

Programme de la première Année

Filière MP/ PC/T

Ce programme a été élaboré, sur la base du contenu et non pas sur la répartition des séances.

SEMESTRE I

Titre des chapitres & contenu		Charge Horaire Approximative
<p>PARTIE I (14 h)</p> <p>Programmation Python</p>	<p>Chapitre I: Rappel des notions algorithmiques</p> <ol style="list-style-type: none"> 1. Notion d'algorithme. 2. Opérations élémentaires, Structures conditionnelles, Structures itératives, tableau à une dimension et tableau à deux dimensions. 3. Notion de coût d'un algorithme et classes de complexité. <p><u>Travaux Dirigés</u></p> <p>Applications introduisant la notion de coût d'un algorithme.</p>	4 H
	<p>Chapitre II: Environnement de développement Python 3</p> <ol style="list-style-type: none"> 1. Historique et raisons du choix du langage. 2. Mode interactif, mode Script, Aide en ligne. 3. Types élémentaires (classes int, str, float, bool, complex). 4. Opérations élémentaires sur les différents types élémentaires (approche classique/ approche orientée objet). 5. Notion de bibliothèque et import des packages prédéfinis (fonctions de bibliothèque). 6. Instructions élémentaires. 7. Structures conditionnelles. 8. Structures itératives. 9. Présentation des types composés : les types mutables (listes, dictionnaires, ensembles) et non mutables (tuples et 	12 H

	<p>chaînes).</p> <p><u>Travaux Dirigés</u></p> <ul style="list-style-type: none"> ▪ Instructions de calculs (opérations arithmétiques, calculs avec import de fonctions prédéfinis,...) en mode interactif (mode console). ▪ Ecrire et exécuter des programmes (en utilisant des instructions élémentaires, des structures conditionnelles et des structures itératives) en mode script. ▪ Manipulation des types mutables et non mutables. 	
<p><u>Remarque</u></p> <p>Les parties algorithmiques et programmation seront enseignées en parallèle. A chaque notion algorithmique on associera son équivalent Python.</p> <p><u>Acquis</u></p> <ul style="list-style-type: none"> – Maîtriser l'environnement Python. – Connaître l'allocation dynamique de la mémoire. – Savoir différencier entre les types mutables et non mutables en important le module copy. – Savoir calculer le coût d'un algorithme et différencier les classes de complexité (linéaire, quadratique, logarithmique, quasi-linéaire, exponentielle...). 		
<p>PARTIE II (10H)</p> <p>Programmation modulaire</p>	<p>Chapitre III: Les sous programmes en algorithmique</p> <ol style="list-style-type: none"> 1. Fonctions et procédures. 2. Passage de paramètres (Entrée, Sortie, E/S). 3. Variables locales et Variables globales. 	4H
	<p>Chapitre IV : Les fonctions en Python</p> <ol style="list-style-type: none"> 1. Définition de fonctions par : <ul style="list-style-type: none"> ▪ <i>def</i> ▪ <i>lambda</i> 2. Variables locales et variables globales. 3. Notion de fonction locale. 4. Réutilisation de modules (import de fonctions). 5. Gestion des erreurs (bloc Try... Except). 6. Documentation des fonctions. 7. Coût de fonctions et classes de complexité. <p><u>Travaux Dirigés(Programmation Python)</u></p> <p>Exercices d'Arithmétiques, nombres premiers, nombres parfaits, nombres amis, calcul de PGCD, PPCM, multiplication égyptienne,</p>	8H

Acquis

- Savoir écrire un programme itératif.
- Maîtriser la programmation modulaire.

Total Semestre I

(28H)

SEMESTRE II

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE II (suite) (4h) Programmation Modulaire	Chapitre V : La Récursivité 1. Définition de la récursivité. 2. Avantages et inconvénients de la récursivité (mémoire et temps de calcul). <u>Travaux Dirigés</u> Suites récurrentes, exponentiation rapide, algorithme d'Euclide, fractales...	4H
PARTIE III (8h) Les itérables et algorithmes de tri	Chapitre VI : Manipulation des itérables en Python 1. Les méthodes de recherche (liste, chaîne de caractères ...). 2. Les Algorithmes de tri (sélection, à bulles, insertion, fusion) <u>Travaux Dirigés(Programmation Python)</u> Applications avec étude des coûts.	8H

Acquis

- Savoir écrire un programme récursif et connaître la notion de critère d'arrêt et d'appel récursif
- Savoir manipuler les conteneurs itérables sous Python
- Maîtriser les algorithmes de recherche et de tri.
- Calcul de coût des algorithmes itératifs

PARTIE IV (16 H) Manipulation des fichiers et Simulation Numérique	Chapitre VII : Manipulation des fichiers Python 1. Ouverture et fermeture d'un fichier (texte/binaire: module pickle). 2. Lecture et écriture depuis/dans un fichier texte.	4H
	Chapitre VIII : Système de numération 1. Représentation des nombres (naturels, relatifs et réels) dans une base b (binaire, octale, Hexadécimale ...). 2. Transcodage : Passage d'une base à l'autre. 3. Opérations élémentaires sur les nombres binaires. 4. Limites de la représentation (erreurs d'arrondi et de débordement de mémoire). <u>Travaux Dirigés</u> <ul style="list-style-type: none"> ▪ Représentation des nombres dans une base b. ▪ Passage d'une base à l'autre. ▪ Opérations élémentaires. 	4H
	Chapitre IX : Simulation numérique – Présentation de la bibliothèque numpy – Utiliser les classes array et ndarray de la bibliothèque numpy et l'ensemble de fonctions d'algèbre linéaire. – Simulation d'expériences et traçage de courbes (matplotlib). – Résolution d'équations différentielles du premier ordre (méthode d'Euler) (scipy). – Méthodes de recherche du zéro d'une fonction (Méthode Newton, Méthode dichotomique, ...). – Méthodes d'intégrations (Méthode des rectangles, Méthode des trapèzes,...) (scipy). .	8H
<u>Acquis :</u> Au bout de cette partie l'étudiant doit :		

