

**Sigle : INF1563 Gr. 20****Titre : Programmation I****Session : Automne 2024 Horaire et local****Professeur : Nguena Timo, Omer****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 procédurale. Le cours couvre les sept unités suivantes, pour lesquelles on indique les compétences attendues en termes de savoir-faire :

- 1) Fonctionnement d'un ordinateur
  - Connaître les différentes unités d'un ordinateur
  - Connaître les types de données élémentaires, leur représentation,
  - Connaître l'organisation de base de la mémoire centrale et la notion d'adresse
  - Décrire le rôle des unités d'un ordinateur lors de l'exécution d'un programme
  - Connaître les instructions élémentaires que peuvent exécuter un ordinateur
  - Connaître les niveaux de langage de programmation
  - Comprendre les notions : problèmes, donnée, solution algorithmique, programme
- 2) Éléments de base de la programmation procédurale : expressions, fonctions, entrées/sorties
  - Interpréter une expression Python
  - 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
- 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
  - 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
- 5) Utilisation de tableaux, algorithmes de listes
  - Manipuler des tableaux : en particulier, algorithmes simples utilisant des parcours de tableaux avec des boucles (somme des éléments, rechercher une valeur...)
- 6) Initiation à l'algorithmique
  - Comprendre des algorithmes simples de tri, recherche dichotomique, etc.
  - Comprendre la récursivité et formuler des algorithmes récursifs simples
- 7) Traitement des erreurs et gestion des exceptions
  - Comprendre et corriger des erreurs de syntaxe
  - Comprendre et corriger des erreurs à l'exécution
  - Gestion d'exceptions

**3. Stratégies pédagogiques :**

Les formules pédagogiques suivantes seront utilisées :

- Toutes les activités d'enseignement ont lieu en présentiel.
- Les séances de cours magistraux (3h/semaine) seront utilisées principalement pour introduire des concepts, les appliquer à la résolution de problèmes et traiter des exercices en mode « tutoriel ».
- Les séances de travaux libres (2 h/semaine) permettent de réviser les notions acquises et de mieux préparer les prochaines séances de cours magistraux. Ces séances peuvent se faire dans un laboratoire de l'UQO et seront encadrées par un(e) assistant(e). L'étudiant est invité à préparer les exercices et effectuer des lectures avant le prochain cours.
- Les mini devoirs pendant les séances de cours permettent de démontrer la compréhension des concepts étudiés.
- Les devoirs à faire à la maison permettent de démontrer l'aptitude à résoudre des problèmes en utilisant les concepts vus en cours.
- Les examens intra et finaux se feront sur papier, à livre fermé.

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

- Consultations : mercredi de 11h45 à 13h45 ou sur rendez-vous à la demande des étudiants.
- Courriel : omer.nguena-timo@uqo.ca

