le 14 février 2025. TD 5 : le 17 février 2025.	12 févr. 2025
6	Examen Intra #1 (1h30) Les fonctions (suite) <ul style="list-style-type: none"> Paramètres optionnels Tests unitaires TD 6 : le 20 février 2025. Gr02 9h-11h TD 6 : le 24 février 2025. Gr03 : 13h-15h	19 févr. 2025
7	Récurtivité <ul style="list-style-type: none"> Fonctions récursives TD 7 : le 28 février 2025 TD 7 : le 10 mars 2025	26 févr. 2025
8	Semaine d'études	5 mars 2025
9	Les listes et tuples <ul style="list-style-type: none"> Accès aux éléments, indices, sous-listes Parcours de listes, algorithmes Listes vs. tuples TD 8 : le 7 mars 2025 TD 8 : le 10 mars 2025	12 mars 2025
10	Les dictionnaires <ul style="list-style-type: none"> Dictionnaires Ensembles TD 9 : le 21 mars 2025 TD 9 : le 24 mars 2025	19 mars 2025
11	Examen intra #2 (3h)	26 mars 2025
12	Manipulation de fichiers texte <ul style="list-style-type: none"> Ouverture, lecture-écriture et fermeture de fichier Modes d'ouverture et gestionnaires de contexte TD 10 : le 4 avril 2025 TD 10 : le 7 avril 2025	2 avril 2025

13	Gestion des erreurs <ul style="list-style-type: none"> Erreurs et Exceptions Brève présentation de la programmation orientée objet en python <ul style="list-style-type: none"> Classes, Objets et méthodes 	9 avril 2025
14	Survol de quelques bibliothèques externes: <ul style="list-style-type: none"> Numpy Requests Matplotlib Exercices divers	16 avril 2025
15	Examen final	23 avril 2025

6. Évaluation du cours :

L'évaluation de ce cours se base sur les 6 unités présentées dans la section (2) ci-dessus, auxquelles sont associés les nombres de points suivants (total = 100) :

- (1) Éléments de base de la programmation impérative **(20 points)**
- (2) Algorithmique de base avec contrôle de flux **(20 points)**
- (3) Fonctions **(20 points)**
- (4) Récursivité **(10 points)**
- (5) Utilisation de listes, tuples et dictionnaires **(20 points)**
- (6) Techniques diverses **(10 points)**

Chaque unité sera évaluée dans un ou plusieurs examens :

- Unités 1 et 2 : examens intra #1 et #2, examen final
- Unités 3, 4, 5 : examen intra #2, examen final
- Unité 6 : examen final

À chaque évaluation d'une unité, cette unité sera notée qualitativement N/P/A/D selon si les objectifs associés sont (N)on atteints, (P)artiellement atteints, (A)tteints ou (D)épassés.

Des points sont associés à ces appréciations selon le barème suivant :

Appréciation	Unité 1	Unité 2	Unité 3	Unité 4	Unité 5	Unité 6
N	0	0	0	0	0	0
P	10	10	10	5	10	5
A	18	18	18	9	18	9
D	20	20	20	10	20	10

Pour les unités qui sont évaluées dans plusieurs examens, les points accordés seront ceux correspondant à la meilleure note obtenue par un(e) étudiant(e).

La note finale de chaque étudiant(e) sera la somme des points obtenus pour les différentes unités, convertie en note littérale, avec les modalités suivantes :

- Deux devoirs seront proposés au cours de la session. Ces devoirs donneront chacun lieu à 3 points de bonification potentiels. Les conditions spécifiques aux devoirs seront communiquées en même temps que l'énoncé.

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 et les notebooks associés

Références recommandées : (disponibles en ligne)

- 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>