

# CAMPAGNE D'HABILITATION CONTRAT 2004-2007

## **1 - FICHE D'IDENTITÉ**

Établissement : **UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS**

Composante principale : UFR Sciences

Composante associée :

Établissement co-habilité :

Domaine de Formation: **Sciences et Technologies**

Intitulé du diplôme : **LICENCE**

Mention : **INFORMATIQUE**

N° d'habilitation :

Secteurs de référence (code sise) :

Localisation des enseignements :

Université de Nice-Sophia Antipolis  
UFR Sciences, Parc Valrose  
06 108 Nice Cedex 2

Date d'ouverture de la formation : septembre 2004

Responsable de la formation :

Carine Fédèle, maître de conférences 27e section CNU  
tél : 04 92 07 66 69, fax : 04 92 07 66 65  
Carine.Fedele@unice.fr

Place de la formation dans l'offre globale de formation :

Voir le document général de l'UFR Sciences de l'UNSA.

Place de la formation dans la carte régionale/nationale des formations :

Voir le document général de l'UFR Sciences de l'UNSA.

Date et avis du CEVU

Date et avis du CA

## Petit explicatif

Les licences du secteur scientifique (hors licences professionnelles) sont regroupées sous une seule appellation (qui correspond en fait au domaine) : la *Licence Sciences et Technologies*, assortie de mentions mono-disciplinaires. Les mentions proposées par l'UFR Sciences sont :

- Mathématiques
- Informatique
- Mathématiques Appliquées aux Sciences Sociales (MASS)
- Méthodes Informatiques Appliquées à la GEstion (MIAGE – IUP)
- Physique
- Électronique Électrotechnique et Automatique (EEA)
- Chimie
- Sciences de la Vie
- Sciences de la Terre (Géosciences – Géorisques – Géoressources (3G))

Les différents parcours permettant de converger vers ces mentions sont d'autant plus communs que l'on se trouve dans les premiers semestres (S1 à S3, voire S4). En effet, pour permettre un suivi pédagogique efficace et des économies de moyen, des regroupements semestriels sont prévus pour les mentions de Licence présentant des contenus voisins ou adjacents.

Il est à noter que des UEs de sensibilisation pour l'orientation vers les licences professionnelles sont prévus en semestre 4. De plus, dans un cadre coopératif IUT/UFR Sciences, l'IUT met également en place une sensibilisation à la poursuite d'études scientifiques longues (Licence d'Informatique et Licence EEA).

Afin de proposer une lecture plus simple, mettant en évidence les parcours communs et pour éviter aussi la répétition des descriptions complètes de ces premiers semestres communs dans chacune des mentions les contenant, l'UFR Sciences a opté pour la présentation qui suit : un dossier général pour la Licence Sciences et Technologies, qui donne une description précise des quatre premiers semestres pour chacun des regroupements ; une maquette par mention (celle-ci pour la Licence d'Informatique) dans laquelle, par une approche inverse, les quatre premiers semestres ne seront décrits que dans leurs spécificités vis-à-vis de la mention décrite, suivis d'une description complète de l'année diplômante et de ces relations potentielles avec les Masters locaux.

## **2. ORGANISATION DE LA FORMATION**

### **2.1 Publics concernés**

#### **2.1.1 Effectifs attendus**

Tableau récapitulatif des effectifs depuis 1999–2000 du parcours habilité en 2000 DEUG MIAS-MI et Licence d'Informatique dont cette Licence prend la suite.

<i>année</i>	<i>nombre d'inscrits</i>		
	<i>DEUG MI1</i>	<i>DEUG MI2</i>	<i>Licence d'Informatique</i>
1999-2000	272	122	77
2000-2001	385	117	75
2001-2002	375	172	68
2002-2003	343	164	94
2003-2004	277	176	86

On peut estimer que les effectifs se stabilisent et que ceux de la première année (L1) seront aux alentours des 300, ceux de la deuxième année (L2) près des 200 et ceux de la troisième année (L3) près des 100.

L'origine des étudiants de L1 est en très grande majorité des titulaires d'un baccalauréat scientifique. Arrivent en L2 des élèves venant des classes préparatoires.

### 2.1.2 Objectifs

Le premier objectif de la licence est d'apporter une formation solide et équilibrée dont les contenus sont en prise directe avec l'évolution des sciences et des techniques et exploitent toute la richesse des démarches scientifiques.

La formation que l'on souhaite dispenser développe consciemment trois niveaux de connaissance :

- Le niveau «modèles» : ce niveau doit permettre de développer des réflexes d'abstraction, de généralisation et de réutilisation liés à une attitude et une pratique de recherche scientifique et technologique. Dans un domaine à évolution «ultra-rapide», qu'est l'informatique, il faut songer, dans l'intérêt des étudiants, à pérenniser la formation que nous proposons.

La première façon de réaliser cela est de miser sur la formation scientifique. Avant d'être un «programmeur», l'étudiant en Licence d'Informatique doit posséder un savoir et un raisonnement scientifique. Il devra comprendre et maîtriser les modèles propres à la discipline informatique mais aussi, car cette discipline est aussi et souvent une discipline outil, quelques modèles mathématiques et physiques de représentations du monde réel. Il s'agit d'ancrer son savoir ailleurs que dans des connaissances très rapidement périssables comme peuvent l'être par exemple les langages de programmation ou les outils à la mode.

