

# Programme

## (Licence Informatique)

### CONTENUS PEDAGOGIQUES :

#### 1<sup>ère</sup> Année L1 (PCMI : Programme Commun Mathématiques Informatiques)

##### Semestre 1

##### UE1 (Fondamentale) 15 crédits

###### • Analyse 1

- Nombres réels et nombres complexe.
- Suites et limites.
- Fonctions à une variable réelle, continuité, dérivabilité.
- Théorème des accroissements finis
- Formule de Taylor et développements limités
- Fonctions élémentaires

###### • Algèbre 1

- Rappels sur l'anneau  $Z$  (théorème de Bézout, équations diophantiennes, idéaux, congruences)
- Applications d'ensembles: injection, surjection, bijection, image réciproque, restriction, prolongement, représentation.
- Relations binaires sur un ensemble: équivalence, ordre.
- Structures algébriques: monoïde, demi-groupe, groupe, exemples.
- Homomorphismes de groupes, isomorphismes, endomorphismes, automorphismes, exemples.
- Anneau de polynômes  $Z[X]$ ,  $R[X]$ ,  $C[X]$ , zéros, polynômes irréductibles.

###### • Informatique 1

###### Objectif :

L'objectif de cette première unité d'introduction à la discipline informatique est de permettre aux étudiants de mieux comprendre les principes de fonctionnement d'une machine et d'un logiciel, ainsi que certains principes de base de la programmation.

- Initiation aux concepts fondamentaux de fonctionnement d'un ordinateur : présentation des composants de base d'une machine et des relations entre ces différents composants.
- Initiation à l'algorithmique et à la programmation :

- Connaître ce qu'est un algorithme, la démarche algorithmique et les énoncés nécessaires à sa représentation en pseudo code.
- Comprendre le fonctionnement de l'exécution d'un programme
- Appliquer les techniques et les règles de programmation en langage C (l'apprentissage du langage C) se fera progressivement en TD et TP.

#### Programme :

- Introduction à l'informatique
  - Structure d'un ordinateur
  - Représentation de l'information
  - Calcul d'expressions logiques
- Mécanismes d'exécution d'un programme :
  - Instructions
  - Phase d'élaboration d'un programme
- Conception d'algorithme
  - Processus de résolution d'un problème.
  - Entrée/ Sortie et Variables
  - Structures de contrôle
- Langage Algorithmique
- Découpage en sous programmes
- Structures de données
  - Tableaux
  - Chaînes de caractères
  - Fichiers

### UE2 (de découverte) 9 Crédits

- **Mécanique du Point** (même programme que SM et STPI)
- **Electricité** (même programme que SM et STPI)
- **Physique optique** (optionnelle : même programme que SM et STPI)
- **Chimie** (optionnelle : même programme que STPI)

### UE3 (Méthodologique) 6 Crédits

- **TP Bureautique**

#### Objectif :

Apprentissage de l'interface graphique Windows (Système Windows), et des outils de bureautique pour la conception de documents sous différents formats : Word, Scientific Word, PowerPoint, Excel, FrontPage.

Familiarisation avec les services d'Internet : Internet Explorer (navigation sur Internet), Moteurs de recherche (Google, Altavista, ...), Messagerie électronique, ...

- **Techniques d'expression et de communication**

- Techniques d'expression écrite : mémoire, rapport, synthèse, etc.
- Techniques d'expression orale : soutenance, exposé, utilisation des moyens de communication modernes. Expression et communication dans un groupe.

- **Anglais 1**

- Amélioration de la compétence linguistique générale sur le plan de la compréhension et de l'expression

## Semestre 2

### UE4 (Fondamentale) 12crédits

- **Analyse 2**

- Intégrales définies, primitives.
- Equations différentielles du 1<sup>er</sup> et 2<sup>ème</sup> ordre à coefficients constants.

- **Algèbre 2**

- Espaces vectoriels de dimension finie, bases, sous-espaces.
- Applications linéaires, matrice d'une application linéaire.
- Déterminants.
- Applications aux systèmes d'équations linéaires, système de Cramer.
- Opérations sur les matrices.

- **Statistique descriptive**

Chapitre 1. Séries statistiques à une variable

- 1- Population. Individu. Echantillon. Caractères quantitatifs, variables statistiques discrètes et continues.
- 2- Effectif. Fréquence. Pourcentage.
- 3- Effectif cumulé. Fréquence cumulée.
- 4- Représentations graphiques: diagramme à bande, diagramme circulaire, diagramme en bâton. Polygone des effectifs (et des fréquences). Histogramme. Courbes cumulatives.
- 5- Caractéristiques de position: mode, moyenne arithmétique, moyenne harmonique, moyenne géométrique, médiane.
- 6- Caractéristiques de dispersion: étendue, variance et écart-type, coefficient de variation, quartiles, étendue interquartile.
- 7- Représentation graphique des résultats à l'aide du box-plot.

Chapitre 2. Séries statistiques à deux variables

- 1- Tableaux de données (tableau de contingence). Nuage de points.
- 2- Distributions marginales et conditionnelles. Covariance.
- 3- Coefficient de corrélation linéaire. Droite de régression et droite de Mayer.
- 4- Courbe de régression, couloir de régression et rapport de corrélation.
- 5- Ajustement fonctionnel.

### UE5 (Fondamentale) 12 crédits

- **Informatique 2**

**Objectif :**

Au second semestre sont abordées les notions de base de la modélisation informatique de problème : analyse et modélisation d'un problème, algorithmique et programmation. L'enseignement s'appuie sur un langage impératif et typé (Pascal ou C).

