

PLAN DE COURS

FOR-4048 : Compétences de programmation pour les sciences forestières

NRC 85119 | Automne 2021

Mode d'enseignement : À distance

Temps consacré : 3-1-5	Crédit(s) : 3
------------------------	---------------

Familiarisation avec des outils d'utilisation et d'analyse de données (bases de données relationnelles, SIG, outils statistiques et de modélisation), et de développement de procédures efficaces et reproductibles. Introduction à la programmation avec le langage Python, et à des outils de gestion et de documentation du flux de production (workflow) tels que git et Markdown.

Ce cours est offert à distance en mode synchrone, en direct, selon l'horaire indiqué. Les enregistrements des séances seront rendus disponibles sur le site Web du cours. En fonction des directives de la santé publique, veuillez prendre note que si des examens sous surveillance peuvent être réalisés, ceux-ci ont lieu en soirée ou la fin de semaine et peuvent donc se dérouler à un autre moment que la plage prévue pour les séances synchrones. Plus de détails seront fournis ultérieurement.

Plage horaire

Classe virtuelle synchrone		
mardi	09h00 à 10h20	Du 30 août 2021 au 10 déc. 2021
mercredi	12h30 à 14h50	Du 30 août 2021 au 10 déc. 2021

Il se peut que l'horaire du cours ait été modifié depuis la dernière synchronisation avec Capsule. [Vérifier l'horaire dans Capsule](#)

Site de cours

<https://sitescours.monportail.ulaval.ca/ena/site/accueil?idSite=134977>

Coordonnées et disponibilités

Steven G. Cumming

Professeur agrégé, enseignant

Abitibi-Price, local 2117

stevec@sbf.ulaval.ca


Disponibilités

Je serai disponible en tout temps par courriel, en utilisant l'outil "Envoi de courriel" à gauche, sur votre écran.

Je serai aussi disponible pour des rencontres à distance sur rendez-vous, par skype, Teams, Zoom, ou google.

Soutien technique

Équipe de soutien - Systèmes technopédagogiques (BSE)

<http://www.ene.ulaval.ca> 

418-656-2131 poste 414331

Sans frais: 1-877 7ULAAVAL, poste 414331

Automne et hiver	
Lundi au jeudi	8 h à 19 h
Vendredi	8 h à 17 h 30
Été	
Lundi au jeudi	8 h à 17 h
Vendredi	8 h à 16 h

Sommaire

Description du cours	4
Introduction	4
Objectifs généraux	4
Approche pédagogique	4
Étudiant ayant un handicap, un trouble d'apprentissage ou un trouble mental	4
Contenu et activités	4
Évaluation et résultats	6
Évaluation des apprentissages	6
Informations détaillées sur les évaluations sommatives	7
affectation et des expressions numérique	7
Docstrings et Fleurs	7
récursion	8
la séries de Fibonacci, par itération	8
Jouer avec les mots.	8
lists: reduce	9
wetlandsHist + anagrammes	9
composition des espèces d'arbres des peuplement forestières	10
Classes: Cercle, Temps, et pour essences écoforestière	10
méthodes et opérateurs pour class Pointe (Ch17.py)	11
Classes pour la composition d'un peuplement	11
Barème de conversion	12
Règles disciplinaires contre la tricherie et le plagiat	12
Correction linguistique, retard et présentation des travaux	12
Utilisation d'appareils électroniques pendant une séance d'évaluation	13
Absence aux examens	13
Étudiants ayant une situation de handicap liée à une limitation fonctionnelle	13
Matériel didactique	14
Matériel obligatoire	14
Matériel complémentaire	14
Bibliographie	15
Bibliographie	15
Annexes	15

Description du cours

Introduction

Les chercheurs actuellement utilisent des outils complexes tels que les bases de données relationnelles, les SIG, des outils de planification de la conservation et pour la simulation spatiale. On ne réussit sauf qu'on procède systématiquement, en développant des procédures qui sont efficaces, reproductibles et adaptables. Cela c'est précisément l'art de la programmation dont ce cours va vous introduire. Ce sera basé sur Python, un langage de programmation puissant et moderne, bien appuyé et très extensible. Une connaissance de Python vous aidera à maîtriser tous les problèmes de programmation que vous êtes susceptible de rencontrer dans votre carrière. Tel connaissance va agir aussi comme une fondation à partir de laquelle on peut plus facilement apprendre autres langages au meme genre, comme Ruby, Java, Julia et R.

Objectifs généraux

Le texte suivante est dérivé du cite-web "Comment maîtriser la science de l'informatique", Section 1 Le chemin du programme.

