

Sigle : INF1563 Gr. 01**Titre : Programmation I****Session : Hiver 2026 Horaire et local****Professeur : Yapi, N'Dah Daniel****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 sept unités suivantes, pour lesquelles on indique les compétences attendues en termes de savoir-faire :

- 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
- 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
- Fonctions : bases
 - 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
- Fonctions : récursivité
 - Écrire des tests unitaires pertinents pour une fonction
 - Comprendre la récursivité et formuler des algorithmes récursifs simples
- Les collections
 - Collections natives
 - Listes, tuples
 - Dictionnaires
- Autres techniques importantes et perspectives :
 - Manipulation de fichiers
 - Traitement des erreurs
 - Gestion d'exceptions
 - Quelques bibliothèques pour Python
- Notion de programmation orienté objet

3. Stratégies pédagogiques :

Les formules pédagogiques suivantes seront utilisées :

- Toutes les activités d'enseignement ont lieu en présentiel.
- Les séances de cours magistraux seront utilisées en partie pour de la programmation sous forme « tutoriel » : 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 :

Par courriel et/ou sur rendez-vous sur demande : yapida01@uqo.ca

5. Plan détaillé du cours sur 15 semaines :

| Semaine | Thèmes | Dates |
|---------|---|---------------|
| 1 | Cours en ligne Zoom Introduction <ul style="list-style-type: none">• Paradigmes de programmation : le cas procédural en python• Matériel et logiciel, Environnement de développement Notions de base : <ul style="list-style-type: none">• Variables et types• Opérateurs mathématiques, opérandes• Entrées et sorties de base• Expressions | 14 janv. 2026 |
| 2 | Contrôle de flux <ul style="list-style-type: none">• Instructions conditionnelles• Blocs d'instructions• Visibilité (portée) des variables Travail dirigé 1 : les 19 et 23 janvier | 21 janv. 2026 |
| 3 | Cours en ligne Zoom Contrôle de flux (suite) <ul style="list-style-type: none">• Instructions répétitives Travail dirigé 2 les 26 et 30 janvier | 28 janv. 2026 |
| 4 | Les fonctions <ul style="list-style-type: none">• Définir une fonction• Paramètres et arguments• Valeur de retour Portée des variables : variables locales et variables globales Travail dirigé 3 les 2 et 6 février | 04 fév. 2026 |
| 5 | Cours en ligne Zoom Les fonctions (suite): concepts plus avancés <ul style="list-style-type: none">• Génération de nombres aléatoires• Fonctions récursives Travail dirigé 4 les 9 et 13 février | 11 fév. 2026 |
| 6 | Listes et tuples <ul style="list-style-type: none">• Séquences• Introduction aux listes• Découpage de liste• Recherche d'éléments dans des listes• Tuples• Accéder aux valeurs d'un tuple | 18 fév. 2026 |

| | | |
|----|--|---------------|
| | <ul style="list-style-type: none"> Tuples vs Liste Travail dirigé 5 les 16 et 20 février | |
| 7 | Examen intra | 25 fév. 2026 |
| 8 | Semaine d'études | 04 mars 2026 |
| 9 | Les dictionnaires <ul style="list-style-type: none"> Dictionnaires Ensembles Travail dirigé 6 les 09 et 13 mars | 11 mars 2026 |
| 10 | Les chaînes de caractères (Strings) <ul style="list-style-type: none"> Opérations de base sur les chaînes Formatage et opérations spécifiques aux chaînes de caractères Travail dirigé 7 les 16 et 20 mars | 18 mars 2026 |
| 11 | Scripts de manipulation des fichiers <ul style="list-style-type: none"> Ouverture, lecture-écriture et fermeture de fichier Modes d'ouverture et gestionnaires de contexte Travail dirigé 8 les 23 et 27 mars | 25 mars 2026 |
| 12 | Gestion des erreurs <ul style="list-style-type: none"> Erreurs et Exceptions | 01 avril 2026 |
| 13 | Classes et la programmation orientée objet en python <ul style="list-style-type: none"> Programmation procédurale et orientée objet Des classes Travailler avec des instances Techniques de conception de classes L'héritage <ul style="list-style-type: none"> Introduction à l'héritage Polymorphisme Travail dirigé 9 le 30 mars et le 3 avril | 08 avril 2026 |
| 14 | Survol de quelques bibliothèques tiers : <ul style="list-style-type: none"> Numpy Panda Matplotlib Seaborn | 15 avril 2026 |
| 15 | Examen final | 22 avril 2026 |

6. Évaluation du cours :

L'évaluation est l'appréciation du niveau d'apprentissage atteint par l'étudiant(e) par rapport aux objectifs des cours et des programmes

L'attribution des notes se fera selon la répartition suivante :

- Examen intra : 35%
- Examen final : 40 %
- Travaux pratiques et/ou projets: 25%

Pour le projet d'équipe, les notes peuvent être attribuées d'une manière individuelle selon la contribution de l'étudiant(e).

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 :

Référence principale : Les notes de cours du professeur et les notebooks associés

Références recommandées : (disponible en PDF)

- Downey, A. (2016). Think Python 2nd edition. How to Think Like a Computer Scientist. Needham, Massachusetts: Green Tea Press.
- Swinnen, G. Apprendre à programmer avec Python 3. Disponible en ligne:
https://inforef.be/swi/download/apprendre_python3_5.pdf
- Al Sweigart. 2019. Automate the Boring Stuff with Python: Practical Programming for Total Beginner, 2nd Edition, No Starch Press.
- **Autres ressources :** <https://www.w3schools.com/python/>

9. Page Web du cours:

<https://moodle.uqo.ca>