

Unités d'enseignements de la première année du master science, mention informatique

7 octobre 2003

Liste des UE

Unités d'enseignements obligatoires	2
(ASE) Architecture et conception des systèmes d'exploitation	2
(AAC) Algorithmique Avancée et Complexité de Problèmes	3
(AEV) Architecture évoluée	4
(CAR) Construction d'applications réparties	5
(COA) Conception objet avancée	6
Unités d'enseignements optionnelles	7
(CALP) Concepts avancés des langages de programmation	7
(FDD) Fouille de données	8
(IA) Intelligence artificielle	9
(IM/ITR) Informatique mobile & Informatique temps réel	10
(SI) Synthèse d'Images	11
(P3D) Programmation 3D	12
(PP) Programmation parallèle	13
(BI) Bioinformatique	14
(PAC) Principes et algorithmes cryptographiques	15
(BDCS) Bases de données et aspects client-serveur	16
(MSC) Modélisation des systèmes complexes	17
(HÉCI) Histoire et épistémologie du calcul et de l'informatique	18
(IHM) Ergonomie de conception des IHM et usages	19
(SVL) Spécification et vérification du logiciel	20
(AEX) Architecture experte	21

Intitulé de l'UE ARCHITECTURE ET CONCEPTION DES SYSTÈMES D'EXPLOITATION

Identifiant ASE

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Obligatoire en première année du master science mention informatique

Objectifs Ce module s'attache à l'explication des éléments fondamentaux mis en œuvre dans la conception des systèmes d'exploitation. L'étudiant doit y acquérir des notions :

- d'architecture des systèmes d'exploitation ;
- d'exploitation des matériels informatiques ;
- de sécurité fournies par les systèmes ;
- du rôle/activité des couches logicielles de base.

Il doit maîtriser les principes fondateurs des systèmes d'exploitation en vue de :

- savoir utiliser les outils logiciels fournis ;
- savoir programmer des circuits matériels ;
- savoir concevoir des composants de systèmes d'exploitation.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé La réalisation de composants système est une activité qui implique de nombreuses heures de manipulation. Ce module comporte la réalisation d'un prototype qui implique un travail personnel substantiel en plus des heures de TD et de TP encadrées.

Contrôle et validation des connaissances Sont notés :

- un examen final ;
- une interrogation écrite en TD ;
- un projet initié dans le cadre des TP (et travail personnel) et réalisé en binôme.

Accompagnement et outils pédagogiques Le cours repose sur un jeu de polys papiers aussi disponible en ligne. Les TPs supposent l'utilisation d'une salle informatique et l'exploitation des outils de compilations pour le langage C ainsi qu'une bibliothèque dédiée.

Évaluation de la charge enseignant

Description du contenu Le cours dresse un état de l'art des différentes techniques et algorithmes dédiés à gestion et à l'abstraction des ressources matérielles élémentaires, à savoir, la mémoire de travail la mémoire persistante, et le microprocesseur. La gestion des entrée/sorties est détaillée dans le cours de réseaux (UE RSX de la licence mention informatique). Un projet de TP vise à reconstituer (par la pratique) les couches composant un système de fichiers. Un autre projet à mettre en œuvre un ordonnanceur de tâches.

Nom du responsable Gilles GRIMAUD

Intitulé de l'UE ALGORITHMIQUE AVANCÉE ET COMPLEXITÉ DE PROBLÈMES

Identifiant AAC

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Obligatoire en première année du master science mention informatique

Objectifs L'accent du cours est plus mis sur les méthodes que sur la connaissance d'algorithmes classiques même si ceux-ci serviront d'exemples pour illustrer le cours.

- Connaître quelques schémas "classiques" d'algorithmes et savoir les utiliser.
- Comprendre la notion de complexité de problèmes.
- Connaître quelques méthodes pour aborder des problèmes « durs ».

Volume étudiant Douze heures de cours, 18 heures de TD et 18 heures de TP sur machines réparties sur 12 semaines.