Le but de ce cours est de vous apprendre à penser comme un informaticien. Cette façon de penser combine certaines des meilleures caractéristiques des mathématiciens, de l'art de l'ingénieur et des sciences. Comme les mathématiciens, les informaticiens utilisent des langages formels pour désigner des idées (spécifiquement, des calculs). Comme les ingénieurs, ils conçoivent des choses, en assemblant des composants en systèmes et en évaluant des compromis entre les différentes solutions. Comme les scientifiques, ils observent le comportement des systèmes complexes, formulent des hypothèses et testent des prévisions.

La compétence la plus importante pour un informaticien est la capacité de résoudre des problèmes. La **résolution de problèmes** signifie la capacité de formuler des problèmes, de penser de manière créative aux solutions, et d'exprimer une solution claire et précise. Il se trouve que le processus d'apprentissage de la programmation est une excellente occasion de pratiquer les compétences de résolution de problèmes.

À un certain niveau, vous allez apprendre à programmer, une compétence utile en elle-même. Sur un autre plan, vous pourrez utiliser la programmation comme un moyen pour une fin. Au fur et à mesure que nous nous avançons, cette fin deviendra plus claire.

Les aspects de l'art de la programmation couvrent inclura les concept qu'une programme soit le combinaison des algorithmes et des structures des données, ou comment dites ces annes-ci, les objets et les opérations effectuées sur eux.

Approche pédagogique


Une combinaison des lectures théorique ainsi que pratique chaque semaine, qui suivront la séquence de chapitres dans la texte obligatoire.

Il y aura une série de devoirs pratiques hebdomadaires, qui va compter de 5 ou 10 points chacun, pour une total de 100 points pour le note finale. La suite de devoirs pourrait être adapté aux problèmes en forseterie ou biologie tel que le traitement des données écoforestières.

Il n'y a pas d'examens.

Étudiant ayant un handicap, un trouble d'apprentissage ou un trouble mental

Les étudiants qui ont une lettre d'Attestation d'accommodations scolaires obtenue auprès d'un conseiller du secteur Accueil et soutien aux étudiants en situation de handicap (ACSESH) doivent rencontrer leur professeur au début de la session afin que des mesures d'accommodation en classe ou pour les évaluations puissent être organisées. Ceux qui ont une déficience fonctionnelle ou un handicap, mais qui n'ont pas cette lettre doivent contacter le secteur ACSESH au 656-2880, le plus tôt possible.

Le secteur ACSESH vous recommande fortement de vous prévaloir des services auxquels vous avez droit afin de pouvoir réussir vos études, sans discrimination ni privilège. Pour plus d'information, voir la Procédure de mise en application des mesures d'accommodations scolaires à l'adresse suivante : <https://www.aide.ulaval.ca/situation-de-handicap/presentation/> 

Contenu et activités

Le tableau ci-dessous présente les semaines d'activités prévues dans le cadre du cours.

Titre	Date
Semaine 1	
Le cours de mardi 31 août est annulé.	
Réunion d'organisation Une réunion informelle pour réviser le plan de cours, ainsi que les besoins et buts des étudiant(e)s, est pour résoudre des conflits de planification qui peuvent survenir. Installer et démontrer Python 3.8 et IDLE, un environnement de développement intégré simple. On va l'utiliser pour les premières semaines du cours.	1 sept. 2021
Semaine 2	
Théorique 1. Le chemin du programme Notez que les nombres initiaux, comme 1 ici, se réfèrent aux chapitres du livre "Pensez en Python". Pour se préparer, faire les exercices et les exemples de Chapitre 1 du texte avec IDLE.	7 sept. 2021
Théorique et pratique Théorique: 2. Variables, expressions et instructions.	8 sept. 2021
Semaine 3	
Théorique 3. Fonctions	14 sept. 2021
Pratique 4. Étude de cas : conception d'une interface. On introduit le monde des tortues.	15 sept. 2021
Semaine 4	
Théorique 5. Structures conditionnelles et récursion	21 sept. 2021
Théorique et Pratique Théorique: 6. Fonctions productives On va introduire Anaconda (www.anaconda.com) qui inclut Spyder, une EDI pour Python beaucoup plus sophistiqué que IDLE. Voyez: https://www.anaconda.com/products/individual	22 sept. 2021
Semaine 5	
Théorique 7. Itération	28 sept. 2021
Méthodes de recherche reproductible Intégration des données, du code et de la documentation avec les cahiers Jupyter (Projet Jupyter https://jupyter.org/). On va convertir des fonctions factorielle à la mode itérative.	29 sept. 2021
Semaine 6	
Théorique 8. Chaînes de caractères	5 oct. 2021
Pratique 9. Étude de cas : jouer avec les mots Débogage (voyez Annexe A de Pensez en Python, et le documentation de IDLE). Voyez aussi https://www.cs.uky.edu/~keen/help/debug-tutorial/debug.html	6 oct. 2021
Semaine 7	
Théorique 10. Listes	12 oct. 2021
Pratique Modules et packages en Python. On va commencer à lire des données écoforestières... donc, on va créer le module CASFRI.py, pour manipuler tels données.	13 oct. 2021
Semaine 8	
Théorique 11. Dictionnaires	19 oct. 2021

