

PLAN DE COURS

FOR-4048 : Compétences de programmation pour les sciences forestières

NRC 84914 | Automne 2020

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 31 août 2020 au 11 déc. 2020
mercredi	12h30 à 14h50	Du 31 août 2020 au 11 déc. 2020
dimanche	09h00 à 10h20	Du 31 août 2020 au 11 déc. 2020

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=121863>

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 formatives	7
Exercice 2, Section 2-10 de Pensez en Python	7
Docstrings et Fleurs	7
Deux études en récursion	8
Fonctions interactif et numérique	8
Lettres doubles	9
Listes et import	9
Anagrams in a more complete list of English words.	9
Classes et Fonctions (Ch 16)	10
Classes et méthodes (Ch 17)	11
Héritage (Ch 18)	11
A python class emulating R's vectorial data (Option A)	12
Term Project (Option B)	12
Barème de conversion	12
Règles disciplinaires contre la tricherie et le plagiat	12
Correction linguistique, retard et présentation des travaux	13
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	14
Matériel didactique	14
Matériel obligatoire	14
Matériel complémentaire	15
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 le 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 forsterie ou biologie tel que le traitement des données écoforestière. Donc les élèves peut choisir à faire un projet de terme qui va compter pour 30% de la note finale. Ce projet et fortement racommandé aux étudiant(e)s gradué(e)s

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 1 septembre 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.	2 sept. 2020
Semaine 2	
Théorique 1. Le chemin du programme Notez que les nombres initiaux, comme 1 ici, se réfèrent aux chapitres de Pensez en Python.	8 sept. 2020
Théorique et pratique Théorique: 2. Variables, expressions et instructions. Pratique: Installer et démontrer Python 3.8 et IDLE, un environnement de développement intégré simple.	9 sept. 2020
Semaine 3	
Théorique 3. Fonctions	15 sept. 2020
Pratique 4. Étude de cas : conception d'une interface. On introduit le monde des tortues.	16 sept. 2020
Semaine 4	
Théorique 5. Structures conditionnelles et récursion	22 sept. 2020
Théorique et Pratique Théorique: 6. Fonctions productives Pratique: Débogage (voyez Annexe A de Pensez en Python, et la documentation de IDLE)	23 sept. 2020
Semaine 5	
Théorique 7. Itération	29 sept. 2020
Méthodes de recherche reproductible Intégration des données, du code et de la documentation avec Markdown et le Projet Jupyter (https://jupyter.org/).	30 sept. 2020
Semaine 6	
Théorique 8. Chaînes de caractères	6 oct. 2020
Pratique 9. Étude de cas : jouer avec les mots	7 oct. 2020
Semaine 7	
Théorique 10. Listes	13 oct. 2020
Pratique Modules et packages en Python. Modules intégré, démontrer avec «array» Modules et packages externe: un univers en expansion.	14 oct. 2020
Semaine 8	
Théorique 11. Dictionnaires	20 oct. 2020
Pratique 12. Tuples 13. Étude de cas : le choix des structures de données	21 oct. 2020

La semaine de lecture	
Semaine 10	
Théorique 14. Fichiers	3 nov. 2020
Pratique Un application du monde réel: traitement des données des cartes écoforestières.	4 nov. 2020
Semaine 11	
Théorique 15. Classes et objets	10 nov. 2020
Théorique 16. Classes et fonctions.	11 nov. 2020
Semaine 12	
Théorique 17. Classes et méthodes	17 nov. 2020
Pratique Methodes, suites. Exercices et matériel complémentaire.Surcharge d'opérateur et Polymorphisme.	18 nov. 2020
Semaine 13	
Théorique 18. Héritage	24 nov. 2020
Pratique Héritage, avec applications.	25 nov. 2020
Semaine 14	
Théorique Python et R, comparés: les langues, utilisateurs et applications	1 déc. 2020
Pratique Les données vectorielles et arithmétique vectorisée sont fournis par R. Ajoutons-les à Python, en appliquant ce que nous avons appris sur la programmation orientée objet en Python.	2 déc. 2020
Semaine 15	
Théorique 19. Les bonus. Certaines autre fonctionnalités de Python: les listes en compréhension, les générateurs, les ensembles, etc.	8 déc. 2020
Pratique (suite de la semaine 14) Les méthodes vectorielles en Python: notre approche vs package NumPy.Présentation des projets à terme.	9 déc. 2020

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

Formatives		
Titre	Date	Mode de travail
Exercice 2, Section 2-10 de Pensez en Python	Dû le 16 sept. 2020 à 12h30	Individuel
Docstrings et Fleurs	Dû le 23 sept. 2020 à 12h30	Individuel

Deux études en récursion	Dû le 30 sept. 2020 à 12h30	Individuel
Fonctions interactif et numérique	Dû le 7 oct. 2020 à 12h30	Individuel
Lettres doubles	Dû le 14 oct. 2020 à 12h30	Individuel
Listes et import	Dû le 21 oct. 2020 à 12h30	Individuel
Anagrams in a more complete list of English words.	Dû le 4 nov. 2020 à 12h30	Individuel
Classes et Fonctions (Ch 16)	Dû le 18 nov. 2020 à 12h30	Individuel
Classes et méthodes (Ch 17)	Dû le 25 nov. 2020 à 12h30	Individuel
Héritage (Ch 18)	Dû le 2 déc. 2020 à 12h30	Individuel
A python class emulating R's vectorial data (Option A)	Dû le 11 déc. 2020 à 17h00	Individuel
Term Project (Option B)	Dû le 11 déc. 2020 à 17h00	Individuel

Notez: le nombre, ponderation et calendrier des devoirs sera comme indiqué ci-dessus.