C'est pourquoi les étudiants ont des UEs de mathématiques obligatoires pendant les deux premières années, ainsi que des UEs d'informatique théorique tout le long des trois années.

La deuxième façon de former les étudiants «à long terme» est de les doter de réflexes d'auto-formation. Ces méthodes sont travaillées lors des projets et des travaux d'études.

- Le niveau «concepts» : ce niveau montre notre souci de conserver un lien entre les fondements théoriques et les outils. L'informatique est une science jeune où tous les mécanismes d'abstraction n'ont pas encore forcément convergé vers des modèles. Les concepts permettent d'introduire des champs de connaissances où l'on ne dispose pas encore de modèles pertinents. Ils sont souvent très liés au monde industriel.

Il est le pré-requis «minimum» de l'existence de tout enseignement.

- Le niveau «outils» : ce niveau doit permettre de concrétiser les modèles et concepts introduits. Les outils ne sont pas la partie la moins noble d'un enseignement. Ils sont les moyens de mise en œuvre d'une réflexion préalable. Les outils que nous avons choisi d'enseigner répondent à deux critères majeurs (qui ne sont d'ailleurs pas exclusifs) :

- ils permettent d'illustrer au mieux des concepts ou des modèles ;
- ils ont une véritable existence dans le monde industriel.

Nous nous efforçons de travailler avec des logiciels libres, ce qui nous assure l'indépendance nécessaire et place nos étudiants dans un courant de pensée que nous soutenons car nous pensons qu'il est l'avenir de l'informatique.

### 2.1.3 Débouchés

Étant donné l'existence à l'UNSA d'une Licence Professionnelle permettant à l'étudiant d'entrer dans la vie active après son diplôme, la Licence d'Informatique n'a pas pour débouché premier le monde industriel. En fait, le parcours logique d'un étudiant ayant obtenu une Licence d'Informatique est de poursuivre ses études dans un Master (national ou étranger), et plus particulièrement le Master STIC proposé par l'UNSA. La première année est pour l'instant à Nice (Valrose) et la deuxième année à Sophia Antipolis, mais à terme les deux années seront situées à Sophia Antipolis grâce à la création du campus STIC, et profiteront de la proximité d'entreprises et de laboratoires de recherche. Le Master STIC a un projet de formation parfaitement synchrone avec les attentes du monde industriel et universitaire en offrant un éventail large de spécialités recherche et professionnelles dans les domaines informatiques.

Toutefois, il ne faut pas perdre de vue que certains étudiants veulent aller dans le monde industriel après une Licence non Professionnelle pour des raisons variées. Jusqu'à présent, cela a été exceptionnel. Cependant, il faut rester conscient que cela pourrait devenir plus fréquent. Il est hors de question de laisser des étudiants en situation

d'échec et c'est pourquoi :

- nous encourageons les programmes d'échange ERASMUS ;
- nous favorisons la connaissance du monde de l'entreprise avec une incitation aux stages en fin de licence ;
- et enfin nous veillons à ce que l'enseignement dispensé, tenant compte des aspects technologiques les plus modernes, permette un réel débouché industriel.

Les parcours donnent donc certaines compétences pratiques pouvant permettre une insertion rapide en entreprise.

Certains étudiants peuvent aussi passer des concours administratifs à l'issue de L2 ou de L3, ou vouloir se diriger vers une carrière d'enseignants. Un parcours à coloration mathématiques a été mis en place pour aider les étudiants désireux d'aller à l'IUFM, afin de préparer les concours pour entrer dans l'enseignement primaire et secondaire.

Après les 120 premiers crédits, les étudiants peuvent poser leur candidature au concours d'entrée dans des écoles d'ingénieurs dont l'École Supérieure en Sciences Informatiques (ESSI) de l'UNSA. Cela est aussi envisageable après l'obtention des 180 crédits, donc de la Licence (les étudiants «perdant» une année). Pour l'instant, il n'y a aucun changement national prévu pour les écoles d'ingénieurs françaises. Le LMD risque de modifier leur manière de fonctionner et donc de recruter. C'est à voir.

## **2.2 Choix pédagogiques : les parcours**

### **2.2.1 Conception des parcours**

Au semestre 4 (UE L2O), les étudiants peuvent choisir « Internet et programmation Web » et se donner ainsi la possibilité de s'inscrire à l'issue de L2 (120 premiers crédits) en Licence Professionnelle Systèmes Informatiques et Logiciels - Métiers de l'Informatique (SIL-MI) délivrée conjointement par l'IUT de Nice Côte d'Azur, l'UFR Sciences et l'ESSI de l'UNSA. Cette inscription n'est pas automatique et se fait sur dossier : un jury validera l'entrée de l'étudiant dans cette Licence Professionnelle.

### **2.2.2 Préciser les parcours types**

La Licence d'Informatique a trois parcours types :

1. parcours I : Informatique,
2. parcours IM : Informatique et mathématiques,
3. parcours IMT : Informatique, microélectronique et télécommunications.

Ces trois parcours ont des UEs obligatoires, la différenciation portant sur une soixantaine d'heures (au maximum) sur les deux derniers semestres (semestres 5 & 6).