Pratique 12. Tuples Illustré avec histogrammes des fréquences lettres en français, et les sorts des terrains humides aux données écoforestières.	20 oct. 2021
La semaine de lecture	
Semaine 10	
Théorique 14. Fichiers	2 nov. 2021
Pratique Dictionnaires, histogrammes; Applications des types de données "list" et "dict" aux données d'écoforestière; Traçage des données.	3 nov. 2021
Semaine 11	
Théorique 15. Classes et objets	9 nov. 2021
Théorique 16. Classes et fonctions.	10 nov. 2021
Semaine 12	
Théorique 17. Classes et méthodes	16 nov. 2021
Pratique Methodes, suites. Ajoute des classes et méthodes à CASFRI.py Surchage d'opérateur et Polymorphisme.	17 nov. 2021
Semaine 13	
Théorique 18. Héritage	23 nov. 2021
Pratique Héritage, avec applications. On va concevoir des classes pour manipuler les données de la composition des essences des arbres d'une peuplement forestière.	24 nov. 2021
Semaine 14	
Théorique Classes pour données écoforestières. Fonctions pour analyses des données écoforestière: essences, hauteur, age d'origine, etc.	30 nov. 2021
Pratique Finaliser les classes Species, Ensemble et Composition pour les données écoforestière;	1 déc. 2021
Semaine 15	
Théorique 19. Les bonus. Certaines autre fonctionnalités de Python: les listes en compréhension, les générateurs, les ensembles, etc.	7 déc. 2021
Fête de l'Université	8 déc. 2021

Note : Veuillez vous référer à la section *Contenu et activités* de votre site de cours pour de plus amples détails.

Évaluation et résultats

Évaluation des apprentissages

Sommatives			
Titre	Date	Mode de travail	Pondération
affectation et des expressions numérique	Dû le 15 sept. 2021 à 00h00	Individuel	5 %

Docstrings et Fleurs	Dû le 22 sept. 2021 à 12h30	Individuel	5 %
réursion	Dû le 29 sept. 2021 à 12h30	Individuel	10 %
la séries de Fibonacci, par itération	Dû le 6 oct. 2021 à 12h30	Individuel	10 %
Jouer avec les mots.	Dû le 13 oct. 2021 à 12h30	Individuel	10 %
lists: reduce	Dû le 20 oct. 2021 à 12h30	Individuel	10 %
wetlandsHist + anagrammes	Dû le 3 nov. 2021 à 12h30	Individuel	10 %
composition des espèces d'arbres des peuplement forestières	Dû le 10 nov. 2021 à 12h30	Individuel	5 %
Classes: Cercle, Temps, et pour essences écoforestière	Dû le 17 nov. 2021 à 15h17	Individuel	10 %
méthodes et opérateurs pour class Pointe (Ch17.py)	Dû le 24 nov. 2021 à 12h30	Individuel	10 %
Classes pour la composition d'un peuplement	Dû le 7 déc. 2021 à 09h00	Individuel	15 %

Notez:

1. Le nombre et calendrier des devoirs sera comme indiqué ci-dessus.
2. Les détails et les critères sont sujets à changement, sauf les 4 premiers.
3. Après l'introduction de Jupyter, tous les devoirs doivent être soumis sous forme de cahiers Jupyter.

Informations détaillées sur les évaluations sommatives

affectation et des expressions numérique

Date de remise : 15 sept. 2021 à 00h00

Normalement, les devoirs devraient soumis avant le cours du labo, une semaine après la date quand ils sont assignés.

Mode de travail : Individuel

Pondération : 5 %

Remise de l'évaluation : [Boîte de dépôt](#)

Directives de l'évaluation :

Part 1: Exercice 2.1 du texte. Le volume d'une sphère de rayon r est $\frac{4}{3} \pi r^3$. Quel est le volume d'une sphère de rayon 5 ? On résoudre cet exercice avec une série de deux affectations. La première préciser le valeur du rayon, le deuxième contient une expression qui faire la bonne calcul, qui est affecté à une variable. Commentaires en suivant le patron de 2.7: **1pt**. Calcul correct: **1pt**.

