

Description des Unités d'Enseignement d'Informatique

Semestre 1

Module : Algorithme et programmation 1 (9 ECTS)

Volume : 27h CM + 36h TD + 12h TP

Pré-requis : aucun

Objectifs :

- Assimiler les bases de
 - la programmation impérative
 - la « pensée informatique »

Compétences :

- Comprendre un programme simple
- Concevoir puis écrire des séquences structurées d'ordres à une machine dans un langage de programmation (ici Python 3)
- Utiliser quelques bibliothèques de base
- Réaliser un petit logiciel
- Connaître son environnement de travail (système de fichiers, commandes de base)

Contenu :

- Valeur, expression, types simples (entiers, caractères, booléens), types complexes (chaînes de caractères, tuples, listes)
- Variables
- Exécution conditionnelle
- Boucles (sur une liste de valeurs)
- Boucles (selon une condition)
- Fonctions
- Algorithmique de base sur les listes
- Utilisation d'objets
 - *message passing* (exemples chaînes, listes, graphisme)
 - graphisme type plot, de type tortue et de type lignes/cercles...
- Mini-projet

Bibliographie :

- Gérard Swinnen, *Apprendre à programmer avec Python 3*, Eyrolles, 2012.
Il existe un version pdf gratuite en ligne.

Ressources pédagogiques :

- Outils (gratuits)
 - Jupyter notebook (<https://jupyter.org/>)
 - Thonny (<http://thonny.org/>)
- À disposition sur Celene, en plus de divers pointeurs
 - sujets de TD (en avance pour préparer)
 - supports de cours partiel (pour compléter ses notes)

Semestre 2

Module : Algorithmique et programmation 2 (7 ECTS)

Volume : 18h CM + 30h TD + 12h TP

Pré-requis : Programmation structurée

Objectifs :

- Approfondir la programmation et résoudre des petits problèmes

Compétences :

- Manipuler et développer une structures de données (arbres, piles, ...)
- Maîtriser la récursivité
- S'initier à la modularisation de son code (modules)
- Traiter des problèmes avancés, comprendre des algorithmes élaborés et les mettre en oeuvre (exemples : compression de texte, manipulation d'image, petit protocole cryptographique, algorithmique géométrique, algorithmes génétiques, réseaux de neurones, ...)
- Idées : mini-projets / programming challenge
- Approfondissement du langage Python 3

Contenu :

- Notion de programme, Machines universelle de Turing, électronique et programmation, Machine de Von Neumann
- Notion d'algorithme, algorithmes vs programmes vs code exécutable. Exemples d'algorithme. Spécification, problème de l'existence d'une solution
- Diagramme de décision, pseudo-langage
- Preuve par invariants de boucles
- Recherche linéaire, sentinelle, recherche dichotomique
- Retour sur les invariants de boucles, exemples, cas de la recherche dichotomique
- Tri par sélection, Tri par insertion, tri fusion et tri rapide
- Récursivité, Correction de programmes récursifs, retour sur python, récursivité terminale
- Structures de données : listes, piles, files et arbres
- Structures de contrôle avancées et parallélisme

Module : Logique-démonstration (4 ECTS)

Volume : 12h CM + 18h TD

Pré-requis : aucun

Objectifs :

- Donner les pré-requis mathématiques nécessaires à la poursuite d'études en licence et masters informatiques.

Compétences :

- Logique. Simplification de formules et modélisation d'énoncés.
- Relations. Relations d'équivalence. Relations d'ordre.
- Définitions inductives. Preuves par induction.

Contenu :

- Ensembles, applications, produit cartésien, relations.
- Logique des propositions et des prédicats, équivalences remarquables.
- Relations d'équivalence, relations d'ordre partiel.

- Définitions inductives et preuves par induction.

Bibliographie :

- Michel Marchand, *Mathématique discrète*, De Boeck Université.

Ressources pédagogiques :

- feuilles de TD

Module : Représentation et codage de l'information (3 ECTS)
--

Volume : 13h CM + 6h TD + 6h TP

Pré-requis : Notions d'algorithmique et de programmation en Python

Objectifs :

- L'informatique est une science dont l'objet est le traitement effectif, par des algorithmes, de l'information numérique. L'information manipulée en informatique est codée sous forme de suites binaires, de suites de 0 et de 1. L'objet de ce cours est de comprendre comment la riche famille des données discrètes manipulées par les programmes informatiques contemporains peut être représentée, codée, sous forme binaire.

Compétences :

- Savoir coder et décoder nombres et textes sous forme binaire
- Acquérir des notions autour des formats de fichiers
- Acquérir des notions autour de la représentation des images, sons et vidéo
- Comprendre l'objectif et les principes des codes de correction d'erreur
- Comprendre l'objectif et les principes de la compression de données

Contenu :

1. Codage binaire de l'information
2. Représentation des nombres entiers
3. Représentation des nombres flottants
4. Représentation des données textuelles
5. Codes détecteurs et correcteurs d'erreurs
6. Codage des images et des sons
7. Codage des séquences vidéo
8. Théorie de l'information et compression de données

Bibliographie :

- Andrew Tanenbaum. *Architecture des ordinateurs*.
- Gilles Dowek et al. *Introduction à la science informatique*. Chapitre Représentation numérique de l'information.
- Didier Müller. *Informatique (presque) débranchée*. Chapitre Codage de l'information.