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

- Polys de cours distribués en cours et en ligne.
- Fiches de TD et de TP distribués et en ligne.
- Problèmes « types » et leurs corrigés en ligne.

Évaluation de la charge enseignant

Description du contenu

- Schémas d'algorithmes : diviser pour régner, programmation dynamique, algorithmes gloutons
- Complexité de problèmes ; notion de réduction. La classe NP.
- Algorithmes d'exploration (branch-and-bound, min-max, A*)
- Algorithmes d'approximation : heuristiques et métaheuristiques
- Algorithmes probabilistes

Nom du responsable Sophie TISON

Intitulé de l'UE ARCHITECTURE ÉVOLUÉE

Identifiant AEV

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Obligatoire en première année du master science mention informatique

Objectifs Le but du cours est de présenter les architectures modernes des ordinateurs ainsi que des concepts les plus récents mis en oeuvre dans ces systèmes afin d'augmenter les performances : de l'analyse de l'architecture Von Neumann au Superscalaire.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- Introduction
 - Architecture Von Neumann et parallélisme
 - Classification des architectures des processeurs modernes
- Mémoire
 - entrelacée
 - hiérarchique
 - cache
 - protocole de cohérence
- Système de communications mono-bus, arbitrage
- Processeur
 - Pipeline
 - Risc/Cisc
 - Superscalaire
 - Jeu d'instruction SIMD

Nom du responsable David SIMPLOT et Jean-Luc DEKEYSER

Intitulé de l'UE CONSTRUCTION D'APPLICATIONS RÉPARTIES

Identifiant CAR

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Obligatoire en première année du master science mention informatique

Objectifs L'objectif du cours est de présenter aux étudiants les concepts de la programmation répartie ainsi qu'un ensemble d'outils permettant la mise en oeuvre d'applications dans des environnements répartis. Les concepts abordés seront la programmation en mode asynchrone et en mode synchrone. Les environnements étudiés seront les environnements TCP/UDP, les outils du Web, les environnements orientés objet tels que RMI et CORBA, ainsi que les Web services.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

- Poly papier et cours en ligne
- Exercices en ligne

Évaluation de la charge enseignant

Description du contenu

- Introduction aux applications réparties.
- Constructions d'applications réparties en mode message.
- L'environnement Web.
- Construction d'applications réparties dans un environnement OO.
- Un exemple avec CORBA.
- Environnements d'exécution à objets réparties.
- La découverte d'applications par les Web Services.
- D'autres modèles de structuration d'applications réparties.

Nom du responsable Laurence DUCHIEN

Intitulé de l'UE CONCEPTION OBJET AVANCÉE

Identifiant COA

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Obligatoire en première année du master science mention informatique

Objectifs

- Approfondir l'approche de la problématique liée à la conception d'applications (en se basant essentiellement sur une approche objet).
- Introduire le langage UML.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu Sont abordés :

- les différentes phases de la conception d'une application, cycle de vie d'une application (besoins, analyse, tests...);
- présentation des problèmes liés à ce cycle de conception et des solutions ;
- présentation de processus d'ingénierie du logiciel (ex : le RUP, le 2TUP, eXtrem Programming...);
- le langage UML (diagrammes, cas d'utilisation...);
- retour et approfondissement sur la notion de design patterns.

Nom du responsable À définir

Intitulé de l'UE CONCEPTS AVANCÉS DES LANGAGES DE PROGRAMMATION

Identifiant CALP

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Ce cours se propose de présenter plusieurs concepts originaux apparaissant dans des langages de programmation modernes s'écartant des langages les plus répandus.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu La présentation de chaque concept s'appuiera sur un langage particulier. Ces concepts n'ont pas forcément de liens entre eux et ne sont pas tous disponibles dans tous les langages abordés. Nous verrons comment ces concepts sont particulièrement bien adaptés à certaines classes d'applications et comment ils facilitent leur écriture.

Le cours est divisé en trois parties correspondant chacune à un concept, un langage associé et une classe d'applications.