Dans le parcours IM, l'architecture des ordinateurs ainsi que l'option du semestre 5 sont remplacés par une majeure de mathématiques (probabilités ou algèbre bilinéaire). Quant au semestre 6, une mineure de mathématiques (analyse numérique) est ajoutée, les matières compilation et environnements de programmation étant diminuées (nous avons jugé nécessaire qu'un étudiant de Licence d'Informatique ait des connaissances minimum dans ces matières-là).

Dans le parcours IMT, l'option du semestre 5 devient une UE d'électromagnétisme. Quant au semestre 6, une UE ondes et composants est ajoutée, les matières compilation et environnements de programmation étant diminuées (même remarque que pour la constitution du parcours IM), et le travail d'études supprimé.

### **2.2.3 Organisation de la formation - contenus**

Pour les effectifs, à l'UFR, les cours en amphithéâtre se font devant 200 étudiants au maximum, les travaux dirigés avec 40 étudiants et les travaux pratiques avec 20 étudiants.

Comme dit précédemment, les parcours ne se différencient qu'en dernière année. Dans un premier tableau appelé tronc commun, apparaissent donc les quatre premiers semestres. Viendront ensuite les trois tableaux correspondant aux parcours (semestres 5 & 6).

**Tronc commun:** aux trois parcours (I, IM et IMT)

<i>Semestre/UE</i>	<i>Coef- ficient</i>	<i>ECTS</i>	<i>Contenu des enseignements</i>	<i>CM</i>	<i>TD</i>	<i>TP</i>	<i>Travail personnel libre</i>	<i>Durée totale</i>
<b>Semestre 1</b>								
<b>UE obligatoires</b>								
L1M1	7,5	7,5	Fonctions d'une variable	26	39		65	130
L1I1	7,5	7,5	Introduction à la programmation	18	23	24	65	130
L1P	5	5	Physique	13	26		39	78
L1C	5	5	Chimie	13	26		39	78
L1E1	2,5	2,5	EEA	13	13		26	52
<b>Sur liste</b>								
L1O1	2,5	2,5		13	13		26	52
<b>Total semestre 1</b>	30	30		96	140	24	260	520
<b>Semestre 2</b>								
<b>UE obligatoires</b>								
L1M2	7	7	Algèbre linéaire et géométrie	18	48		66	132
L1M3	5	5	Mathématiques discrètes	18	24		42	84
L1I2	7	7	Programmation et structures de données	18	24	24	66	132
L1I3	5	5	Systèmes informatiques	18		24	42	84
L1E2	4	4	EEA	12	12	18	42	84
<b>Sur liste</b>								
L1O2	2	2		12	12		24	48
<b>Total semestre 2</b>	30	30		96	120	66	282	564
<b>Semestre 3</b>								
<b>UE obligatoires</b>								
L2M1	6	6	Algèbre	18	36		54	108
L2M2	6,5	6,5	Intégrales et séries	18	36		54	108
L2I1	6	6	Informatique théorique	24	24		48	96
L2I2	6,5	6,5	Programmation impérative	24		36	60	120
L2E	3	3	EEA	12		12	24	48
L2LV	2	2	Langue vivante		24		24	48
<b>Total semestre 3</b>	30	30		96	120	48	264	528
<b>Semestre 4</b>								
<b>UE obligatoires</b>								
L2M3	6,5	6,5	Algèbre	18	36		54	108
L2M4	2,5	2,5	Statistiques	12	12		24	48
L2I3	6,5	6,5	Algorithmique	24	36		60	120
L2I4	4	4	Programmation fonctionnelle	24		24	48	96
L2I5	2,5	2,5	Bases de données	12		12	24	48
L2P	2,5	2,5	Physique pour informaticiens	12	12		24	48
<b>Sur liste</b>								
L2O	5,5	5,5		26	26		52	104
<b>Total semestre 4</b>	30	30		128	122	36	286	572
<b>Total DEUG MI</b>	120	120		416	502	174	1092	2184

**Parcours n° I: Informatique**

<i>Semestre/UE</i>	<i>Coef- ficient</i>	<i>ECTS</i>	<i>Contenu des enseignements</i>	<i>CM</i>	<i>TD</i>	<i>TP</i>	<i>Travail personnel libre</i>	<i>Durée totale</i>
<b>Total DEUG MI</b>	120	120		416	502	174	1092	2184
<b>Semestre 5</b>								
<b>UE obligatoires</b>								
L3I1	6,5	6,5	Optimisation combinatoire	24	36		60	120
L3I2	6,5	6,5	Langages et automates	24	24	12	60	120
L3I3	5,5	5,5	Concepts fondamentaux des bases de données	24		24	48	96
L3I4	5,5	5,5	Architecture des ordinateurs	24		24	48	96
L3LV	3,5	3,5	Langue vivante		30		30	60
<b>Sur liste</b>								
L3O	2,5	2,5			24		24	48
<b>Total semestre 5</b>	30	30		96	114	60	270	540
<b>Semestre 6</b>								
<b>UE obligatoires</b>								
L3I5	8	8	Compilation 1	24	24	24	72	144
L3I6	8	8	Systèmes et réseaux	32	18	22	72	144
L3I7	5,5	5,5	Programmation objet	24		24	48	96
L3I8	6,5	6,5	Environnements de programmation et éléments de génie logiciel	24	12	24	60	120
L3I9	2	2	Travail d'études		18		18	36
<b>Total semestre 6</b>	30	30		104	72	94	270	540
<b>Total Licence</b>	180	180		616	688	328	1632	3264