De plus, un enseignement est conçu autour d'une étude de cas dont le thème porte sur une application de l'informatique à la résolution d'un problème de mathématique ou de physique

- Approfondir les notions de base de la programmation
- Etudes de nouvelles structures de données
- Etude de quelques techniques algorithmes plus complexes : méthodes de tri et de recherche

On insistera sur la distinction entre l'aspect abstrait et l'aspect implémentation d'une donnée.

### **Programme :**

- Rappel
- Manipulation de tableaux
  - Méthodes de recherche
  - Méthodes de tri
  - Notion de complexité
- Manipulation de fichiers
  - Les structures d'enregistrements
  - Traitements de fichiers structurés
- Allocation dynamique
- Structures de données : listes
- Structures de données : piles

## **• Calcul Formel**

1. Introduction à la programmation fonctionnelle
2. Notions fondamentales
  1. L'interprétation et l'évaluation
  2. La fonction
  3. Les types
  4. La récursivité
  5. La liste
3. Présentation du langage CaML
  1. La boucle d'interprétation
  2. L'évaluation
  3. Définition des fonctions
  4. La précedence des opérateurs
  5. Déclaration de types
  6. Récursivité
  7. Filtrage
  8. Exceptions, fonctions partielles
  9. Les listes
4. Polymorphisme et ordre supérieur
  1. Fonctions curriifiées
  2. Polymorphisme

### **TP :**

- Apprentissage d'un langage de calcul scientifique (Mathematica, ...)
- Quelques techniques de résolution des problèmes numériques,
- Evaluation des performances (prévision/efficacité) d'une méthode de calcul.

## • **Structure Machine**

### **Objectif:**

Prendre connaissances de la théorie formelle basée sur l'Algèbre de Boole pour la synthèse des circuits.

### **Plan du cours:**

## **Partie 1**

- Les systèmes de numération
- Les conversions entre ces systèmes
- Les opérations de base (base 2, base 16, base 8)
  - Addition
  - Soustraction
  - Multiplication
  - Division
  - Le complément à 1 et 2
  - Les différents codages

## **Partie 2 : Algèbre de Boole**

- Définition
- Définition axiomatique de l'algèbre de Boole
- Théorèmes et propriétés de l'algèbre de Boole
  - Principe de dualité
  - Théorèmes fondamentaux
  - Précédence des opérateurs
  - Diagramme de Venn
- Fonctions booléennes
- Manipulations algébriques
- Complément d'une fonction
- D'autres Opérateurs Binaires

### **Simplification des fonctions booléennes**

- Méthode de Karnaugh
- Table à deux et trois variables
  - Propriété des carrés adjacents
- Table à quatre variables
- Table à cinq et six variables
- Simplification en produits de somme
- Conditions indéfinies et fonctions incomplètes
- Méthode de Quine – Mc Cluskey
  - Détermination des monômes premiers
  - Sélection des monômes premiers

### **Les circuits combinatoires**

- Analyse d'un circuit combinatoire
- Synthèse d'un circuit combinatoire

**Exemple :** Additionneur .

Un circuit particulier : les Multiplexeurs / Demultiplexeurs

**• Technologie Web**

- Introduction à l'Internet
- Réseau et Communication
- Introduction au World-Wide-Web (WWW) : technologies Web, protocole HTML , format d'une page web, outils de création d'un site web
- Technologies des données : son, image, animation et vidéo, outils pour le développement multimédia
- Interactivité sur le Web : rôle des applets

**• Histoire des Sciences****Présentation:**

L'histoire des sciences est d'une importance capitale quand il s'agit de comprendre les civilisations et l'évolution de l'esprit humain à travers les âges. L'histoire des sciences nous aide aussi à apprécier les tentatives des hommes dans leurs efforts à comprendre leur environnement et à le maîtriser. Elle sert enfin, à travers ses dimensions pédagogiques, scientifiques, didactiques, épistémologiques et culturelles à améliorer le contenu du savoir et sa transmission vers les apprenants.

Ce module vise :

- A étudier l'évolution des idées scientifiques, l'élaboration des outils et leur utilisation dans la résolution de problèmes concrets puis théoriques.
- A suivre les différentes étapes de la formation des concepts scientifiques, en se basant sur des textes originaux.
- A sensibiliser les étudiants à la dimension civilisationnelle de la pratique scientifique et à l'importance et au rôle de l'environnement culturel dans lequel naissent et se développent les sciences et dans lequel travaillent les hommes de science.

**Programme :****I. Apparition de la science, ses caractéristiques**

- a) Naissance et développement des activités scientifiques,
- b) Interaction entre science et société.

**II. Les sciences dans les civilisations anciennes**

- a) Contenu des sciences dans la civilisation babylonienne (médecine, astronomie, mathématiques, botanique),
- b) Contenu des sciences dans l'ancienne civilisation égyptienne (médecine, astronomie, mathématiques, architecture, chimie),
- c) Quelques aspects de la civilisation indienne et chinoise.

**III. Les sciences dans la civilisation grecque**

- a) Ecoles philosophiques grecques,
- b) Euclide et le livre des Eléments,
- c) Diophante et la science du nombre,
- d) Ptolémée et l'astronomie,
- e) Archimède et la méthode infinitésimale,
- f) Apollonius et les coniques
- g) Hippocrate et les sciences médicales.

**IV. Les sciences dans la civilisation arabe**

- a) Traduction en arabe d'ouvrages scientifiques écrits dans diverses langues,
- b) L'algèbre ou la naissance d'une nouvelle discipline,
- c) Les sciences expérimentales chez les arabes (mécanique, optique, chimie, botanique, agriculture, médecine...).

**V. Les sciences dans la civilisation européenne**

- a) Traduction en latin d'ouvrages scientifiques arabes et circulation des sciences grecques et arabes en l'Europe,
- b) Introduction à la période de la renaissance en Europe (Fibonacci, Léonard de Vinci, Cardan, Galilée, Copernic),
- c) Introduction à la période de la révolution scientifique en Europe (Pascal, Descartes, Leibniz, Newton).