Les détails et les critères sont sujets à changement, sauf les 4 premiers.

Le projet de term (Option B) est optionel. Donc, on doit choisir une de Option A ou Option B.

Informations détaillées sur les évaluations formatives

Exercice 2, Section 2-10 de Pensez en Python

Date de remise : 16 sept. 2020 à 12h30

Mode de travail : Individuel

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

Directives de l'évaluation :

S'il vous plait, mettre au boite de dépôt une page de texte qui contient le code python vous avez utilise pour faire chaque calculation.

Pour points 2 et 3, les code doit include des commentaires bréf mais informtative comment intrdoduit en section 2-7.

1. 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 ? (1pt)
2. Supposons que le prix d'un livre est de 24,95 €, mais les librairies obtiennent un rabais de 40 %. Les frais d'expédition s'élèvent à 3 € pour le premier exemplaire et à 75 centimes pour chaque exemplaire supplémentaire. Quel est le coût total de gros pour 60 copies ? (1pt pour le bon réponse, 1pt pour le commentaire)
3. Si je pars de chez moi à 6 h 52 et que je cours 1,5 kilomètre à un rythme facile de 5 minutes et 10 secondes par km, puis 5 km au tempo plus soutenu de 4 minutes et 20 secondes par kilomètre et encore 1,5 km au rythme facile du début, à quelle heure serai-je de retour à la maison pour le petit déjeuner ? (1pt pour le bon réponse, 1pt pour le commentaire)

Docstrings et Fleurs

Date de remise : 23 sept. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
Les docstrings sont tous informative et bref.	3
Les fonctions qui dessin le fleur sont correct et complète	5
Les fonctions sont bien documenté avec commentaires et docstrings	2

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

Directives de l'évaluation :

1. Écrire des docstrings pour les fonctions polygon arc et circle comme elles sont définies dans Section 4.7 (Réusinage) de Pensez-Python.

2. Écrivez un ensemble convenablement général de fonctions qui peuvent dessiner des fleurs comme dans la Figure 4.1 de Pensez-Python. Soumettre cela dans un fichier .py qui on peut accéder par le IDE.

Deux études en récursion

Date de remise : 30 sept. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
Palindrome	5
Power	5

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

Directives de l'évaluation :

Exercice 3 (palindrome; ne lire pas la solution!) 5pts.

```
x="bonjour", x=x[1:-1], print(len(x))
```

Exercice 4 (is_power()) 5pts

Pense toujours:

c'est quoi les cas de base?

comment peut-t-on y arriver par moyen de la récursion appliqué aux paramètres?

Pour chaque exercice, faire un fichier .py qui contient le code bien documenter, et deux ou trois cas de test pour illustrer et vérifier qu'ils sont correct.

Fonctions interactif et numérique

Date de remise : 7 oct. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
Ex1	5
Ex2	5

Remise de l'évaluation :

[Boîte de dépôt](#)

Directives de l'évaluation :

Write, test and document a function to compute a decimal approximation of pi, as explained in exercise 7.5 of the text (5pt)

Write, test and document a function to approximate the cube root of a positive argument. Apply the Newton method, presented in section 7.5 of the text for the calculation of a square root. (5pt).

Note: don't neglect your docstrings and other internal documentation to your functions.

Lettres doubles

Date de remise :

14 oct. 2020 à 12h30

Mode de travail :

Individuel

Remise de l'évaluation :

[Boîte de dépôt](#)

Directives de l'évaluation :

Write a program that processes a list of words (as in the file words.txt, provided) and detects and reports all the words containing a sequence of three (3) consecutive twice-repeated letters, i.e. a substring of the form "aabbcc".

From Downey, exercise 9.7: I'll give you a couple of words that almost qualify, but don't. For example, the word committee, c-o-m-m-i-t-t-e-e. It would be great except for the 'i' that sneaks in there. Or Mississippi: M-i-s-s-i-s-s-i-p-p-i. If you could take out those i's it would work. But there is a word that has three consecutive pairs of letters and to the best of my knowledge this may be the only word. ... What is the word? [and are there others?]

To write this you need to read and process each word in the file. So the main challenge is to write a function that detects the pattern of interest. Hint: be careful of edge cases like "aaxbbccdd".


Trouvez les mots (dans words.txt) avec trois lettres doubles consécutives.