**Parcours n° IM: Informatique et mathématiques**

<i>Semestre/UE</i>	<i>Coef- ficient</i>	<i>ECTS</i>	<i>Contenu des enseignements</i>	<i>CM</i>	<i>TD</i>	<i>TP</i>	<i>Travail personnel libre</i>	<i>Durée totale</i>
<b>Total DEUG MI</b>	120	120		416	502	174	1092	2184
<b>Semestre 5</b>								
<b>UE obligatoires</b>								
L3I1	6,5	6,5	Optimisation combinatoire	24	36		60	120
L3I2	6,5	6,5	Langages et automates	24	24	12	60	120
L3I3	5,5	5,5	Concepts fondamentaux des bases de données	24		24	48	96
L3LV	3,5	3,5	Langue vivante		30		30	60
<b>Sur liste</b>								
L3M1	8	8		24	48		72	144
<b>Total semestre 5</b>	30	30		96	138	36	270	540
<b>Semestre 6</b>								
<b>UE obligatoires</b>								
L3I5a	5,5	5,5	Compilation 1: construction d'une partie avant	16	16	16	48	96
L3I6	8	8	Systèmes et réseaux	32	18	22	72	144
L3I7	5,5	5,5	Programmation objet	24		24	48	96
L3I8a	4	4	Éléments de génie logiciel	18	9	9	36	72
L3I9	2	2	Travail d'études		18		18	36
L3M2	5	5	Analyse numérique	21	24		45	90
<b>Total semestre 6</b>	30	30		111	85	71	267	534
<b>Total Licence</b>	180	180		623	725	281	1629	3258

**Parcours n° IMT: Informatique, microélectronique et télécommunications**

Semestre/UE	Coef- ficient	ECTS	Contenu des enseignements	CM	TD	TP	Travail personnel libre	Durée totale
<b>Total DEUG MI</b>	120	120		416	502	174	1092	2184
<b>Semestre 5</b>								
<b>UE obligatoires</b>								
L3I1	6,5	6,5	Optimisation combinatoire	24	36		60	120
L3I2	6,5	6,5	Langages et automates	24	24	12	60	120
L3I3	5,5	5,5	Concepts fondamentaux des bases de données	24		24	48	96
L3I4	5,5	5,5	Architecture des ordinateurs	24		24	48	96
L3LV	3,5	3,5	Langue vivante		30		30	60
L3E1	2,5	2,5	Électromagnétisme – circuits	12	12		24	48
<b>Total semestre 5</b>	30	30		108	102	60	270	540
<b>Semestre 6</b>								
<b>UE obligatoires</b>								
L3I5a	5,5	5,5	Compilation 1: construction d'une partie avant	16	16	16	48	96
L3I6	8	8	Systèmes et réseaux	32	18	22	72	144
L3I7	5,5	5,5	Programmation objet	24		24	48	96
L3I8a	4	4	Éléments de génie logiciel	18	9	9	36	72
L3E2	7	7	Ondes et composants	30	30		60	120
<b>Total semestre 6</b>	30	30		120	73	71	264	528
<b>Total Licence</b>	180	180		644	677	305	1626	3252

**Enseignements au choix sur liste définie par l'équipe pédagogique**

Intitulé	Crédits	Parcours-type n°	Commun aux UE du parcours n°
L2O: EEA	5,5	I, IM, IMT	Licence EEA
L2O: Probabilités	5,5	I, IM, IMT	Licence de Mathématiques
L2O: Internet et programmation Web	5,5	I, IM, IMT	
L2O: Méthodes numériques et méthodes formelles	5,5	I, IM, IMT	Licence de Mathématiques
L3O: Prolog	2,5	I	
L3O: Théorie des graphes	2,5	I	
L3O: Algorithmes pour le traitement d'images	2,5	I	
L3O: Programmation pour Internet	2,5	I	
L3O: Administration systèmes - réseaux	2,5	I	
L3M1: Probabilités	8	IM	Licence de Mathématiques
L3M1: Algèbre bilinéaire	8	IM	Licence de Mathématiques

## 2.2.4 Passerelle et réorientation

Les étudiants de L1 et L2 peuvent à tout moment basculer vers la Licence de Mathématiques moyennant des aménagements minimes jusqu'au semestre 3 inclus. À partir du semestre 4, cela dépend de l'option qu'ils choisiront.

## 3. ÉQUIPE DE FORMATION

Équipe pédagogique (ordre alphabétique)

<i>Nom des enseignants</i>	<i>Grade (section CNU)</i>
BAUDE-DREYSSE Françoise	MC 27e
BEAUQUIER Bruno	MC 27e
BELLEUDY Cécile	MC 61e
BUFFA Michel	MC 27e
COLLET Philippe	MC 27e
CRESCENZO Pierre	MC 27e
DALLE Olivier	MC 27e
FORMENTI Enrico	PR 27e
FÉDÈLE Carine	MC 27e
GAFFÉ Daniel	MC 61e
GRIN Richard	PRAG Maths
HIRSCHOWITZ André	PR Maths
JULIA Sandrine	MC 27e
KOUNALIS Emmanuel	PR 27e
LANTERI Henri	PR 61e
LECARME Olivier	PR 27e
MADY Franck	MC 63e
MALLET Frédéric	MC 27e
MENEZ Gilles	MC 61e
MERLE Michel	PR Maths
MIGLIACCIO Claire	MC 63e
PIERRE Laurence	PR 27e
ROUSSEAU Roger	MC 27e
ROY Jean-Paul	PRAG Maths
SILICI Paul	MC Maths

## 4. LES PARTENARIATS

### 4.1 Partenariats industriels et académiques français et étrangers

Dans une Licence non Professionnelle et à ce stade du cursus étudiant, il ne paraît pas anormal de ne pas avoir de partenariat industriel, même dans notre discipline.