**Références:**

- .1970 6 ( ) : .1
- 3 .2
- .1997
- .1981 : .3
- .1985 (15) : .4
- : .5
- .1968
- : .6
- .1989
- .1941 : .7
- 1 ( ) .8
- .1998 1990/12/3
- 1993 ( ) .9
- .1996
10. DJEBBAR, A. : Enseignement et recherche mathématique dans le Maghreb des 12e s.-14es., publication mathématique d'Orsay N°81-02, Université Paris-Sud., 1981.
11. DJEBBAR, A. : Mathématiques et Mathématiciens dans Maghreb médiévale (IXe-XIIIe siècles) : contribution à l'étude des activités scientifiques de l'occident musulman, thèse de Doctorat, Université de Nantes, 1990.
12. DJEBBAR, A. : Un histoire de la science arabe, Paris, le Seuil, 2001.
13. DIEUDONNE, J. : Abrégé d'histoire des mathématiques, Hermann, 1978.
14. GILLISPIE, Ch. C. (édit.) : Dictionary of Scientific Biography, New York, Scribner's son, 1970-1980, 16 vol.
15. MAITTE, Bernard : Histoire de la lumière, Paris, Seuil, 1987.
16. MARTZLOFF, J. C.: Histoire des mathématiques chinoises, Paris, Masson, 1988.
17. RASHED, R. : Entre Arithmétique et Algèbre, Paris, Les Belles Lettres, 1984.
18. ROSMORDUC, J. : Une histoire de la physique et de la chimie, Le Seuil, 1985.
19. SARTON, G. : Introduction to the History of Science, Baltimore, Williams & Wilkins, 1927.
20. SEDILLOT, I.-A. : Mémoire sur les instruments astronomiques des Arabes, Paris, Imprimerie Royale, 1844.
21. VERNET, J. : La cultura hispanoarabe en Oriente y Occidente, Madrid, 1978. Traduction française sous le titre "Ce que la culture doit aux Arabes d'Espagne", Paris, Sindbad, 1985.

22. Youschkevitch A. P. : Les mathématiques arabes (VIIIe-XVe siècles) : M. Casenave & K. Jaouiche (trad. partielle), Paris, Vrin, 1976.

## • Anglais 2

### Objectif :

Soutenir une conversation technique avec un interlocuteur anglophone, comprendre et rédiger des documents techniques. Chaque étudiant aura la possibilité de se présenter au TOEFL. Ce cours est organisé en groupes de niveau :

### Plan du cours :

- Anglais de base
- Anglais technique
- Préparation au TOEFL.

## 2<sup>ème</sup> Année L2 (Licence d'Informatique)

### Semestre 3

UEI 7 (Fondamentale) 14 crédits

## • Architecture des Ordinateurs

### Objectif :

Le module se décompose en trois grandes parties : La première partie porte sur l'architecture externe d'un processeur 32 bits, et insiste sur la définition de l'interface matériel / logiciel et la programmation en assembleur. On illustre les concepts sur l'exemple du processeur industriel MIPS R3000. La seconde partie consiste en un rappel des fondements théoriques des systèmes matériels numériques synchrones. La troisième partie présente l'architecture interne du processeur MIPS R3000, dans une réalisation micro-programmée. On peut résumer les objectifs de la façon suivante :

- Présenter les organes principaux d'un ordinateur et leurs interactions : Processeur, Mémoire, Organes Périphériques.
- Définir l'interface matériel / logiciel et introduire les concepts de langage machine et de langage d'assemblage. Initier les étudiants à la programmation en langage d'assemblage.
- Expliciter les étapes de la transformation d'un programme écrit dans un langage procédural tel que le langage C en un code exécutable en langage machine. (on va jusqu'à la description fine de l'utilisation de la pile pour les appels et retours de procédures.
- Décrire les mécanismes matériels permettant à un processeur de supporter un fonctionnement multi-tâches sous le contrôle d'un superviseur, ainsi que le mécanisme général de traitement des interruptions.
- Démystifier l'objet microprocesseur en analysant l'architecture interne d'un processeur microprogrammé (découpage partie contrôle / partie opérative et réalisation de la partie contrôle comme un automate d'états synchrone)
- Initier les étudiants aux techniques de microprogrammation, très largement utilisées dans les systèmes informatiques industriels.

Programme :

- La machine de Von Neuman. Les relations entre le processeur et la mémoire. Le concept d'instruction et de langage machine. La représentation des différents types de données en mémoire.
- Architecture externe du microprocesseur 32 bits MIPS R3000 : Les registres visibles du logiciel. L'adressage et la structuration de l'espace adressable. Le langage d'assemblage du processeur MIPS R3000.
- La programmation structurée et les appels de procédures : L'utilisation de la pile pour les variables locales, les sauvegardes de contextes, et le passage des paramètres. Le rôle du compilateur et le partage des tâches entre le matériel et le logiciel.
- Les deux modes utilisateur / superviseur comme support matériel au fonctionnement multi-tâches et multi-utilisateurs : Le rôle du système d'exploitation. Le traitement des interruptions, exceptions et trappes.
- Architecture générale d'un ordinateur moderne. Rôle des mémoire cache et hiérarchie mémoire. Communications entre le processeur et les organes périphériques. Rôle du bus système et mécanismes d'entrées/sorties.
- Algèbre de Boole. Simplification des sommes, des produits booléens. Correspondance entre expressions booléennes et implantation matérielle. Réalisation des principaux opérateurs combinatoires.
- Logique séquentielle. Modélisation des systèmes numériques synchrones, réalisation des registres et mémoires. Notions de temps de propagation / temps de pré-établissement / temps de maintien.
- Théorie des automates d'état synchrones comme modèle général des systèmes numériques synchrones. Synthèse et implantation matérielle des automates de Moore et de Mealy.
- Architecture interne du microprocesseur MIPS R3000 microprogrammé : Décomposition entre partie opérative et microséquenceur centralisé. Principe de la microprogrammation.
- Description structurelle complète de la partie opérative du processeur : registres, opérateurs de calcul, bus de communications, en utilisant les opérateurs matériels introduits dans la seconde partie du cours.
- Modélisation et réalisation du micro-séquenceur comme un automate d'état synchrone. Correspondance entre microprogramme et automate. Microprogrammation effective de quelques instructions.