Hint: pour commencer, trouvez les mots avec une paire. Alors, cherchez ceux qui en ont deux, c'est à dire deux lettres doubles consécutives.

On peut utiliser le fait qu'il n'y a aucun mot avec lettres triples;

Ne regardez jamais la solution publiée au site web du texte Penez-Python.

Fichiers à consulter :

 [Word list for the assignment](#). (1,43 Mo, déposé le 31 août 2020)

Listes et import

Date de remise :

21 oct. 2020 à 12h30

Mode de travail :

Individuel

Remise de l'évaluation :

[Boîte de dépôt](#)

Directives de l'évaluation :

À préciser plus tard

Anagrams in a more complete list of English words.

Date de remise : 4 nov. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
Make a more complete English word list	5
Find all the anagrams, print the largest sets first.	5

Remise de l'évaluation :

[Boîte de dépôt](#)

Directives de l'évaluation :

5pts) Create a comprehensive English word list, containing no hyphenated words (e.g. Abdul-Aziz), no abbreviations (e.g. A.F.A.M.), no proper nouns *(e.g. Adam), or punctuation (e.g. accretion's). Start from <https://raw.githubusercontent.com/dwyl/english-words/master/words.txt>, and show the code that does the tidying, saving the output to a file.

5pts) Find all the anagrams!

1. Write a program that reads a word list (use the results of the previous exercise) and prints all the sets of words that are anagrams.

Here is an example of what the output might look like:

```
['deltas', 'desalt', 'lasted', 'salted', 'slated', 'staled']['retainers', 'ternaries']['generating', 'greatening'
```

Hint: you might want to build a dictionary that maps from a set of letters to a list of words that can be spelled with those letters. The question is, how can you represent the set of letters in a way that can be used as a key. Print the largest set of anagrams first, followed by the second largest set, and so on.

Classes et Fonctions (Ch 16)

Date de remise : 18 nov. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
mul_time	2
temps_par_km	3
Nelder Plot module basic specifications	5

Remise de l'évaluation :

[Boîte de dépôt](#)

Directives de l'évaluation :

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.

Write a function called `mul_time` which takes as arguments an object of class `Temps` and an object of numeric type, and returns a new object of class `Temps` which is the product of the arguments `Temps` and number.

(2pts)

Ensuite, utilisez `mul_time` pour écrire une fonction qui prend un objet `Temps` qui représente le temps de l'arrivée dans une course, et un nombre qui représente la distance, et retourne un objet `Temps` qui représente le rythme moyen (temps par kilomètre).

Then, use `mul_time` to write a function that takes a `Time` object that represents the time of arrival in a race, and a number that represents the distance, and returns a `Time` object that represents the average rate (time per kilometer).

(3 pts)

State the requirements for a module to help design and implement Nelder plots, in terms of base functionality and input requirements. Do some research on how such initial specifications should be written. **(5pts)**

Classes et méthodes (Ch 17)

Date de remise : 25 nov. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
Point + Point	3
Point + tuple	3
Documentation	2
Cas de test	2

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

Directives de l'évaluation :

À titre d'exercice, écrivez une méthode `add` pour des objets `Point` qui fonctionne sur un objet de type

`Point` ou un tuple :

- si le second opérande est un `Point`, la méthode doit retourner un nouveau `Point` dont l'abscisse `x` est la somme des abscisses `x` des opérandes, et dont l'ordonnée `y` est la somme des ordonnées `y`; **(3pts)**
- si le second opérande est un tuple, la méthode doit additionner le premier élément du tuple à l'abscisse `x` et le second élément à l'ordonnée `y`, et renvoyer un nouveau `Point` dont les coordonnées représentent le résultat; **(3pts)**

Documentation; **(2pts)**

Test cases (pour vérifier qu'ils marchent) **(2 pts)**

Héritage (Ch 18)

Date de remise : 2 déc. 2020 à 12h30

Mode de travail : Individuel

Critères de correction :

Critère	Notation
<code>markov.py</code>	5
Exercice 2	5

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

Directives de l'évaluation :

L'exercice de la section 18-10: Encapsulation de données (5 pts; ne regarde jamais la solution)

Exercice 2: Écrivez une méthode de Paquet appelée `distribue_mains` qui prend deux paramètres, le nombre de mains à distribuer et le nombre de cartes par main. Elle doit créer le nombre voulu d'objets `Main`, distribuer le nombre approprié de cartes par main et renvoyer une liste de `Mains` (5 pts).

A python class emulating R's vectorial data (Option A)

Date de remise : 11 déc. 2020 à 17h00

Mode de travail : Individuel

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

Pour ceux que ne fait pas un projet de terme.

Directives de l'évaluation :

pas encore finalisé

Term Project (Option B)

Date de remise : 11 déc. 2020 à 17h00

Mode de travail : Individuel

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

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 formateurs 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/Secretaire_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 de la première édition 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 be 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 ..."

Python for Software Design

URL : [Python for Software Design](#)

Auteur : Allen B Downey

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.