En revanche, grâce à Erasmus, nous avons des partenariats avec des universités européennes : Université de

Mannheim (Allemagne), Blekinge Institute of Technology (Karlskrona, Suède) et Université d'Athènes (Grèce). Dans le passé, nous comptons chaque année une poignée d'étudiants d'Édimbourg (Écosse) dans notre formation (surtout au niveau licence). En revanche, aucun de nos étudiants niçois ne partait (ils semblent bien sédentaires). Nous encourageons vivement ces départs et allons renforcer notre action.

L'Université de Nouméa (Nouvelle Calédonie) n'ayant pas de Licence ni de Maîtrise d'Informatique, nous avons conclu un accord avec le Département d'Informatique de cette université afin d'accepter leurs étudiants automatiquement après l'obtention des 120 premiers crédits (études scientifiques mathématiques et informatique).

## 4.2 Partenariat avec les établissements étrangers d'origine des étudiants

La plupart de nos étudiants étrangers viennent de Mauritanie (Nouakchott), avec une bourse délivrée par leur gouvernement pour leurs années d'études en France. Un accord inter-universitaire existe entre l'Université de Nouakchott et l'UNSA. Une collaboration avec le Département de Mathématiques et Informatique de la Faculté des Sciences et Techniques de l'Université de Nouakchott pourrait être envisagée pour permettre à des étudiants mauritaniens d'études scientifiques à double compétence mathématiques et informatique de s'inscrire directement en troisième année de la Licence d'Informatique de l'UNSA.

## 5. CONTRÔLE DES CONNAISSANCES

Une politique générale de l'UFR sur les modalités de compensation et de capitalisation des UEs, ainsi que sur la mise en place de deux sessions rapprochées pour chaque semestre est en train d'être élaborée. Chaque Licence doit la suivre : il faut donc se référer au document général de l'UFR Sciences de l'UNSA.

Le contrôle continu est présent dans chaque matière et sa note contribuera à la note finale de l'UE pour au moins 20% (cela dépendra de chaque UE). Il n'y a pas de note éliminatoire dans l'ensemble du cursus.

L'anonymat des copies est respecté. Les examens seront plutôt des écrits, sauf pour l'UE de travail d'études de L3 (son but étant d'apprendre aux étudiants à présenter une synthèse à l'oral). Il y aura deux sessions (politique générale de l'UFR).

## 6. ÉVALUATION

### 6.1 De la formation : effectifs, taux de réussite, débouchés

Vraiment très peu de monde s'arrêtait après un DEUG. Dans le tableau qui suit ne sont données que les statistiques pour la Licence d'Informatique depuis 1999–2000.

<i>Année</i>	<i>Nombre d'inscrits</i>	<i>Nombre de diplômés</i>	<i>Taux de réussite</i>
1999-2000	77	57	74,00%
2000-2001	75	48	64,00%
2001-2002	68	48	70,60%
2002-2003	94	60	63,80%
2003-2004	86		

Comme dit précédemment, c'était vraiment exceptionnel qu'un étudiant s'arrêtât après une Licence. Est-ce que le LMD va favoriser l'entrée dans le monde industriel après un Bac+3 ? C'est à voir.

### 6.2 Des enseignements

Chaque semestre, tous les étudiants sont conviés, par l'intermédiaire d'un formulaire électronique, à proposer une évaluation des enseignements de la période qui vient de s'achever. Leur représentant propose, dans le cadre

de la réunion semestrielle, une synthèse orale de l'évaluation. Voici une ébauche de questionnaire :

<b>COURS</b>	<i>Nom de l'enseignant:</i>
Par rapport à votre attente, son contenu est	<input type="radio"/> très satisfaisant <input type="radio"/> satisfaisant <input type="radio"/> non satisfaisant
La manière dont le cours vous a été présenté vous a paru	<input type="radio"/> très claire <input type="radio"/> moyennement claire <input type="radio"/> peu claire
La prise de notes en cours était	<input type="radio"/> facile <input type="radio"/> possible <input type="radio"/> difficile
Le volume alloué à cette matière vous semble	<input type="radio"/> insuffisant <input type="radio"/> correct <input type="radio"/> trop important
L'adéquation de l'examen à l'évaluation de vos connaissances est	<input type="radio"/> forte <input type="radio"/> moyenne <input type="radio"/> faible
<b>TD/TP</b>	<i>Nom de l'enseignant:</i>
Les réponses de l'enseignant à vos questions étaient	<input type="radio"/> très satisfaisantes <input type="radio"/> satisfaisantes <input type="radio"/> non satisfaisantes
En général, les sujets des TD/TP vous ont semblé	<input type="radio"/> très instructifs <input type="radio"/> instructifs <input type="radio"/> peu instructifs
Les corrigés écrits ou oraux des exercices sont	<input type="radio"/> très clairs <input type="radio"/> clairs <input type="radio"/> peu clairs
La cohérence entre le cours et les TD/TP était	<input type="radio"/> forte <input type="radio"/> moyenne <input type="radio"/> faible
Les horaires des séances et les volumes horaires ont été	<input type="radio"/> respectés <input type="radio"/> à peu près respectés <input type="radio"/> rarement respectés
Si vous avez d'autres remarques ou précisions...	