Référence bibliographique :

- Architecture des ordinateurs : Interface Matériel / Logiciel David Patterson / John Hennessy

- **Algorithmique et Structures de données 1**

Objectif:

Comprendre les notions d'algorithme, de structure de données et de complexité. Sensibiliser à la notion de preuve d'algorithme. Montrer l'impact du choix des structures de contrôle et des structures de données sur la complexité. Acquérir la connaissance des structures de données de base et des algorithmes de base sur les tris, les arbres, et les graphes. Acquérir un savoir-faire théorique et pratique sur ces notions à travers Cours, TDs et TP.

Programme :

- Notion d'algorithme et preuve d'algorithme.
- Complexités d'un algorithme.
- Structures séquentielles: piles, files et listes.
- Structures hiérarchiques: arbres.
- Structures hiérarchique: arbres binaires de recherche.

- Structures hiérarchiques : Les tas.
- Structure en table: Hachage.
- Introduction aux graphes: définitions, connexités, représentations, graphes particuliers.
- Parcours de graphes: cas des graphes non orientés; parcours particuliers: en profondeur et largeur.
- Parcours de graphes: cas des graphes orientés; parcours particuliers: en profondeur et largeur.
- Algorithme de Dijkstra.

### Références bibliographiques

- Introduction to Algorithms; Cormen, Leiserson et Rivest; Wiley
- Eléments d'Algorithmique; Berstel, Beauquier et Chrétienne; Masson
- Types de données et algorithmes; Gaudel, Froidevaux et Soria; INRIA

## • Systèmes d'Information

### Objectif :

Ce cours s'articule autour de trois parties essentielles : le monde de l'entreprise, les outils d'analyse fondamentaux et l'introduction aux systèmes d'information

### Programme :

1. Définition d'une organisation
2. L'entreprise (différentes formes de structure, principales fonctions de l'entreprise, les différentes sous systèmes de l'entreprise)
3. L'entreprise et la technologie de l'information et communication : parler des TIC et de leur rôles quant à l'amélioration de la qualité des systèmes.
4. Les outils d'analyse (notion d'information, représentation de l'information, codification, contrôle, coût de stockage et de transport de l'information, confidentialité)
5. Fichiers
6. Introduction aux systèmes d'information (rôle, place, cycle de vie)
7. Notions de méthodes d'analyse et de conception d'un SI.

## UEI 8 (Fondamentale) 12 crédits

### • Analyse numérique

- Notions d'erreurs
- Approximation et interpolation polynomiale
- Dérivation et intégration numérique
- Résolution des systèmes linéaires
- Résolution d'équations et systèmes non linéaires

### • Probabilités Statistiques

#### 1-Théorie des probabilités

- Introduction au calcul de probabilités : Notion d'événement aléatoire, définition classique et axiomatique de la probabilité ; indépendance d'évènements et probabilité conditionnelle (probabilités totales et formule de Bayes).
- Variables aléatoires (discrètes et continues) : densité de probabilité et fonction de répartition ; moments (espérance mathématique, dispersion..) ; lois de probabilités usuelles (Binomiale, Géométrique, Poisson), approximation de la loi binomiale par une loi de Poisson, (Uniforme, Exponentielle , Normale ou Gauss).
- Fonction de variables aléatoires, fonction caractéristique, fonction génératrice, transformée de Laplace.
- Vecteurs aléatoires : Etude élémentaire d'un couple de variables aléatoires, lois conjointe et marginale, covariance et corrélation, loi conditionnelle ; loi normale multidimensionnelle.

- Modes de convergence (en probabilité, presque sûre) ; loi des grands nombres, Théorème de limite central.

## **2- Statistique Inférentielle**

- Echantillonnage : constitution d'échantillon, moments empiriques, Distributions d'échantillonnage ;
- Estimation : estimation ponctuelle (qualité d'un estimateur, construction d'estimateurs par la méthode des moments et la méthode du maximum de vraisemblance), Intervalles de confiance.
- Tests d'hypothèses : Principe de construction de tests basés sur les intervalles de confiance ; tests de comparaisons de moyennes et de proportion ; test de la variance. Test du Khi-deux (d'ajustement et d'indépendance). .

## **• Logique Mathématique**

### **Objectif :**

Ce cours a pour objectif de donner aux étudiants des notions de calculabilité et les bases de la logique formelle à partir de l'étude de la logique propositionnelle

### **Programme:**

I : Calculabilité

- I.1 Les fonctions récursives et les fonctions primitives récursives
- I.2 Les Machines de Turing
- I.3 Thèse de Church

II : Le calcul Propositionnel

- II.1 Le Langage
- II.2 Dédution de Gentzen
- II.3 La sémantique
- II.4 Théorème de consistance et de complétude.
- II.5 Algorithme de réfutation

III : Calcul des Prédicat

- III.1 Langage
- III.2 Dédutions
- III.3 Interprétation
- III.4 Formes prénexes et forme de Skolem

UEI 9 (méthodologique) 4 crédits

## **• Anglais 3**

Techniques d'expression orale : exposé, soutenance et communication en groupes

## • Cognition

**Objectifs (*compétences visées*):** ce cours a pour objectif de permettre à l'étudiant d'approfondir ses connaissances dans le domaine de la cognition tel que le rapport de la représentation et extraction des connaissances avec la cognition, traitement du langage naturel et la vision.