Typage statique fort et inférence de type

- Langage support : Objective Caml
- Classe d'applications : manipulations symboliques

Évaluation paresseuse

- Langage support : Haskell
- Classe d'applications : structures de données potentiellement infinies

Paradigme synchrone

- Langage support : Esterel
- Classe d'applications : applications réactives

Nom du responsable Pierre BOULET

Intitulé de l'UE FOUILLE DE DONNÉES

Identifiant FDD

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Apprendre à retirer de l'information d'une masse de données :

- Quelle information peut-on espérer trouver ?
- Quelles sont les techniques utilisables, leur coût, leur rendement ?
- Quelle démarche suivre ?

À la fin du cours, les étudiants doivent être capables de définir et implémenter des algorithmes de fouilles de données, disponibles dans les suites logicielles du marché (SQL Server, Oracle, BO...), en présenter les principes et les résultats aux utilisateurs terminaux.

Volume étudiant Cours : 20h, Travaux dirigés : 10h, Travaux pratiques : 14h, Conférences : 6 heures (3 conférences)

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances

- Examen sur le contenu du cours.
- Note de travaux pratiques : illustration de techniques.

Accompagnement et outils pédagogiques Mise à la disposition des étudiants de divers matériels, par l'intermédiaire d'un site web (sur le modèle de la page actuelle, accessible depuis : www.lifl.fr/~decomite/):

- Copie des supports de cours.
- Sujets en ligne.
- Mini-challenges : recherche d'informations à partir de données fournies par l'enseignant.
- Programmes d'illustration (C et Java).
- Liens vers d'autres sites similaires.

Évaluation de la charge enseignant Classique : préparation des supports de cours, réalisation et suivi des challenges...

Description du contenu

- Définition de la fouille de données.
- Le cycle de la recherche de connaissances.
- Les données : sources, formats, vérification, nettoyage, transformations.
- Les entrepôts de données : Structures, modèles, alimentation, utilisation.
- L'analyse en ligne : OLAP, MOLAP, ROLAP..
- Les différentes méthodes de recherche de connaissances :
 - Les règles d'association.
 - Les méthodes de classification :
 - Arbres de décision.
 - Réseaux de neurones.
 - Segmentation, apprentissage non supervisé.
 - La recherche des anomalies.
- Aperçu des nouveaux domaines d'application (Web, images, textes, données hétérogènes...)

Bibliographie

- Data Mining, Concepts & Techniques *Jiawei Han & Micheline Kamber*
- Data Mining *Ian Witten & Eibe Frank*

Nom du responsable Francesco DE COMITÉ

Intitulé de l'UE INTELLIGENCE ARTIFICIELLE

Identifiant IA

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs En face d'un problème, pouvoir décider quelles méthodes d'Intelligence Artificielle peuvent s'appliquer.

- Identification de problèmes similaires.
 - Conditionnement du problème (données à utiliser, format des résultats).
 - Comparaisons de méthodes.
 - Choix de l'algorithme.
 - Exploitation du résultat.
 - Validation de la méthode par l'expérience.
- Être capable d'utiliser des logiciels (weka, SNNS), ou d'implémenter soi-même un algorithme d'IA.

Volume étudiant Cours : 20h, Travaux dirigés : 10h, Travaux pratiques : 20h

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances

- Examen sur le contenu du cours.
- Note de travaux pratiques (réalisation d'un petit projet)

Accompagnement et outils pédagogiques

- Mise à la disposition des étudiants de divers matériels, par l'intermédiaire d'un site web (sur le modèle de la page actuelle, accessible depuis www.lifl.fr/~decomite):
 - Copie des supports de cours.
 - Sujets en ligne.
 - Suivi de projet en ligne (parties de programmes, FAQ, comparaison de résultats...)
 - Programmes d'illustration (C et Java).
 - Liens vers d'autres sites similaires.
- Utilisation de logiciels d'Intelligence Artificielle (Weka,SNNS)

Évaluation de la charge enseignant Classique : préparation des supports de cours, réalisation et suivi des projets...