Part 2: Écrivez une série de quatre affectations qui trouve une racine de la formule quadratique $a + bx + cx^2 = 0$. La première trois affectations préciser les trois coefficients a , b et c . La dernière mettre une solution dans une variable. Commentaires: **1pt**. Calcul correct: **2pt**.

Mettez ses deux parties dans un module `votreNom_Dev1.py`. Vous devez le développer dans l'IDLE.

Docstrings et Fleurs

Date de remise : 22 sept. 2021 à 12h30

Mode de travail : Individuel

Pondération : 5 %

Remise de l'évaluation : [Boîte de dépôt](#)

Directives de l'évaluation :

1. Écrire des docstrings pour les fonctions `polygone` et `arc` comme elles sont définies dans Section 4.7 (Réusinage) de *Pensez-Python*. Mettre vos fonctions documentées dans un fichier `.py` (2pts)

2. Écrivez un ensemble convenablement général de fonctions qui peuvent dessiner des fleurs comme dans la Figure 4.1 de Pensez-Python. Ajoute celles fonctions à la même fichier, avec des appels pour crée des fleurs. (3 pts)

3. Le module Turtle définit les méthodes pour accéder et changer la couleur du stylo d'une tortue. Trouvez-vous ces méthodes (en lissant le module, en cherchant par google, ...), modifier les fonctions afin que l'utilisateur puisse spécifier la couleur d'une fleur. HINT: ajoute une parametre optionnele de chaque fonction avec noir comme le valeur par default. Mettre ces fonctions modifier au deuxième fichier. (5 pts)

réursion

Date de remise : 29 sept. 2021 à 12h30

Mode de travail : Individuel

Pondération : 10 %

Remise de l'évaluation : [Boîte de dépôt](#)

Directives de l'évaluation :

1) définir une fonction recursive booleanne isPair(x) qui determine, pour x un entier non-negatif. Il devrait renvoyer True si x est paire et False se x est impair, sans utiliser les opérateurs // ou %. Gardez-vous ta fonction contre les arguments invalide, comme demonstre pour la fonction factorielle() (voyez les diaspos pour mercredi 22 septembre). On peut faire une synthèse entre les fonctions compte_a_rebours et fibonacci() dans la texte, et la version de la fonction factorial avec la fonction locale imbriquée. **5pts**

2) La fonction koch() était corrigé en classe (voir koch.py sur le site teams) Il marche, mais les dessins qu'il produit sont trop plat. Rends-les colorés! Ajouts des appels à une fonction qui définit la couleur du stylo d'une tortue. Il devrait être simple d'ajoute une mélange de quatre colours, selon la structure de la fonction koch(). Il faut restaurer la couleur initiale avant le retour final . **5pts**

la séries de Fibonacci, par itération


Date de remise : 6 oct. 2021 à 12h30

Mode de travail : Individuel

Pondération : 10 %

Remise de l'évaluation : [Boîte de dépôt](#)

Directives de l'évaluation :

En Section 6-7 du texte, la fonction recursive fibonacci(n), qui calcule n-iem element du [séquence Fibonacci](#) , F_n . Comme on a vu au labo, cette fonction marche très lent pour $n > 30$, si lentement qu'il est inutilisable. Pour ce devoir, écrivez-vous une version itérative. Tu peux utiliser les versions itératives du factoriel() comme un modèle: voyez **W5L1.py**, au site Teams/ici.

Ton fichier devrait inclure ton fonction correct (4pt) avec docstring (2pts) et commentaires interne (1pts) assez pour expliquer son comportement et il faut valider que l'intrans est un entier non négatif (1pts) comme déjà démontré pour la fonction factoriel(). Ton fichier devrait inclure les tests de validation qui seront executer quand je "Run" ton fichier dedans Spyder ou une autre IDLE (2pts).

Ce n'est pas nécessaire que tu utilise des cahiers jupyter pour cet travaux. Par contre, prends cette opportunité pour utiliser Spyder/PyScript pour écrit ton fonction dans son propre fichier dans une nouveau Projet pour toutes les missions à venir.

Jouer avec les mots.