## 5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	<p><b>Présentation du plan de cours</b></p> <p><b>Composants d'un ordinateur</b></p> <ul style="list-style-type: none"> <li>• Unités d'entrée</li> <li>• Unités de sortie</li> <li>• Unités de stockage, bit, mot, données de différents types et représentations, adresse mémoire</li> <li>• Unités d'arithmétique et logique</li> <li>• Unités de contrôle</li> <li>• Unité de transfert de données (bus)</li> <li>•</li> </ul>	4 sept. 2024
2	<p><b>Principes de base du fonctionnement d'un ordinateur</b></p> <ul style="list-style-type: none"> <li>• Notion d'instruction élémentaires et codage</li> <li>• Types d'instructions élémentaires</li> <li>• Exécution d'une instruction élémentaire et d'une séquence d'instructions</li> <li>• Niveaux de langage de programmation (binaire, assembleur, haut niveau)</li> </ul> <p><b>Introduction la résolution des problèmes et à la programmation</b></p> <ul style="list-style-type: none"> <li>• Problèmes, donnée et variable</li> <li>• Instance d'un problème,</li> <li>• Méthodologie de résolution de problème et écriture de programme</li> </ul> <p><b>TD 1 le 12 sept. 2024 (installer PyCharm Edu, saisir et exécuter des programmes Python, installer un interpréteur Python en ligne de commande)</b></p>	11 sept. 2024
3	<p><b>Mini examen 1 (semaine 1 et 2, objectif : démontrer la compréhension des concepts)</b></p> <p><b>Langage algorithmique et Python,</b></p> <ul style="list-style-type: none"> <li>• Identifiants et mots réservés</li> <li>• Données : littéraux et types de données élémentaires</li> <li>• Les chaînes de caractères</li> <li>• Stockage de données en mémoire : adresse, pyobject</li> <li>• Réserver la mémoire pour stocker des données: variable, nom/identifiant de variable</li> <li>• Initialisation des variables</li> <li>• Acquisition et restitution des données avec Python : entrées et sorties de base (écran et clavier)</li> </ul> <p><b>Algorithme</b></p> <ul style="list-style-type: none"> <li>• Structure et syntaxe</li> <li>• Tracer un algorithme</li> </ul> <p><b>Script Python et module Python</b></p> <ul style="list-style-type: none"> <li>• Structure</li> <li>• Exécution à l'aide d'un interpréteur Python</li> <li>• Introduction aux erreurs de syntaxe et correction</li> </ul> <p><b>Devoir 1 à rendre avant le 25 sept. 2024 (saisir, exécuter, tracer et tester des algorithmes et des programmes Python)</b></p>	18 sept. 2024
4	<p><b>Traitements élémentaires sur des données :</b></p> <ul style="list-style-type: none"> <li>• Opérateurs, opérandes</li> <li>• Expressions arithmétiques (syntaxe et sémantique)</li> <li>• Expression de test (syntaxe et sémantique)</li> <li>• Expression logique (syntaxe et sémantique)</li> </ul> <p><b>Évaluer et déterminer le type d'une expression</b></p> <p><b>Modifier la donnée stockée dans d'une variable :</b></p> <ul style="list-style-type: none"> <li>• Instruction d'affectation (syntaxe et sémantique)</li> <li>• Affectation multiple</li> </ul>	25 sept. 2024

	<ul style="list-style-type: none"> <li>Introduction à la conversion des données</li> </ul> <p><b>TD 2 le 26 sept. (Semaine 3 et 4)</b></p>	
5	<p><b>Mini examen 2 (semaine 3 et 4)</b></p> <p><b>Bloc d'instructions, portée des variables</b> Instructions élémentaires et blocs d'instructions Visibilité (portée) des variables</p> <p><b>Instructions de contrôle de flux</b></p> <ul style="list-style-type: none"> <li>Exécution conditionnelle d'un bloc d'instruction : if</li> </ul> <p><b>Résoudre un problème (illustration)</b></p> <p><b>TD 3 le 03 oct. (Semaine 5)</b></p> <p><b>Devoir 2 à rendre avant le 23 oct. (résolution de problèmes simples à l'aide de programmes Python)</b></p>	02 oct. 2024
6	<p><b>Les tableaux, les listes, le tuples et les dictionnaires</b></p> <p><b>Types de données mutables et immuables en Python</b></p> <p><b>Instructions de contrôle de flux (suite)</b></p> <ul style="list-style-type: none"> <li>Exécution répétitive d'un bloc d'instruction : for, while</li> <li>Ruptures de séquence : break</li> </ul> <p><b>Algorithmes</b> : compter et trouver</p> <p><b>TD 4 le 10 oct. (Semaine 6)</b></p>	09 oct. 2024
7	<p><b>Semaine d'études</b></p>	16 oct. 2024
8	<p><b>Mini examen 3 (semaine 5, 6)</b></p> <p><b>Fonctions</b></p> <ul style="list-style-type: none"> <li>Rôle d'une fonction</li> <li>Signature d'une fonction : type de retour, nom, paramètres formels,</li> <li>Retourner une valeur</li> <li>Contexte d'exécution d'une fonction</li> <li>Spécifier (déclarer) et définir (implémenter) une fonction</li> </ul> <p><b>Appel de fonction et pile d'appel,</b></p> <ul style="list-style-type: none"> <li>Instruction d'appel de fonction</li> <li>Passage de paramètres : <ul style="list-style-type: none"> <li>cas des types de données mutables</li> <li>cas des types de données immuables</li> <li>paramètres optionnels</li> </ul> </li> </ul> <p><b>Tracer un script/module avec appel de fonction</b></p> <p><b>TD 5 le 24 oct. (Semaine 8)</b></p>	23 oct. 2024
9	<p><b>Examen intra</b></p>	30 oct. 2024
10	<p><b>Algorithmique</b></p> <ul style="list-style-type: none"> <li>Trier un tableau/une liste</li> <li>Recherche dichotomique</li> </ul> <p><b>TD 6 le 07 nov. (semaine 8 et 10)</b></p> <p><b>Devoir 3 à présenter le 20 nov.</b></p>	06 nov. 2024
11	<p><b>Fonctions récursives</b></p> <p><b>TD 7 le 14 nov. (semaine 10 et 11)</b></p>	13 nov. 2024