Description du contenu

- Définition de l'intelligence artificielle et des différentes approches.
- Apprentissage à partir d'exemples : Apprentissage supervisé. Segmentation, apprentissage non supervisé. Apprentissage par renforcement.
- Arbres de décisions : définition, usage, algorithmes, application en travaux pratiques (c4.5,weka).
- Réseaux de neurones : définition, usage, algorithmes (descente du gradient, rétropropagation du gradient), manipulation en travaux pratiques et en projet (snns, weka, programmes 'locaux').
- Approche probabiliste de l'apprentissage : la règle de Bayes.
- Réseaux bayésiens : définition, inférence, apprentissage.

Bibliographie

- Machine Learning *Tom Mitchell*
- Artificial Intelligence : A Modern Approach *Stuart Russell & Peter Norvig*
- Data Mining *Ian Witten & Eibe Frank*

Nom du responsable Francesco DE COMITÉ

Intitulé de l'UE INFORMATIQUE MOBILE & INFORMATIQUE TEMPS RÉEL

Identifiant IM/ITR

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis (Licence mention informatique). Architecture des systèmes d'exploitation (ASE).

Parcours Optionnel en première année du master science mention informatique

Objectifs Les objectifs de la première partie du cours sont d'initier les étudiants aux contraintes de l'informatique mobile aussi bien au niveau de la connexion réseau qu'au niveau applicatif. Les objectifs de la seconde partie sont d'introduire une réflexion sur les contraintes amenées par le temps réel sur les aspects fondamentaux des systèmes d'exploitation.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu Dans la première partie, seront étudiés les réseaux mobiles avec infrastructures (type GSM ou WiFi) ainsi que les réseaux sans fils peer-to-peer (type bluetooth). Les systèmes informatiques temps réel sont utilisés pour assurer une gestion efficace des événements en provenance de l'environnement. Le temps consommé par les traitements internes doit être le plus court possible mais aussi garanti ; une réponse tardive est une réponse fautive ! Cette propriété induit une révision des mécanismes fondamentaux des systèmes d'exploitation (gestion des processus, prise en compte des interruptions, partage de ressources et exclusion mutuelle, synchronisation, entrées/sorties, gestion du temps...). En particulier il y a lieu de focaliser sur les propriétés des algorithmes d'ordonnancement.

Nom du responsable Philippe MARQUET et David SIMPLOT

Intitulé de l'UE SYNTHÈSE D'IMAGES

Identifiant SI

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Acquérir les fondements de la modélisation et de la visualisation des mondes virtuels 3D.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques Polycopié disponible, support de cours, domaine bien couvert sur le web.

Évaluation de la charge enseignant

Description du contenu

- Introduction et rappels : repère 3D, coordonnées homogènes, positionnement, intersection, localisation...
- Courbes et surfaces : hermites, béziers, b-splines, nurbs, implicite.
- Volumes et Brepes polyédriques (généralités + winged-edges).
- Modèle d'éclairage (couleur, phong).
- Rendu projectif (élimination parties cachées, interpolation...).
- Autres méthodes de rendu : lancer de rayons, radiosité.
- Textures (constructions, placages, bump, light...).
- Animation et réalité virtuelle : sensibilisation aux problèmes de la définition du mouvement, de la collision et de l'interaction.

Nom du responsable Fabrice AUBERT

Intitulé de l'UE PROGRAMMATION 3D

Identifiant P3D

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Pratiquer la programmation dédiée à la visualisation de scènes 3D. Le contenu se focalise principalement sur l'exploitation de la librairie OpenGL. Le langage de description de scènes 3D VRML/X3D et l'affichage de graphes de scènes par Java3D sont également introduits dans cette UE.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques support de cours.

Évaluation de la charge enseignant

Description du contenu À part la partie « Graphes de scène », tout le contenu est abordé dans le contexte OpenGL.