Date de remise :	13 oct. 2021 à 12h30
Mode de travail :	Individuel
Pondération :	10 %
Remise de l'évaluation :	Boîte de dépôt
Directives de l'évaluation :	<ol style="list-style-type: none"> 1. Trouvez les mots dans le fichier scrabble.txt avec trois lettres doubles consécutives. Hint: pour commencer, trouvez les mots avec une pair. Alors, cherchez ceux qui en a deux, cet à dire deux lettres doubles consécutives, et en fin, les triples. On peut utiliser le fait qu'il n y aucune mot avec lettres triples. <i>Ne regardez jamais la solution!</i> Le cas de test devrait utiliser des méthodes du package os pour aller au dossier d'où se trouve le fichier des mots. 5pts 2. Écrire une fonction que prendre trois argument: str1 et str2 qui sont chaînes de caractères, et un drapeau booléen avec un valeur par défaut. Il faut trouver ou compte des characters en str1 qui ne se trouvent pas dans str2; il faut renvoyer le nombre de ces caractères ou une sous-chaîne qui les contient, selon le valeur de drapeau. Hint: on peut commence d'une des fonction de \$9-3 du texte. 5pts

lists: reduce

Date de remise :	20 oct. 2021 à 12h30
Mode de travail :	Individuel
Pondération :	10 %
Remise de l'évaluation :	Boîte de dépôt
Directives de l'évaluation :	<p>Le class list à une méthode remove:</p> <pre>t.remove(x)</pre> <p>qui effacer la premier instance de valeur x sur place. Donc</p> <pre>t=[1,2,3]</pre> <pre>t.remove(2) #il renvoie None</pre> <pre>print(t) # produit [1,3]</pre> <p>Part 1</p> <p>Ecrit une fonction de sort filtre qui reproduit ce comportement, en utilisant la method pop (3pt) . Inclut une docstring et commentaires comme necessaire (1pt). Inclut un cas de test (1pt)</p> <p>Part 2</p> <p>Ecrivez une deuxieme fonction de sort filtre qui renvoyer une nouvelle list d'ou toutes les instances de valeur sont effacés (3pt) Inclut une docstring et commentaires comme necessaire (1pt). Inclut un cas de test (1pt).</p> <p>Part 3</p> <p>Ecrivez une 3eme fonction basé sur la 2eme, d'où le 2eme l'argument peut être une liste, est qui renvoyer une nouvelle list d'où chaque instance de chaque element de cette list est effacer. Il faut marche si cettte liste est vide; si l'argument n'est pas une liste, la fonction devrait avoir la même comportement comme en Part 2. 5pts, divisé entre le code (3pt), le docstrings+commentaires (1pt) et les cas de tests (1pt).</p> <p>Mettre toutes ses fonctions dans une seule fichier.</p>

Date de remise : 3 nov. 2021 à 12h30
Mode de travail : Individuel
Pondération : 10 %
Remise de l'évaluation : [Boîte de dépôt](#)
Directives de l'évaluation :
1) Modifier wetlandsHisto.py pour produire et tracer une histogramme de fréquences des sort de terrains humides.
Correct 3pts; docstring et commentaires 1pt; traceage 1pt
2) 12-10 Exercice 2.1.
Correct 3pts; docstring et commentaires 1pt; cas te dest 1pt

composition des espèces d'arbres des peuplement forestières

Date de remise : 10 nov. 2021 à 12h30
Mode de travail : Individuel
Pondération : 5 %
Remise de l'évaluation : [Boîte de dépôt](#)
Directives de l'évaluation :
Data/CIMP/CASFRISummaryPerFire.txt contient le attributs écoforestières des peuplement d'une zone de la territoire nord-ouest.
Chaque peuplement a 1, 2, 3 ou 4 essences des arbres, dans le collonnes nommé SP1, SP2, SP3 et SP4.
Tous les peuplements n'ont pas quatre espèces, mais il y a toujours au moins SP1. Les collonnes d'ou une essence n'était pas défnit sont codés "XXXX MISS"
Pour ce devoir, on va adapter le module **histogram.py** pour calculer le fréquence de chaque combinaison des essence qui se trouvent dans la base de données. On devrait acceder les quatres collones en utilisant leurs noms symboliques: voir comme il est fait dans **wetlandsHisto.py**
2pts
Compte le nombres des combinaisons des essences uniques, et afficher le 5 et 10 le plus communs.
1pt
Faire une barplot des frequences des essence qui apparait en SP1. **1pt**
docstrings et commentaires **1pt**
Mettez - vous toutes vos logiciels dans un nouveau module casfri.py

Classes: Cercle, Temps, et pour essences écoforestière

Date de remise : 17 nov. 2021 à 15h17
Mode de travail : Individuel
Pondération : 10 %
Remise de l'évaluation : [Boîte de dépôt](#)
Directives de l'évaluation :
5-9-1a Écrivez une définition pour une classe nommée Cercle ayant les attributs centre et rayon, où le centre est un objet Point et le rayon est un nombre. **(1pt)**
15-9-1b Instanciez un objet Cercle qui représente un cercle ayant son centre à (150, 100) et un rayon de 75. **(1pt)**