12	<p><b>Module</b></p> <ul style="list-style-type: none"> <li>• Définition</li> <li>• Importation de modules</li> <li>• Obtenir de l'aide sur les modules importés</li> <li>• Quelques modules courants</li> </ul> <p><b>Création de modules</b></p> <ul style="list-style-type: none"> <li>• Créer son propre module</li> <li>• Utiliser son propre module</li> <li>• Documenter son module</li> <li>• Visibilité des fonctions dans un module</li> </ul> <p><b>Fichiers</b></p> <ul style="list-style-type: none"> <li>• Lecture dans un fichier</li> <li>• Écriture dans un fichier</li> <li>• L'instruction With</li> </ul> <p><b>TD 8 le 21 nov. (semaine 12)</b></p> <p><b>Devoir 4 à remettre et présenter avant le 04 déc.</b></p>	20 nov. 2024
13	<p><b>Mini examen 4 (semaine 8, 10, 11 et 12)</b></p> <p><b>Introduction au test des programmes python</b></p> <p><b>Gestion d'erreurs : Les exceptions Python</b></p> <p><b>Introduction aux enregistrements et à la programmation orientée objet</b></p> <p><b>TD 9 le 28 nov. (semaine 13)</b></p>	27 nov. 2024
14	<p><b>Introduction aux enregistrements et à la programmation orientée objet (suite)</b></p> <p><b>Bonnes pratiques de programmation en Python</b></p> <p><b>Révisions par des exemples et divers exercices</b></p> <p><b>TD 10 le 05 déc. (semaine 13, 14)</b></p>	04 déc. 2024
15	<b>Examen final</b>	11 déc. 2024

## 6. Évaluation du cours :

L'évaluation de ce cours porte sur la connaissance des concepts et leur application à la résolution de problème et l'écriture de programme en Python. Pour réussir chaque évaluation, l'étudiant doit démontrer qu'il :

- comprend les concepts appris en cours et est capable de les expliquer à l'aide d'exemples;
  - est capable d'analyser un problème afin d'identifier les données d'entrée, de sortie et proposer un solution en langage naturel ou algorithmique;
  - est capable d'effectuer toutes les tâches de l'activité de programmation en Python (résolution des problèmes, écriture de programmes, traçage/exécution manuelle de programmes, correction des programmes Python);
- 4 mini examens en présentiel: 20 %
  - 4 devoirs à faire à la maison : 20%
  - Examen intra en présentiel : 30 %
  - Examen final en présentiel : 30 %

## 7. Politiques départementales et institutionnelles :

- Politiques relatives à la tenue des examens
- Note sur le plagiat et les fraudes
- 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](mailto:Biph@uqo.ca)

## 8. Principales références :

### Livre suggéré

*Apprendre à programmer avec Python 3*, Gerard Swinnen, 14<sup>e</sup> édition, Eyrolles, 2012, ISBN 978-2-416-00703-3.

### Autres ressources

- *cours sur Python 3 de Robert Cordeau* : <https://perso.limsi.fr/poinal/media/python:cours:courspython3.pdf>
- [https://python.sdv.univ-paris-diderot.fr/12\\_plus\\_sur\\_les\\_fonctions/](https://python.sdv.univ-paris-diderot.fr/12_plus_sur_les_fonctions/)
- <https://qayerie.dev/docs/python/python3/introduction.html>

## 9. Page Web du cours :

<http://moodle.uqo.ca>