- Présentation générale d'OpenGL : principe de visualisation projective, polygones, coordonnées, mise en oeuvre avec librairie événementielle.
- Positionnement : caméra, changement de repère, conception avec les piles de matrices.
- Visualisation des polygones : contrôle de l'éclairage, normales, gouraud, orientation, depth buffer.
- Pipeline au niveau pixel : tests, stencil, blending; illustrations : ombre, coupe, réflexion, transparence.
- Texture : par coordonnées, par projection, par environnement.
- Pixel Shaders, Vertex Shaders et effets spéciaux.
- Graphes de scène : description par VRML/X3D; affichage par Java3D.

Nom du responsable Fabrice AUBERT

Intitulé de l'UE PROGRAMMATION PARALLÈLE

Identifiant PP

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Le cours est une introduction à la programmation parallèle orientée vers la manipulation des outils actuels ; des rudiments d'algorithmiques parallèles sont introduits.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu Nous traitons des

- paradigmes de programmation parallèle (parallélisme de tâches, parallélisme de données, mémoire partagée, communication par message) ;
- éléments d'algorithmique parallèle ;
- outils et environnements de la programmation parallèle : pthreads, OpenMP, MPI, DPCE ;
- expérimentation sur machines parallèles.

Nom du responsable Philippe MARQUET

Intitulé de l'UE BIOINFORMATIQUE

Identifiant BI

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs L'objectif de ce cours est de montrer comment l'informatique permet l'analyse à grande échelle de données biologiques, comme des séquences d'ADN. L'enseignement commence avec des rappels approfondis de biologie, pour entrer dans le vif du sujet. Il se poursuit par la présentation des techniques bioinformatiques couramment utilisées, avec des aspects informatiques fondamentaux. Il est assuré en tandem par un biologiste et un informaticien.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

Introduction à la biologie moléculaire

- l'organisation de la cellule
- l'ADN et l'information génétique
- les protéines
- les organismes vivants

Les données de la bioinformatique

- les programmes de séquençage
- les banques de données
- les puces à ADN

Outils informatiques d'analyse de séquences

- algorithmes d'alignement de séquences (programmation dynamique)
- algorithmes d'alignement multiple (heuristiques, algorithmes génétiques)
- recherche de motifs
- phylogénie

Les TP sont l'occasion d'appliquer les techniques d'analyse de séquences à des exemples biologiques "en conditions réelles" : comment localiser un gène par des techniques de bioinformatique, comment déterminer sa fonction...

Nom du responsable Hélène TOUZET et Patrice CHAGNAUD

Intitulé de l'UE PRINCIPES ET ALGORITHMES CRYPTOGRAPHIQUES

Identifiant PAC

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Donner aux étudiants un aperçu des principes et des algorithmes utilisés en cryptographie contemporaine. L'étudiant sera finalement à même de comprendre et analyser, voire critiquer, les systèmes de chiffrement, signature... disponibles sur le marché.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu La première partie du cours traite essentiellement des algorithmes classiques reposant sur les principes de substitution et de transposition (Jules César, Vigenère...), puis sur la présentation de l'unique procédé de chiffrement inconditionnellement sûr : le masque jetable. Ce dernier nécessite la génération d'une suite aléatoire, mais dans la pratique nous devons nous contenter de suites pseudo-aléatoires ; nous nous intéressons alors aux critères que doit vérifier un tel générateur pour que le système résiste aux attaques connues.

Nous avons besoin de manipuler certains objets mathématiques, comme les corps finis. Nous nous intéressons donc à ces structures, en soulignant les aspects algorithmiques de leur construction. Ces corps sont utilisés dans la présentation de schémas de chiffrement à clef secrète (génération de pseudo-alea pour le chiffrement à flot, structure des substitutions dans l'AES) ou à clef publique (RSA...).

La cryptographie à clef publique, relativement récente, repose sur la difficulté de résolution de problèmes mathématiques complexes (factorisation de grands entiers, calcul du logarithme discret). Outre le chiffrement, elle permet de faciliter l'échange de clefs, et permet de signer les documents. Néanmoins, ce concept soulève un nouveau problème, celui de la certification des clefs publiques.