15-9-1c Écrivez une fonction nommée `point_du_cercle` qui prend un `Cercle` et un `Point` et renvoie `Vrai` si le `Point` se trouve dans le cercle ou sur sa circonférence. **(1pt)**

15-9-1d Écrivez une fonction nommée `rect_du_cercle` qui prend un `Cercle` et un `Rectangle` et renvoie `Vrai` si le rectangle se trouve entièrement dans le cercle ou sur sa circonférence. **(1pt)**

15-9-2 a Écrivez une fonction appelée `dessine_rect` qui prend un objet `Tortue` et un `Rectangle` et utilise la `Tortue` pour dessiner le `Rectangle`. Voir le chapitre pour des exemples utilisant des objets `Tortue`. **(1pt)**

16-7-1 a Écrivez une fonction appelée `mul_time` qui prend un objet `Temps` et un nombre et retourne un nouvel objet `Temps`, qui contient le produit entre le `Temps` d'origine et le nombre. **(1pt)**

Pensez-vous: comment définir des classes pour des essences et pour la composition d'une peuplement comme dans le fichiers `CASFRI`. Préciser leurs attribues et proposé des fonctions /méthodes nécessaires. ou utiles. Brièvement, dans un fichier texte ou google doc **(4pt)**

méthodes et opérateurs pour class `Pointe` (Ch17.py)

Date de remise : 24 nov. 2021 à 12h30

Mode de travail : Individuel

Pondération : 10 %

Remise de l'évaluation : [Boîte de dépôt](#)

Directives de l'évaluation :

ajoute un méthode pour calcule le distance Euclidean entre 2 points:e.g `p1.distEuclid(p2)`
(docstring+comments = 1pt, correct = 1pt, cas de test = 1pt).

surcharge l'opérateur `*` pour multiplié une `Pointe` par un `int`, `float` ou `Pointe`. Pour multiplier `p1 * p2`, calculez le "vector dot product". (docstring+comments = 1pt, correct = 3pt, cas de test = 1pt). Bonus: adapter `__rmult__` 2pt.

ajoute und méthode `p.rotate(angle)`: faire pivoter un point autour de l'origine (0,0) de l'angle donné (docstring + comments 1, correct = 2, cas de test = 1pt).

En donnat que ton fichier soit module, mettre les cas de tests dans une déclaration conditionnelle: `if __name__ == "__main__":`

Classes pour la composition d'un peuplement

Date de remise : 7 déc. 2021 à 09h00

Mode de travail : Individuel

Pondération : 15 %

Remise de l'évaluation : [Boîte de dépôt](#)

Directives de l'évaluation :

Part 1: Le fichier `CASFRI.py` definit deux classes `Species` et `Ensemble`.

Document chaque classe avec docstrings et commentaires (1pt + 1pt) et instancié un `Ensemble` qui inclut tout les essences que ses trouvent dans `CASFRIsummaryerFire.txt` (1pt). **(3pts)**

Part 2: Définir et pleinement documente une nouvelle classe `Composition`, adapté pour decrire et manipuler la composition des essences dans un peuplement. Indice: liste des tuples d'une instance de class `Species` et un entier. Ajoute les méthodes `__str__`, `__init__`, et `ajoute_element(self, Species, pct)`, **(6 pts)**.

Part 3: Utilise les methodes de cette nouvelle classe pour trouver et sommarise toutes les combinaisons des essences dans `NWT` (Devoir 8), mais aussi rapporter la distrubution des pourcentages feuillue. **(3 pts)**

Faites un cahier Jupyter dont tu pourrais être fier.

Barème de conversion

Cote	% minimum	% maximum
A+	89,5	100
A	86,5	89,49
A-	83,5	86,49
B+	80,5	83,49
B	77,5	80,49
B-	74,5	77,49

Cote	% minimum	% maximum
C+	71,5	74,49
C	68,5	71,49
C-	64,5	68,49
D+	60,5	64,49
D	54,5	60,49
E	0	54,49

Règles disciplinaires contre la tricherie et le plagiat

Tout étudiant(e) qui commet une infraction relative aux études, au sens du Règlement disciplinaire à l'intention des étudiants de l'Université Laval, dans le cadre du présent cours, notamment en ce que constitue du plagiat, est passible des sanctions qui sont prévues par ce Règlement. Il est très important que chaque étudiant(e) prenne connaissance des articles 22 à 32 dudit Règlement, à : <http://ulaval.ca/reglement-disciplinaire>