### Programme :

- o Science cognitive
  - Nature de la science cognitive
  - Ordinateurs dans la science cognitive
  - Science cognitive appliquée
  - Nature pluridisciplinaire de la science cognitive
  
- o Intelligence artificielle
  - La nature de l'IA
  - Représentation des connaissances
  - Extraction des connaissances
  
- o I.A : Recherche, contrôle et Apprentissage
  - Recherche et contrôle
  - Technique de recherche des heuristiques
  - Apprentissage
  
- o Linguistique : représentation du langage
  - Etude de la connaissance linguistique
  - Syntaxe
  - Grammaires
  
- o Résolution de problèmes
  
- o Langage naturel
  
- o Vision 'assisté par ordinateur'

## Semestre 4

## UEI 10 (Fondamentale) 16 crédits

- **Bases de données**

- Objectif:

- Comprendre les objectifs, les architectures et les langages de bases de données. Maîtriser les fondements théoriques et les algorithmes de base des systèmes de gestion de bases de données, depuis la conception de base de données jusqu'au traitement de requêtes et la gestion de transactions. Le module s'appuie sur le modèle relationnel et les langages associés, en particulier SQL et sur le SGBD ORACLE par exemple.

- Programme :

- Introduction: besoin de SGBD dans les applications, objectifs des SGBD, modélisation des données et niveaux d'abstraction, modélisation Entité Association., modèle relationnel
    - Les langages relationnels :l'algèbre relationnelle , langages prédicatifs et SQL (Interrogation d'une base de données en SQL: requêtes simples, requêtes imbriquées, agrégats et groupement).
    - Conception et optimisation de schéma relationnel : notion de redondance, dépendance fonctionnelle, déduction (axiome d'Armstrong) et couverture minimale, formes normales.
    - Architecture :
      - Introduction : différentes étapes d'analyse d'une requête (interprétation, optimisation).
      - Définition et modification d'une base de données en SQL:
      - création des tables, insertion, suppression et mise à jour des données.
      - Création d'index primaire et secondaire, accès en B-arbre.
      - Contraintes d'intégrité : typologie, vérification, Triggers: définition (événement, condition et action).
      - Contrôle de concurrence : notion de transaction, sérialisabilité, verrouillage deux phases, interblocages, ordonnancement par estampillage.
      - Les reprises après pannes. (journalisation, validation, reprise à froid et à chaud).

- Références bibliographiques :

- Georges Gardarin. Bases de données: objet et relationnel. Eyrolles, 1999.
    - Raghu Ramakrishnan, Johannes Gehrke. Database Management Systems. 2nd edition. Mc Graw-Hill,1999.
    - Tamer Özsu, Patrick Valduriez. Principles of Distributed Database Systems. 2nd edition, Prentice Hall, 1999.

- **Algorithmique et Structures de données 2**

- Objectif:

- L'orientation souhaitée pour l'associer à ce cours est: " présenter les mécanismes offerts par les langages de programmation et identifier leur usage dans le contexte d'un développement modulaire." Les concepts de base présentés iront jusqu'à " l'orée de l'univers objet " et seront illustrés avec pascal (principalement) et C (en particulier sur les aspects interfaçage). Il est souhaité également de présenter aux étudiants un environnement de production dont la philosophie, orientée " production, " s'inspire de ce que l'on peut trouver dans le monde industriel. Afin de détailler les notions que l'on trouve dans les langages de programmation, il sera utilisé en TD et en TP Pascal (choisi pour le premier semestre de la licence). Des comparaisons seront faites avec le langage

C (choisi pour le second semestre de la licence). Outre l'écriture de programmes en Pascal, il sera demandé aux étudiants de lire et comprendre des services écrits en C.

### Programme :

- Organisation du cours, Description de l'environnement de travail. rappel des notions de base (types simples, structure d'un programme etc.)
- Rappel des notions de base. Ecriture de programmes simples.
- Structuration d'un programme. Procédures et fonctions. Récursivité. In-lining.
- Types tableaux et types articles ("record"). Notion de paquetage.
- Notion de types abstraits. Protection avec les types (privés, public, etc.). Contrôle de la visibilité en Ada et C.
- Pointeurs et gestion de la mémoire dynamique (tas). Structures de données récursives.
- Echappement et gestion des erreurs par exception.
- Truc et astuces pour déboguer un programme.
- Généricité.
- Vers les mécanismes objets. Présentation de la notion de fichiers.
- Notions avancées de structuration. Conclusions sur le cours.

### Références bibliographiques :

- "Programmer en Pascal" de (Addison Wesley)
- "Le langage C - norme ANSI", B.W. Kernighan et D. M. Ritchie (Dunod)
- "Méthodologie de la programmation en langage C", J-P Braquelaire (Masson)

## • Systèmes d'exploitation 1

### Objectif:

#### *Chapitre 1 : Introduction aux systèmes d'exploitation*

1. Définition d'un S.E
  2. Fonctions d'un S.E
  3. Organisation en couches d'un S.E - Virtualisation de la machine
  4. Evolution des systèmes informatiques
4. Exemples de S.E

#### **Chapitre 2 :** Mécanismes de base d'exécution des programmes

1. Structure matérielle d'une machine de Von Neumann
2. Cheminement d'un programme dans un système
3. Concepts de processus et multiprogrammation (contexte d'un processus, états, mécanisme de commutation de contexte).
4. Les systèmes d'interruption
  - Définition et organigramme général d'une interruption.
  - Mécanismes de gestion des interruptions
  - Systèmes d'interruption sur les PCs.