On abordera également les notions de partage de secret, et de protocoles d'identification (notamment les protocoles dits « zero knowledge », qui permettent de prouver que l'on connaît bien un secret sans pour autant révéler quoi que ce soit à son sujet).

Nom du responsable Caroline FONTAINE et Éric WEGRZYNOWSKI

Intitulé de l'UE BASES DE DONNÉES ET ASPECTS CLIENT-SERVEUR

Identifiant BDCS

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Comprendre comment fonctionne une application base de données client serveur.
Apprendre quelques notions élémentaires d'administration.

Volume étudiant Deux heures de cours/TD et deux heures de travaux pratiques par semaine sur 13 semaines

Volume de travail personnel étudiant estimé Environ 30 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- construction d'applications C/S : (Oracle est toujours serveur)
- utilisation d'ODBC avec (par exemple) Access en client
- utilisation de JDBC avec le développement d'une application java
- utilisation d'un outils dédié (JDeveloper ou Developer 2000)
- administration
 - gestion des utilisateurs et des droits
 - notion de transaction, de contrôle de la concurrence
 - stockage et accès aux données, optimisation
 - reprise après panne

Nom du responsable Anne-Cécile CARON et Mireille CLERBOUT

Intitulé de l'UE MODÉLISATION DES SYSTÈMES COMPLEXES

Identifiant MSC

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs L'analyse de performances des architectures client-serveur, le dimensionnement des réseaux de communication, l'étude de la fiabilité, de la maintenabilité et de la disponibilité des systèmes complexes (vols spatiaux, centrales nucléaires etc.) repose en grande partie sur la théorie des probabilités, en particulier des chaînes de Markov ou des réseaux de Pétri stochastiques.

L'analyse d'une liste d'évènements repose sur la théorie des automates revisitée sous un angle essentiellement pratique.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- Chaînes de Markov en temps discret et en temps continu :
 - équations d'état,
 - distribution stationnaire,
 - temps moyen de séjour dans un état, délai d'absorption,
 - ergodicité,
 - « transformée en z » et transformée de Laplace,
 - méthode de simulation sur ordinateur.
- Application à l'étude des files d'attente et à l'étude des architectures client-serveur.
- Application au calcul des indicateurs de la sûreté de fonctionnement d'un système complexe : fiabilité, maintenabilité et disponibilité.
- Initiation aux réseaux de Pétri.
- Méthode de dénombrement d'objets décrits par une grammaire en utilisant la technique des « séries génératrices ».

Les notions qui sont introduites dans ce module sont réutilisables dans d'autres disciplines comme la bioinformatique, la théorie de l'information et de l'apprentissage, les calculs pratiques de complexité.

Nom du responsable Michel PETITOT

Intitulé de l'UE HISTOIRE ET ÉPISTÉMOLOGIE DU CALCUL ET DE L'INFORMATIQUE

Identifiant HÉCI

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs

- Présenter les moments importants de l'histoire de l'informatique.
- Montrer que le développement de l'informatique produit des outils ayant un contenu philosophique intéressant (théorie de la complexité par exemple) et pose des problèmes délicats (l'intelligence artificielle par exemple).

Volume étudiant Deux séances de 2h de cours/TD par semaine sur 13 semaines.

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- Histoire du calcul et de l'informatique
 - Instruments de calcul anciens.
 - De Pascal à Babbage.
 - La mécanographie.
 - La naissance du calcul électronique, l'Eniac, Turing et Enigma...
- Problèmes philosophiques liés à l'IA, aux théories du calcul, de l'information et de la complexité

Thèmes possibles parmi d'autres.

Discussions autour de l'intelligence artificielle, le test de Turing, les sciences cognitives et le computationnalisme. L'objection de Lucas. Penrose.

Effectivité et praticabilité. Thèse de Church (diverses versions). Classes de complexité, notion de faisabilité. Modèles de calcul classique et quantique. Franchissement de la barrière de Turing. Rapports avec la physique.