- Savoir manipuler les fichiers avec Python : récupérer et/ou stocker des données depuis des fichiers pour simulation numérique
- Savoir représenter un nombre dans une base b et convertir d'une base à l'autre, connaître les limites de la représentation des données sur machine.
- Savoir manipuler les bibliothèques numpy, scipy et matplotlib.
- Savoir résoudre des problèmes scientifiques avec Python.

Total Semestre II

(28 H)

Programme de la deuxième Année

Filière MP/ PC/PT

SEMESTRE I

Titre des chapitres & contenu		<i>Charge Horaire Approximative</i>
PARTIE I (8h) Rappel et Structures de Données Avancées en Python	Chapitre 0 : Rappel général – Programmation Python. – Syntaxe, types de bases et types composés. <u><i>Travaux Dirigés</i></u> Exercices extraits de concours.	6h
	Chapitre I: Structures de données avancées en python 1. Notion de Pile. 2. Notion de File. <u><i>Travaux Dirigés</i></u> <ul style="list-style-type: none"> ▪ Modélisation avec les listes sous Python. ▪ Notation post fixée, expression bien parenthésée. 	4h
<u><i>Acquis</i></u> Savoir modéliser les piles et les files sous python au moyen de listes.		
PARTIE II (8h) Programmation Orientée Objet avec Python	Chapitre II: Programmation orientée objet avec Python 1. Objet 2. Classe (Attributs, méthodes, constructeur/ destructeur ...). 3. Encapsulation. 4. Héritage simple. <u><i>Travaux dirigés:</i></u> <ul style="list-style-type: none"> ▪ Représentation des formes géométriques (droite, carré, triangle, etc.)(Module turtle) 	6h

	<ul style="list-style-type: none"> ▪ Jeu de carte. 	
<p><u>Acquis</u></p> <ul style="list-style-type: none"> - Savoir créer et instancier des classes. - Savoir programmer en utilisant la notion d'objet (attributs et méthodes). - Connaître la notion d'héritage. 		
<p style="text-align: center;">PARTIE II (12h)</p> <p style="text-align: center;">Les bases de données</p>	<p>Chapitre III: Introduction aux bases de données</p> <ol style="list-style-type: none"> 1. Limite des fichiers 2. Notions de bases de données (définition) 3. Les systèmes de gestion de bases de données <ul style="list-style-type: none"> - Définition. - Objectifs. 4. Architecture des SGBD <ul style="list-style-type: none"> - Architecture client/ serveur - Architecture trois-tiers 5. Modèle de données <ul style="list-style-type: none"> - Le modèle relationnel: l'algèbre relationnelle - La relation : <ul style="list-style-type: none"> ✓ Schéma de la relation (Attributs et contraintes d'intégrité). ✓ Opérations unaires: projection, sélection. ✓ Opérations binaires : intersection; union; jointure; différence. 	6H
	<p>Chapitre IV: Passage à la pratique (module SQLite3 de Python)</p> <ol style="list-style-type: none"> 1. Création d'une base. 2. Création de tables (schéma de la table, alimentation). 3. Langage SQL: manipulation des données <ul style="list-style-type: none"> - Projection - Sélection - Opérateurs SQL <ul style="list-style-type: none"> ✓ Clauses(<i>fonctions d'agrégations, having, order by, group by, etc.</i>) ✓ Union, intersection, différence, jointure <p><u>Travaux Dirigés</u></p>	6H

	Requêtes simples et imbriquées.	
<p><u>Acquis</u></p> <ul style="list-style-type: none"> - Assimiler les notions de base de données et les systèmes de gestion de base de données. - Savoir créer et alimenter une base de données avec Python. - Savoir écrire des requêtes en langage algébrique et langage SQL. 		
<p>Total Semestre I (28 H)</p>		
<p>SEMESTRE II</p>		
Titre des chapitres & contenu		<i>Charge Horaire Approximative</i>
<p>PARTIE III (22h)</p> <p>Simulation numérique</p>	<p>Thèmes à aborder:</p> <ol style="list-style-type: none"> 1. Algèbre linéaire: Calcul matriciel, méthode de pivot de Gauss, Décomposition LU 2. Cryptographie: Méthode Hill, code de César, code de Vigenère, méthode RSA, codage de Hamming ... 3. Traitement d'image: Compression, détection de contour Convolution, Dilatation 4. Problèmes d'interpolations: Interpolation de Lagrange 	<p>22H</p>
<p><u>Acquis</u></p> <p>L'étudiant doit être capable de résoudre des problèmes liés aux mathématiques, aux sciences physiques, ou à n'importe quel autre domaine cité ci-haut.</p>		