#### **Chapitre 2 :** Gestion des Entrées / Sorties physiques

1. Définition d'une E/S
2. Types d'E/S
3. Organisation des transferts (instructions d'E/S, découpage fonctionnel matériel/logiciel d'une E/S)
4. Modes de pilotage d'une E/S : mode synchrone, asynchrone, canal
5. Gestion des E/S simultanées.

#### **Chapitre 3 :** Gestion du processeur central

1. Définition du scheduling / Scheduler.
2. Objectifs de scheduling.
3. Critères de scheduling.
4. Niveaux de scheduling (scheduling des jobs, scheduling des processus).
5. Politiques de scheduling.
6. Contrôle de processus ( Etats d'un processus, Bloc de contrôle de processus PCB, création de processus, destruction, ...).

**Chapitre 4 : Gestion de la mémoire centrale**

1. Objectifs d'un gestionnaire de la mémoire.
2. Fonctions.
3. Modes de partage de la mémoire.
4. Protection de la mémoire.
5. Partage de code.

**Chapitre 5 : Gestion des périphériques****Chapitre 6 : Gestion des fichiers****Références bibliographiques :**

- A. Silberschatz, P. Galvin Principes des Systèmes d'Exploitation, Addison-Wesly, 1994
- A. Tanenbaum Systèmes d'Exploitation : Systèmes Centralisés, Systèmes Distribués Prentice-Hall 1994
- G. Nutt Les Systèmes Ouverts, InterEdition 1995

UEI 11 (Fondamentale) 12 crédits

**• Théorie des langages****L'objectif:**

L'étudiant, par cette unité, doit connaître et maîtriser les concepts issus de la théorie des langages et plus particulièrement des langages algébriques, de grammaire, de dérivateur, d'automate à pile qui sont en fait à la base de tous les algorithmes d'analyse syntaxiques.

**Programme:****○ Les langages**

- Introduction et rappels mathématiques
- Opérations sur les langages
- Représentation des langages : grammaires et automates
- Hiérarchie de Chomsky

**○ *Les automates d'états finis***

- automates déterministes et minimisation
- Automates indéterministes et passage à un automate déterministe

**○ *Les langages réguliers***

- propriétés des langages réguliers
- Expression régulières
- Passage des expressions régulières aux automates et réciproquement
- Grammaire et automates (grammaire de Kleene)

**○ *Les langages algébriques***

- Propriétés des langages algébriques
- Les automates à pile

**○ *Les langages à contexte lié***

- définition et propriétés
- Les automates à bornes linéaires

## o *Les machines de Turing*

- Notion de machine de Turing
- Langages de type 0 et machine de Turing
- Introduction à la calculabilité

### **Références bibliographiques :**

- [1] H.Hopcroft, D.Ullman.: Introduction to automata, theory langages and computation.  
 [2] M.Gross and A.Lentiu: Introduction to formal grammars.  
 [3] Patrice Séebold: Théorie des automates.  
 [4] A.V.Aho and J.D.Ullman: Principles of Compiler Design

## • **Programmation linéaire**

**Objectifs (compétences visées):** Ce cours dresse un panorama des techniques de modélisation utilisées en programmation linéaire, il permet le développement d'applications industrielles en optimisation.

### **Programme :**

- o **Rappels Mathématiques (Algèbre linéaire)**
  - Espace vectoriel
  - Dimension, base
  - Matrice, déterminant d'une matrice, inverse d'une matrice ...
- o **Introduction et propriétés de la programmation linéaire**
  - Forme générale d'un programme linéaire, forme canonique, standard et mixte.
  - Résolution graphique, notion de polyèdre.
  - Résolution analytique.
- o **Méthode du simplexe**
  - Introduction de la méthode, algorithme du simplexe, tableau du simplexe
  - Méthodes particulières : méthode des pénalités, méthode des deux phases
  - Forme révisée du simplexe
- o **Dualité**
  - Introduction, règles de passage du primal au dual
  - Algorithme dual du simplexe
- o **Problème du transport**
  - Introduction du problème, graphe associé au tableau du transport
  - Algorithme du transport
  - Algorithme dual du transport.

## • **Génie Logiciel et Programmation Orientée Objet**

### **Objectifs:**

- Comprendre le processus de développement du logiciel, en particulier les phases de définition et de spécification des besoins, d'analyse et de conception orientée objets.
- Programmation Objets : Java par exemple.

**Programme :****1. INTRODUCTION AU GENIE LOGICIEL**

- Les principes du génie logiciel
- Les Cycles de vie de développement de logiciels
- Les bases de la qualité du logiciel
- Des méthodes fonctionnelles aux méthodes « Objet »
- Test et maintenance du logiciel

**2. L'APPROCHE ORIENTEE OBJET**

- Le Paradigme Orienté Objet
- Les Objets
- Les Messages ou la Communication entre objets
- Les Classes
- L'Héritage entre classes
- Notion de Polymorphisme et de Liaison Dynamique

**3. INTRODUCTION A UN LANGAGE OBJET**

- Introduction et Caractéristique
- Primitives du langage
- Concepts de programmation

**UEI 12 (méthodologique) 2 crédits****• Anglais 4**

Techniques d'expression écrite et orale : rapport, mémoire, exposé, soutenance, communication en groupes.