La théorie du calcul et la logique (le concept de système formel, l'opposition vrai/démonstrable, l'indécidabilité et son interprétation, la notion de modèle, etc.)

Réflexion sur la notion de démonstration automatique : une démonstration faite par ordinateur et trop complexe pour être vérifiée par l'homme est-elle valide ?

La théorie algorithmique de l'information et son utilisation en épistémologie (le principe du rasoir d'Occam), en physique (entropie, calculs réversibles, etc.), en philosophie des mathématiques (les nombres oméga de Chaitin).

La notion de hasard en informatique (les générateurs pseudo-aléatoires, les générateurs pour la cryptographie, les suites aléatoires au sens de Martin-Löf).

Nom du responsable Jean-Paul DELAHAYE

Intitulé de l'UE ERGONOMIE DE CONCEPTION DES IHM ET USAGES

Identifiant IHM

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Acquisition de connaissances en ergonomie cognitive pour la conception d'IHM. Acquérir des méthodes de conception basées sur les usages existants et futurs. Maîtriser les méthodes d'évaluation ergonomiques.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- Notions de base de la psychologie ergonomique : tâche, activité, usage, affordance, connaissances....
- Bases du fonctionnement humain : mémoire, perception, motricité, apprentissage, connaissances
- Méthode d'analyse des usages pour la conception
- Méthode de conception
- Méthode d'évaluation

Nom du responsable Patricia PLÉNACOSTE et Fabrice AUBERT

Intitulé de l'UE SPÉCIFICATION ET VÉRIFICATION DU LOGICIEL

Identifiant SVL

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis Aucun (licence mention informatique)

Parcours Optionnel en première année du master science mention informatique

Objectifs Sensibiliser à l'utilisation des méthodes formelles (spécification et vérification de programmes) pour l'amélioration de la qualité du logiciel. Présenter les notions théoriques sous-jacentes et les mettre en pratique principalement en utilisant des outils existants de spécification et vérification pour Java.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- Programmation avec des invariants : logique de Hoare, mise en pratique de la programmation par contrat en Java (ex : outils liés à JML, Jass), utilisation de génération automatique d'obligation de preuves et d'un assistant de preuve pour les résoudre, raffinements.
- Test : généralités (place dans le cycle de développement, techniques de test), mise en pratique du test unitaire en Java (ex : Junit).
- Vérification automatique : systèmes de transition, logique temporelle, algorithmes de "model-checking", mise en pratique sur des programmes Java.

Nom du responsable Mirabelle NEBUT

Intitulé de l'UE ARCHITECTURE EXPERTE

Identifiant AEX

Proposée par UFR IEEA - INFORMATIQUE

Nombre de crédits 5

Pré-requis (Licence mention informatique). Architecture évoluée (AEV).

Parcours Optionnel en première année du master science mention informatique

Objectifs Le but de ce cours est de présenter les techniques les plus avancées en architecture des ordinateurs utilisées dans les microprocesseurs actuels : l'architecture EPIC pour les processeurs évolués, analyse des systèmes SoC pour les architectures spécialisées.

Volume étudiant Un cours et une séance de travaux dirigés par semaine, 6 séances de travaux pratiques réparties sur 13 semaines

Volume de travail personnel étudiant estimé Environ 50 heures

Contrôle et validation des connaissances Un examen final et un contrôle continu (interrogation, compte-rendu de travaux pratiques ou mini-projets).

Accompagnement et outils pédagogiques

Évaluation de la charge enseignant

Description du contenu

- Processeur
 - Traitement désordonné
 - Multi threadé
 - VLIW
 - Vectoriel
- Machine à mémoire partagée
 - Architecture SMP
 - Protocole d'accès mémoire
 - Cohérence de cache (Snoop, directory)
- Cluster
 - Réseaux
 - Routage
- Evaluation de performances
- Système sur puce
 - Composants
 - FPGA, ASIC
 - Notion de codesign
 - Vérification et simulation

Nom du responsable Jean-Luc DEKEYSER