Tout étudiant(e) est tenu, en réalisant tout travail écrit requis dans un cours, de respecter les règles relatives à la protection du droit d'auteur et à la prévention du plagiat dans ses travaux formatés soumis à l'évaluation. Constituent notamment du plagiat les faits de :

- i. copier textuellement un ou plusieurs passages provenant d'un ouvrage sur support de papier ou électronique sans mettre ces passages entre guillemets ni en hors-texte et sans en mentionner la source;
- ii. résumer l'idée originale d'un auteur(e) en l'exprimant dans ses propres mots (paraphraser) sans en mentionner la source;
- iii. traduire partiellement ou totalement un texte sans en mentionner la provenance;
- iv. remettre un travail copié partiellement ou totalement d'un autre étudiant(e) (avec ou sans son accord);
- v. remettre un travail téléchargé partiellement ou totalement d'un site d'achat ou d'échange de travaux scolaires.

[Sources: En application de l'article 161 du Règlement des études de l'Université Laval, https://www.ulaval.ca/fileadmin/Secrtaire_general/Reglements/Reglement_des_etudes.pdf. Commission de l'Éthique de la science et de la technologie, *La tricherie dans les évaluations et les travaux à l'université: l'éthique à la rescousse* (rédaction: Denis Boucher), Québec, 15 mai 2009; texte adapté ici le 16 juillet 2009.]

Correction linguistique, retard et présentation des travaux

Évaluation de la qualité du français

La Faculté de foresterie, de géographie et de géomatique se réfère à la [Politique sur l'usage du français à l'Université Laval](#) ainsi qu'aux [dispositions relatives à son application](#).

De plus, la Faculté recommande aux enseignants d'attribuer jusqu'à concurrence de 15 % de la note totale de tout examen, rapport, travail long ou tout autre document évalué, à la correction orthographique et grammaticale.

Une plus grande tolérance est accordée lors de la correction des travaux et des examens des étudiants non francophones.

Au besoin, profitez des services d'amélioration de la qualité du français à votre disposition sur le campus :

- [Ateliers gratuits d'aide à la rédaction](#) offerts par la Bibliothèque
- [Cours de perfectionnement en français de 1 à 3 crédits](#) offerts en classe par l'École des langues
- [Cours de perfectionnement en français de 1 à 3 crédits](#) offerts à distance par l'École des langues

Retard et présentation des travaux

Aucun retard injustifié à la remise des travaux ne sera toléré.

Utilisation d'appareils électroniques pendant une séance d'évaluation

Le seul appareil électronique toléré pendant une séance d'évaluation est la calculatrice.

Les calculatrices autorisées durant les séances d'examen pour tous les cours offerts par la Faculté de foresterie, de géographie et de géomatique sont les suivantes :

- Hewlett Packard HP 20S, HP 30S, HP 32S2, HP 33S, HP 35S
- Texas Instrument TI-30Xa, TI-30XIIB, TI-30XIIS, TI-36X (plus fabriqué),
- BA35
- Sharp EL-531**, EL-535-W535, EL-546**, EL-510 R, EL 516*, EL-520**
- Casio FX-260, FX-300 MS, FX-350 MS, FX-300W Plus, FX-991MS, FX-991ES (plus fabriqué), FX-991W*, FX-991ES Plus C*

* Modèles qui ne seront plus autorisés dès 2016.

** Calculatrices Sharp: sans considération pour les lettres qui suivent le numéro.

Absence aux examens

Un étudiant absent à un examen ou à toute autre séance d'évaluation obtient automatiquement la note zéro à moins qu'il ait des motifs sérieux justifiant son absence.

Les seuls motifs acceptables pour s'absenter à un examen et avoir droit à un examen de reprise sont les suivants :

- **Convocation par une cour de justice** durant la plage horaire prévue pour l'examen avec preuve de convocation.
- **Maladie durant la plage horaire prévue pour l'examen avec un billet de médecin** précis incluant les dates d'invalidité et les coordonnées du médecin.
- **Mortalité d'un proche** avec preuve de décès et lettre d'une tierce personne attestant du lien de parenté ou autre lien entre l'étudiant et la personne décédée.

Les pièces justificatives doivent être des originaux et doivent être présentées à l'enseignant, au directeur de programme ou au secrétariat des études (1250 pavillon Abitibi-Price) le plus rapidement possible.

Aucune justification d'absence reliée à des événements sportifs (sauf pour les athlètes du Rouge et Or, sur approbation préalable de la direction de programmes) ou reliée à un emploi, à un conflit d'horaire avec d'autres cours ou examens, à des horaires de voyage conflictuels (billets d'avion déjà achetés, par exemple) ou à des motifs religieux quelconques n'est acceptable.