**3<sup>ème</sup> Année L3 (Licence d'Informatique)****Semestre 5****UEI 13 (Fondamentale) 18 crédits****• Systèmes d'Exploitation 2****Objectif :**

- Inculquer à l'étudiant les concepts et les outils de base des systèmes d'exploitation.
- Introduire la problématique du parallélisme dans les systèmes d'exploitation et étudier la mise en œuvre des mécanismes de synchronisation, de communication dans l'environnement centralisé

**Recommandations :**

- Il est conseillé d'utiliser un système d'exploitation ( UNIX par exemple) comme exemple en termes d'outils pour chaque concept étudié.
- Prévoir des TPs pour la mise en application des concepts étudiés.

- Le chapitre 5 peut faire l'objet d'un rapport demandé aux étudiants.

### **Programme :**

#### **Chapitre 1 : NOTIONS DE PARALLELISME, DE COOPERATION ET DE COMPETITION (10 %)**

- + Systèmes de tâches, outils d'expressions
- + Déterminisme et parallélisme maximal
- + Threads

#### **Chapitre 2 : SYNCHRONISATION (35 %)**

- + Problème de l'exclusion mutuelle
- + Synchronisation
  - . Evénements, Verrous
  - . Sémaphores
  - . Moniteurs
  - . Régions critiques.
  - . Expressions de chemins

#### **Chapitre 3 : COMMUNICATION (20%)**

- . Partage de variables ( modèles : producteur/ consommateur, lecteurs/ rédacteurs)
- . Boîtes aux lettres
- . Echange de messages ( modèle du client/ serveur)
- . Communication dans les langages évolués (CSP, ADA, JAVA..)

#### **Chapitre 4 : INTERBLOCAGE (20 %)**

- + Modèles
- + Prévention
- + Evitement
- + Détection/ Guérison
- + Approche combinée

#### **Chapitre 5 : ETUDE DE CAS : SYSTEME UNIX (15 %)**

- + Principes de conception
- + Interfaces (programmeur, utilisateur)
- + Gestion de processus, de mémoire, des fichiers et des entrées/sorties
- + Synchronisation et Communication entre processus.

### **Bibliographie :**

- [1] J-L. Peterson, F. Silbershartz , P. B. Galvin " Operating Systems Concepts," Fourth Edition.
- [2] Crocus, " Systèmes d'exploitation des ordinateurs," Dunod informatique 1975.
- [3] J. Beauquier, B. Berard "Systèmes d'exploitation : concepts et algorithmes" McGraw Hill 1990.
- [4] A. Silberschatz, P. B. Galvin " Principes des systèmes d'exploitation," 4 e Edition, Addison Wesley
- [5] Andrew S. Tanenbaum, " Modern Operating Systems," Second Edition Prentice Hall.
- [6] Maurice J. Bach, traduit par G.Feallah, "Conception du Système UNIX," Masson et Prentice Hall 1990.

### **• Compilation**

#### **Objectif:**

Introduction au problème de la compilation :

- du texte-source au code assembleur en passant par l'arbre de syntaxe abstraite
- sur la base d'un petit compilateur réalisé en C, en trois versions de complexité croissante. L'assembleur visé est celui du cours d'Architecture. Technique classique

d'analyse syntaxique : Lex et Yacc. Le but du cours est de montrer le rôle de la pile dans la compilation des fonctions sur le modèle de C.

### Programme :

#### **I. Introduction à la Compilation**

- Les différentes étapes de la Compilation
- Compilation, Interprétation, Traduction

#### **II. Analyse lexicale**

- Expressions régulières
- Grammaires
- Automates d'états finis
- Un exemple de générateur d'analyseurs lexicaux : LEX

#### **III. Analyse syntaxique**

- Définitions : grammaire syntaxique, récursivité gauche, factorisation d'une grammaire, grammaire  $\epsilon$ -libre.
- Calcul des ensembles des débuts et suivants.
- Méthodes d'analyse descendantes : la descente récursive, LL(1).
- Méthodes d'analyse ascendantes : LR(1), SLR(1), LALR(1), (méthode des items).
- Un exemple de générateur d'analyseur syntaxique : YACC.

#### **IV. Traduction dirigée par la syntaxe (Analyse sémantique)**

#### **V. Formes intermédiaires**

- forme postfixée
- quadruplés
- triplés directs et indirects
- arbre abstrait

#### **VI. Allocation – Substitution- Organisation des données à l'exécution**

#### **VII. Optimisation du code objet**

#### **VII Génération du code objet**

### Références bibliographiques :

- Aho, Sethi, Ullman : Compilers (Addison-Wesley) Trad. française chez InterÉditions
- Christopher Fraser and David Hanson. A Retargetable C Compiler : Design and Implementation. Benjamin/Cumming, 1995

## • **Réseaux**

### Objectif :

Ce module est une introduction au monde des réseaux informatiques. Il constitue un noyau de base des connaissances " réseaux " dont la compréhension est essentielle, car tous les concepts présentés sont utilisés dans les réseaux actuels et à venir. Il a comme objectifs :

- la compréhension des concepts fondamentaux utilisés en réseaux : structuration d'une architecture de communication en couches, notion de protocole, principaux mécanismes de communication tels que : contrôle d'erreur, contrôle de flux, contrôle de congestion, commutation, mode de communication, routage, adressage ;
- l'initiation à des protocoles courants, notamment Ethernet, IP, TCP ;
- la familiarisation de l'étudiant avec le vocabulaire " réseau ".

Nous insisterons ainsi sur les infrastructures de transport de l'information, utilisées par les applications de l'Internet. À la fin du semestre, des séances de TD sur machine permettent

de mettre en pratique les notions vues et d'effectuer une synthèse des connaissances acquises.