## **7. SUPPLÉMENT AU DIPLÔME**

### **7.1 Programme détaillé du parcours suivi**

Les descriptions des enseignements de mathématiques, de physique et de chimie sont identiques à celles de la Licence de Mathématiques ou de Physique pour les années 1 et 2.

#### **7.1.1 Année 1**

##### **L1I1 : Introduction à la programmation**

Cet enseignement est le premier en informatique pour une bonne partie des étudiants. Il s'agit donc d'une initiation à l'informatique à l'aide de l'introduction à la programmation. Le modèle choisi est impératif.

Présentation de l'ordinateur, premiers éléments d'algorithmique. Conditions, boucles. Procédures, récursivité. Première approche de la résolution des problèmes.

##### **L1E1 : EEA**

Introduction à l'électronique analogique et numérique. Circuits et systèmes numériques : représentation des nombres entiers dans les systèmes numériques ; les systèmes numériques de base et leurs propriétés ; modélisation des circuits combinatoires ; synthèse de systèmes combinatoires simples. Réseaux électroniques : dipôles ; régime sinusoïdal.

##### **L1O1 : option d'approfondissement ou de réorientation**

##### **L1I2 : Programmation et structures de données**

**Pré-requis :** L1I1

Cet enseignement constitue la base de l'activité de programmation des années à venir.

Programmation impérative. Algorithmes fondamentaux : consultation de table, tri; structures chaînées, structures arborescentes, fichiers, graphes. Évaluation des performances. Validité des programmes. Types abstraits.

##### **L1I3 : Systèmes informatiques**

**Pré-requis :** L1I1

Cet enseignement est l'introduction à un système d'exploitation moderne.

Historique des systèmes d'exploitation et des réseaux. Structure d'un système, concepts fondamentaux. Processus, fichiers, langage de commande. Système de fenêtrage. Réseaux, protocoles, sécurité.

##### **L1E2 : EEA**

Introduction aux composants électroniques discrets. Circuits numériques (les systèmes séquentiels). Six séances de travaux pratiques.

##### **L1O2 : Options de découverte**

- **Les Limites de l'Informatique (ou les mauvaises nouvelles de l'informatique).** Cette option a pour but de présenter de façon simple les limites théoriques et pratiques de la discipline informatique. Les thèmes qui seront abordés tout au long de ce cours sont : les très mauvaises nouvelles (ou les limites théoriques de l'informatique) : il y a beaucoup de choses (problèmes) qu'un ordinateur ne pourra jamais faire ; les mauvaises

nouvelles (ou les limites pratiques de l'informatique) : il y a beaucoup de choses qu'un ordinateur ne pourra traiter en pratique ; Comment retourner les mauvaises nouvelles en des bonnes : les bases de la cryptographie et de la sécurité moderne.

- **Traitement d'images.** La représentation des images sous forme numérique permet non seulement leur conservation et leur transmission sous une forme inaltérable, mais aussi leur manipulation de manières très variées à l'aide d'outils informatiques. Il est par ailleurs possible de construire de nouvelles images, ou d'ajouter des parties artificielles à des images réelles. Tout cela a beaucoup d'applications dans de très nombreux domaines. Les deux séances de cours présenteront les techniques de représentation des images, et spécialement des couleurs, et les problèmes qu'elles posent. Les travaux pratiques qui suivront utiliseront le logiciel libre et gratuit Gimp, et se feront à partir de feuilles descriptives très détaillées.
- **Algorithmique des télécommunications.** Le but de cette option est de donner une idée de ce qu'est l'algorithmique des télécommunications. Le cours considérera des problèmes posés dans l'industrie de l'informatique ou des télécommunications. Après une modélisation du problème, on cherchera à dégager des solutions constructives (susceptibles d'être engendrées informatiquement ou construites *via* un algorithme) au moins sur des exemples simples. On verra que certains problèmes sont «simples» à résoudre alors que d'autres de formulation très voisine sont très compliqués. Comme exemple de problèmes qui pourront être considérés, on cherchera comment diffuser une information de manière rapide, comment répandre une rumeur, comment sortir d'un labyrinthe, comment communiquer vite en n'envoyant que peu de messages, comment construire un réseau embarqué dans un satellite qui doit amplifier des signaux avant de les renvoyer, même en cas de panne d'amplificateurs et comment dans certain cas ce problème est relié à trier des nombres, comment regrouper des trafics de petit débit dans des tuyaux de plus gros débit.