Les conflits d'horaire doivent être résolus au tout début de la session, avant la fin de la période de modification du choix de cours, par l'étudiant lui-même. Un étudiant inscrit au cours après cette date est réputé ne pas avoir de conflit d'horaire et pourra se présenter à tous ses examens.

L'étudiant dont l'absence est dûment justifiée a l'obligation de se rendre disponible pour un examen de reprise à la date fixée par l'enseignant sans quoi il obtiendra la note zéro pour cet examen.

Étudiants ayant une situation de handicap liée à une limitation fonctionnelle

Afin de bénéficier de mesures d'accommodement pour les cours ou les examens, un rendez-vous avec une conseillère ou un conseiller du Centre d'aide aux étudiants travaillant en **Accueil et soutien aux étudiants en situation de handicap (ACSESH)** est nécessaire. Pour ce faire, les étudiants présentant une situation de handicap liée à une limitation fonctionnelle permanente doivent visiter le site monPortail.ulaval.ca/accommodement et prendre un rendez-vous, le plus tôt possible. Au cours de la semaine qui suit l'autorisation des mesures, l'activation des mesures doit être effectuée dans monPortail.ulaval.ca/accommodement pour assurer leur mise en place.

Les étudiants ayant déjà obtenu des mesures d'accommodements scolaires doivent procéder à l'activation de leurs mesures pour les cours et/ou les examens dans monPortail.ulaval.ca/accommodement afin que celles-ci puissent être mises en place. Il est à noter que l'activation doit s'effectuer au cours des deux premières semaines de cours.

Matériel didactique

Matériel obligatoire

Pensez en Python

URL : [Pensez en Python](#)

Auteur : Allen B. Downey

Date d'accès : 30 août 2020

La traduction Française du première edition du livre "How to think like a Computing Scientist in Python"

Think Python: How to Think Like a Computer Scientist 2nd Edition. Version 2.2.23

URL : [Think Python: How to Think Like a Computer Scientist 2nd Edition. Version 2.2.23](#)

Auteur : Allen B. Downey

Date d'accès : 30 août 2020

This English language version is updated from the original Cambridge University Press publication. It may not identical to the French Language version in some details, notably the Exercises.

Python Modules and Packages – An Introduction

URL : [Python Modules and Packages – An Introduction](#)

Date d'accès : 30 août 2020

Explorez plus profondément le concept de "packages", pour profiter du travail des autres.

Choosing Python or R for Data Analysis? An Infographic

URL : [Choosing Python or R for Data Analysis? An Infographic](#)

Date d'accès : 30 août 2020

Markdown Cheat Sheet

URL : [Markdown Cheat Sheet](#)

Date d'accès : 30 août 2020

Avec liens aux documentation plus formel et complète.

Matériel complémentaire

Introduction au langage de programmation Python 2/3

URL : [Introduction au langage de programmation Python 2/3](#)

Auteur : Fabrice Sincère

Date d'accès : 14 septembre 2017

Un cours en française: notez en particulière Chapitre 11- Débogage. Il va vous introduire l'environnement IDLEMacOS: il faut installer MacPython: <https://www.python.org/>

Python as a first language: pre-conference workshop

URL : [Python as a first language: pre-conference workshop](#)

Auteur : Allen B Downey

Date d'accès : 11 septembre 2017

Dit l'auteur: "Python is one of the best languages for beginning programmers. It is syntactically and semantically clean, which spares students from the distractions that plague other languages. It provides lists and dictionaries (hashtables) as built-in types, so students can focus more on algorithms and less on the implementation of data structures. It is dynamically typed, which allows students to explore alternative designs more easily ..."

Pÿthon for Software Design

URL : [Pÿthon for Software Design](#)

Auteur : Allen B Downer
Date d'accès : 11 septembre 2017
La première édition, en anglais.

Learning with Python 3 (RLE)

URL : [Learning with Python 3 \(RLE\)](#)

Auteur : Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers
Date d'accès : 11 septembre 2017
Une version révisé. Le monde du tortue est introduit en chapitre 3.

Full Stack Python: Markdown

URL : [Full Stack Python: Markdown](#)

Date d'accès : 30 août 2020
Une introduction brève, avec liens aux liens utiles.

Bibliographie

Bibliographie

Downey AB. Python as a first language: pre-conference workshop. Journal of Computing Sciences in Colleges. 2007 Jun 1;22(6):3-4.

Downey AB. Python for software design: how to think like a computer scientist. Cambridge University Press; 2009 Mar 9.

Downey AB. Think Python. O'Reilly Media, Inc.; 2012 Aug 13.

Annexes

Il n'y a aucune annexes à ce temps.