### Programme :

- Qu'est-ce qu'un réseau ? Définitions, historique, motivations, applications, architectures
- Transmission physique de l'information : traitement du signal, signal numérique/analogique, codage, circuit de données, multiplexage de signaux
- Fiabilisation de la transmission : contrôle d'erreur, sécurité, illustration avec les protocoles de transmission HDLC et PPP
- Normalisation : modèles OSI
- Réseaux locaux : plan de câblage, topologie, méthodes d'accès au support de communication,
- Interconnexion de réseaux : les équipements d'interconnexion (routeurs, commutateurs ou switches, ponts, hubs). illustration avec Ethernet et Token Ring
- Réseaux grande distance : techniques de commutation, adressage, routage, contrôle de congestion, illustration avec des réseaux d'opérateurs (X.25, Relais de Trames ou Frame Relay, ATM)
- Protocoles TCP /IP
- Commandes Unix pour la configuration et l'administration d'un réseau
- Installation et configuration d'un serveur web et d'un proxy
- Exemples d'applications : http, mail ; accès au réseau pour les utilisateurs ; serveurs DNS

### Références bibliographiques

- Comer, D. Ed.: InterEditions. *TCP/IP: Architecture, Protocoles, Applications.*
- Rolin, P. Ed. : Hermès. *Réseaux locaux, normes et protocoles*
- Tanenbaum, A.. Ed.: InterEditions. *Réseaux: Architectures, Protocoles, Applications.*

## UEI 14 (Fondamentale) 12 crédits

### • Programmation Logique

- Introduction à la programmation logique : langage PROLOG
- Principales caractéristiques de ce type de programmation
- Syntaxe et structures de données – opérateur de coupure
- Sémantique des programmes PROLOG
- Le problème de la négation en PROLOG : l'hypothèse du monde clos et la négation par échec.
- Utilisation de la méthode de résolution dans l'implantation machine de ce type de langage.

#### Références bibliographiques :

- Chazarain, Programmer avec SCHEME . De la pratique à la théorie. Thomson International, 1996.
- Hoogger. Programmer en logique. Masson, 1987
- Weis & Leroy. Le langage CAML. Interéditions, 1993.

### • Théorie des Graphes

**Objectifs (compétences visées) :** Appréhender les algorithmes des graphes utilisés dans les réseaux informatiques, dans les problèmes de calcul de coût minimal, dans la recherche du meilleur chemin et dans les méthodes d'ordonnancement (Gestion des projets,...)

#### Programme :

- **Notions fondamentales de la théorie des graphes**
  - Définitions d'un graphes et différentes représentations
  - Applications multi-graphes

- Applications multivoque, degré, demi degré, cycles, cocycles, connexité...
- **Les nombres fondamentaux de la théories des graphes**
  - Nombres de stabilités
  - Noyau, fonction ordinale, fonction de Grundy
  - Nombre chromatique, clique, théorème de Koenig
- **Graphes particuliers**
  - Graphes planaires, Graphe dual
  - Graphes aux arêtes, graphes aux arcs
- **Arbres et Arborescence**
  - Construction d'un arbre
  - Construction d'une forêt
  - Algorithme de Kruskall (cycles et cocycles)
  - Algorithme de Sollin
- **Problèmes de flots**
  - Définitions
  - Cycles élémentaires et flots élémentaires
  - Problème du flot maximal dans un réseau de transport
  - Graphe d'écart
  - Algorithme de recherche du flot maximal (Ford-Fulkerson)
- **Problèmes du plus court chemin**
  - Introduction au problème du plus court chemin
  - Algorithme de Dantzig
  - Algorithme de Ford
  - Algorithme de Dijkstra
- **Méthodes d'ordonnement**
  - Diagramme de Gantt
  - Méthode PERT
  - Méthode MPM

#### Références Bibliographiques:

[1] Christian Prins :Algorithmes de graphes (avec programmes en Pascal) Eyrolles, Paris, 1994.

[2] Bernard Roy : Algèbre moderne et théorie des graphes TomeII, Dunod, 1989

Le livre de M.Gondrou et M.Minoux existe en version française

Graphes et Algorithmes, Eyrolles, Paris 1984.

### • Ingénierie des connaissances

**Objectifs :** la conception d'un prototype de système de représentation de connaissances

#### - **Acquisition des connaissances**

- Apprentissage numérique
- Apprentissage symbolique

#### - **Représentation des connaissances**

- Les différents types de connaissances
  - Connaissances factuelles
  - Connaissances ontologiques
  - Connaissances assertionnelles

- Les formalismes de représentation des connaissances
  - Frame
  - Réseaux sémantiques
  - Graphes conceptuels
  - Logique
- Les différents langages de représentation des connaissances
  - KIF
  - XML
  - RDF et RDFS
  - La logique de description

### **- Les systèmes de représentation des connaissances**

- KL-ONE
- CLASSIC
- LOOMS

### **- Des systèmes experts aux systèmes à base de connaissances**

### **- Modélisation des connaissances**

### **- Conception et réalisation d'ontologies en ingénierie des connaissances**

### **- Raisonnement à partir de cas**

### **- Gestion des connaissances**

### **Références Bibliographiques**

J. Charlet, M. Zacklad, G. Kassel, D. Bourigault. « Ingénierie des connaissances : Evolutions récentes et nouveaux défis ». Editions Eyrolles. France, 2000.

### **Références bibliographiques :**

- M. Campione, K.Walrath. The Java tutorial, (second edition) 1998.
- A. Nye et T.O'Reilly. The definite Guide to X-Windows System, (Motif Edition) O'Reilly&Associates.
- Welch Brent balding. Practical Programming in TCL/TK (second edition), Prentice hall, 1997.

• ...

## Semestre 6

UEI 15 (Fondamentale) 10 crédits

- Introduction à l'Intelligence Artificielle
- Sécurité des Systèmes informatique

UEI 15 (Fondamentale) 20 crédits

- **Projet**