## 7.1.2 Année 2

### L2I1 : Informatique théorique

**Pré-requis :** L1M3

Ce cours d'informatique théorique présente une initiation au formalisme utilisé en informatique incluant au besoin les rappels mathématiques nécessaires. Il présente la dénombrabilité des ordinateurs, leur aspect symbolique et la notion centrale d'ordre. Il se poursuit par la présentation de la logique booléenne et du 1<sup>er</sup> ordre (raisonnement, applications de programmation logique). Il constitue également une première approche théorique de l'évaluation des performances des ordinateurs (équation de récurrence, complexité des algorithmes). Les propriétés des objets couramment manipulés dans les structures de données ou les modélisations sont largement étudiées. Il s'agit des mots, les listes, les langages, les arbres, les graphes, etc.

### L2I2 : Programmation impérative

**Pré-requis :** L1I2 – L1I3

Cet enseignement a pour objectif de rendre les étudiants opérationnels dans la construction de programmes travaillant à un niveau proche de celui de la machine : utilisation généralisée des pointeurs, programmation d'entrées-sorties, gestion dynamique de la mémoire, utilisation directe des ressources offertes par le système d'exploitation.

### L2E : EEA

Informatique industrielle : les mémoires ; notion de graphe d'état ; les systèmes séquentiels complexes. Acquisition de données : notions de base sur les amplificateurs opérationnels ; conversion numérique-analogique ; conversion analogique-numérique ; étude d'un système d'acquisition de données.

### L2I3 : Algorithmique

**Pré-requis :** L1I2 – L2I1

### **L2I4 : Programmation fonctionnelle**

**Pré-requis :** L1I2

### **L2I5 : Bases de données**

Construction d'un schéma de données. Structures physiques d'archivage des données. Définition des termes concept, modèle, méthode. Une méthode permettant de définir un modèle de données d'une application. Utilisation des formalismes de type entité-association. Bases de données relationnelles. Concepts de relation, attribut, table, domaine, contrainte d'intégrité, schéma. Algèbre relationnelle. Introduction à SQL.

## **7.1.3 Année 3**

### **L3I1 : Optimisation combinatoire**

**Pré-requis :** L1I2 - L2I1 - L2I3

Comment formaliser des problèmes d'optimisation combinatoire ? Chemin de la transformation d'un problème concret à un problème abstrait.

Comment résoudre les problèmes d'optimisation combinatoire ? Plusieurs stratégies sont présentées :

- stratégie glouton (« school timetable ») ;
- stratégie programmation dynamique (« World games ») ;
- stratégie exhaustive : génération des objets combinatoires (permutations, combinaisons, partitions, parties) ;
- stratégie approximative (« vertex-cover ») ;
- stratégie probabiliste (Monte-Carlo et Las Vegas).

Puis sont traités les graphes :

- algorithme générique du parcours d'un graphe et applications (points d'articulation, tri topologique, etc.) ;
- arbres de recouvrement minimaux (algorithme de Kruskal et Prim) ;
- chemins optimaux et ses applications à la bourse (algorithme de Bellman Ford, le problème du voyageur de commerce (approximatif)) ;
- flots et ses applications.

### **L3I2 : Langages et automates**

**Pré-requis :** L2I4

- Langages et automates (36h)

Cette partie a pour objet de débiter l'étude classique des langages formels en suivant la hiérarchie de Chomsky (langages réguliers, contextuels, non-contextuels, théorème de Kleene, lemme de l'étoile, etc.), des automates (fini, à pile, machine de Moore, machine de Turing, etc.) et du système de réécriture puissant que constituent les grammaires. Chaque notion est illustrée par une ou plusieurs applications empruntées aux domaines de la reconnaissance de motifs, la compression, la spécification des langages de programmation, la compilation, la calculabilité, l'électronique. D'autres applications (systèmes dynamiques discrets, codes auto-correcteurs, décodage du génôme, linguistique) y sont évoquées.

- Programmation applicative avancée et interprétation (24h)

Après un rappel sur la programmation multi-paradigmes en Scheme (fonctionnel, impératif, ordre supérieur, macros), on étudie les objets : implémentation sous forme de fonctions, puis utilisation de la couche objet à la Java de DrScheme et de son API pour programmer des interfaces graphiques. L'outil lex/yacc est utilisé pour traduire un langage à syntaxe infixée classique vers du Scheme. On implémente ensuite un interprète Scheme, à liaison statique ou dynamique, à environnement et sans continuations. À la fois fonctionnel (fonctions de 1<sup>e</sup> classe) et impératif (séquençement, affectation), en insistant sur les

fermetures et sur l'importance de letrec. Enfin, on étudie le contrôle, avec l'étude des continuations, soit par fermetures CPS, soit comme valeurs de première classe avec call/cc. On pourra implémenter les continuations dans l'interprète.

### **L3I3 : Concepts fondamentaux des bases de données**

#### **Pré-requis : L2I5**

Architecture ANSI/SPARC. Architecture client-serveur. Architecture 3-tiers. Évolution des SGBD. Modèles réseau et hiérarchique. Modèles relationnel et objets. Modèle relationnel-objet.

Le modèle Entités-Associations (E-A). Définitions. Entités. Attributs. Associations. Clés. Associations 1-1, 1-n, n-n. Associations multiples. Héritage. Relations « isa ». Diagrammes E-A.

Le modèle relationnel. Définitions. Dépendances fonctionnelles (DF). Graphe de DF. Fermeture transitive. Couverture minimale classique des DF. Passage du diagramme E-A aux relations. Normalisation de schémas 1e, 2e, 3e, 4e, 5e formes normales, Forme normale de Boyce-Codd. Contraintes d'intégrité. Perte de données et de dépendances.

Complément sur l'algèbre relationnelle : division.

Langage de requêtes SQL. Langage de modification des données : insert, delete, update. Langage d'interrogation des données : projection, sélection, jointures, union, intersection et différences, requêtes imbriquées, opérations d'agrégation (group by, having). Langage de modification de schéma relationnel : create, alter, drop table. Accès concurrents. Définition de vues et index. Autorisations.

Utilisation et implantation d'un SGBD relationnel. Index. Accès concurrents. Intégrité des données. Sécurité. Résistance aux pannes.

### **L3I4 : Architecture des ordinateurs**

#### **Pré-requis : L1E1 – L1E2 - L2E**

- Rappels : codage de l'information et arithmétique des ordinateurs, circuits combinatoires, circuits séquentiels et mémoires.
- Évaluation de performances.
- Architectures des ordinateurs : architecture du jeu d'instructions et organisation matérielle (unité de calcul, unité de contrôle), architecture de Von Neumann, architecture Harvard, DSP.
- Architectures à pile, à accumulateur, à registres généraux, spécialisées.
- Modèle de programmation : langage assembleur, modes d'adressage, passages de paramètres, interfacage avec un langage de haut niveau (C).
- Accélération des performances : notion de pipeline, mémoires cache, prédiction de branchement.
- Hiérarchie mémoire et mémoires distribuées.
- Introduction à la CAO de circuits.

### **L3I5 : Compilation 1**

#### **Pré-requis : L3I2**

S'appuyant sur les aspects théoriques des automates vus en L3I2, ce cours a pour objectif de comprendre comment se passe la partie-avant d'un compilateur :

- Analyse lexicale (automates d'états finis) : principes, programmation manuelle, utilisation d'un générateur.
- Analyse syntaxique (automates à pile) : analyse descendante, descente récursive, utilisation d'un générateur ; analyse ascendante, utilisation d'un générateur ; réparation des erreurs syntaxiques.
- Analyse sémantique : grammaires attribuées, gestion d'une table des symboles, analyse des types et compatibilité, génération d'un arbre abstrait.

## **L3I6 : Systèmes et réseaux**

### **Pré-requis : L1I3 - L2I2**

Approfondir les principes de la programmation système et de l'organisation d'un système d'exploitation et d'un système de communication organisé en couches, par l'utilisation pratique des primitives d'un noyau de système d'exploitation : gestion de fichiers, processus, signaux, mécanismes de communication et de synchronisation, sockets, etc. Application à la mise en œuvre de commandes et d'outils systèmes de base, d'interprètes de commandes, etc.

- Gestion de processus.
- Communications asynchrones, traitements de signaux, programmation événementielle, deadlocks, sections critiques, mémoire partagée, sémaphores, moniteurs, entrées-sorties asynchrones.
- Gestion des ressources : mémoire virtuelle, pagination, allocations, etc.
- Communications synchrones : rendez-vous, communications tubulaires (tubes, FIFOs, sockets), mécanismes d'attente, entrées-sorties multiplexées (poll/select).
- Processus vs threads, différentes formes de parallélisme, coroutines, processus/traitements d'interruption/signaux.

## **L3I7 : Programmation objet**

Concepts de base de la programmation orientée objets. Maîtrise d'un langage orienté objet. Étude de quelques API en insistant sur l'illustration des concepts et des bonnes pratiques de programmation : collections, entrées-sorties, interfaces graphiques, etc.

## **L3I8 : Environnements de programmation et éléments de génie logiciel**

Comprendre les enjeux du développement logiciel à grande échelle (qualité, cycle de vie), maîtriser les outils de base nécessaires au génie logiciel (outils de construction et de documentation, gestionnaires de versions et de configurations, analyseurs de performance), acquérir des compétences en conception orientée objet en utilisant une notation standardisée.

## **L3I9 : Travail d'études**

L'objectif de cette UE est d'habituer les étudiants à préparer un rapport et à présenter convenablement devant un jury le travail effectué. Les étudiants seront par groupe de 2 ou 3 et devront faire une synthèse bibliographique sur un sujet choisi. Le travail consistera donc à

- effectuer des recherches sur le sujet (une aide en ligne sur comment chercher dans une masse d'informations (bibliothèque, Web, etc.) leur est fournie) ;
- préparer une synthèse écrite sous forme d'un rapport à rendre (une aide en ligne sur des outils de rédaction leur est proposée) ;
- présenter oralement cette synthèse devant un jury (quelques cours de communication-expression sont proposés).

## **7.2 Compétences acquises**

L'étudiant a acquis de solides compétences en programmation, particulièrement en programmation impérative et fonctionnelle. Il est en outre capable de concevoir et d'implémenter des programmes de taille raisonnable, évolutifs, traitant des volumes de données importants : conception et programmation orientée objet, conception et utilisation de bases de données.

Il a aussi acquis des bases dans la programmation système ainsi que dans la programmation en réseau. Mais pour autant, des aspects fondamentaux qui sous-tendent la science informatique sont maîtrisés : algorithmique, outils formels sous-jacents.

De plus, des champs d'application concrète dans lesquels ces compétences à la fois théoriques et pratiques s'appliquent ont été étudiés : compilation, optimisation combinatoire, traitement d'images, etc.