

## Master Informatique

### Parcours Logiciels pour systèmes embarqués

#### Objectifs

Le parcours LSE est indifférencié, c'est une formation orientée vers l'ingénierie et la recherche dans le domaine des systèmes embarqués. Les systèmes embarqués constituent un domaine en fort développement, que ce soit dans le secteur des produits grand public (smartphones, objets connectés GPS, set top box, consoles de jeux) ou dans le secteur des dispositifs enfouis de contrôle de véhicules (avions, automobiles, drones) et de chaînes de production.

L'objectif de ce master est de former les étudiants aux outils et techniques logicielles nécessaires à l'élaboration d'applications embarquées, mais aussi (c'est sa spécificité) de donner aux étudiants la capacité de concevoir des méthodes et outils de développement pour ce domaine dans les secteurs clés : développement logiciel, système d'exploitation, compilation, configuration des ressources matérielles, optimisation, test et vérification...



#### Compétences acquises

À l'issue de leur formation, les diplômés sont capables de :

- Concevoir, mettre en œuvre et optimiser des applications spécifiques (applications mobiles, objets connectés, applications de traitement d'image et multimédia, géolocalisation...) pour systèmes embarqués.
- Tester, simuler et valider des architectures matérielles pour systèmes embarqués.
- Réaliser et utiliser des outils de synthèse d'architecture, tout en s'appuyant sur des connaissances algorithmiques et théoriques solides.
- Conditionner et porter des applications ou des systèmes pour l'embarqué (terminaux mobiles, systèmes sur puces).

#### Conditions d'accès

- > En master 1 : être titulaire d'une licence d'informatique ou équivalent.
- > En master 2 : être titulaire d'un master 1 d'informatique ou équivalent.

Accès également possible selon expérience sur procédure de Validation des Acquis Professionnels. Pour tous renseignements : bureau REVA – Tél. : 02 98 01 63 32 ou [reva@univ-brest.fr](mailto:reva@univ-brest.fr)

#### Poursuite d'études

- > Accès possible au doctorat conditionné par un stage orienté recherche.

#### Insertion professionnelle

Les postes généralement occupés par les étudiants sont les suivants: ingénieur de développement embarqué, ingénieur de développement logiciel, ingénieur d'étude, ingénieur consultant,

ingénieur de recherche, etc. Et ce, dans différents organismes ; services R&D des grands groupes, PME/PMI, SSii, ainsi que les organismes publics de recherche. Quelques exemples sont donnés ci-dessous :

- > des services R&D des grands groupes (Thalès, Orange, Dassault, Schneider, General Electric, NXP Semiconductors, Schneider, ACTIA, ...)
- > des PME/PMI, SSii, (Ellidis, Openwide, IS2T, INTERFACE Concept, Altran, ...)
- > des organismes publics de recherche (CEA, IRT, INRIA, CNRS, Universités, ...).

Stage en M1 :

Le master étant indifférencié (professionnel/recherche), le stage peut être réalisé dans un cadre industriel ou universitaire (laboratoire de recherche public). Le stage doit être d'une durée d'au moins 1,5 mois et d'au plus 4 mois. Les stages se déroulent de début Mai jusqu'à fin Aout. Les stages à l'étranger sont possibles.

Stage en M2 :

Le master étant indifférencié (professionnel/recherche), le stage peut être réalisé dans un cadre industriel ou universitaire (laboratoire de recherche public). Le stage doit être d'une durée d'au moins 4 mois et d'au plus 6 mois. Les stages se déroulent de début Avril jusqu'à fin Septembre. Un large ensemble de sujets est mis à disposition des étudiants grâce aux partenariats tissés par l'équipe pédagogique. Les stages à l'étranger sont possibles

L'alternance est possible en M2

#### Infos pratiques

**Faculté des Sciences et Techniques** à Brest  
Ouvert en stage

#### Contacts

**Responsable pédagogique**

DEZAN Catherine

**Contact administratif**

Secrétariat du département informatique

[dept-info@univ-brest.fr](mailto:dept-info@univ-brest.fr)

Tel. 02 98 01 62 06 / 80 11

## Programme

### M1

#### Semestre 7

**S7 INF Compilation** 44h

**S7 INF Renforcement POO** 44h

**S7 INF Ingénierie du logiciel** 44h

**S7 INF Algorithmes et systèmes distribués** 44h

**S7 INF Complément tronc commun, 1 UE à choisir parmi :**

- S7 INF Langage objet scientifique, C++ 22h
- S7 INF Sécurité logicielle 22h
- S7 INF Architectures logicielles 22h

**S7 INF Orientation** 22h

- Initiation à la lecture d'articles 11h
- 1 UE préparation à choisir parmi
  - Préparation à l'alternance 11h
  - Préparation à la mobilité internationale 11h
  - Préparation stage / temps d'accueil en laboratoire 11h

**S7 INF Complément spécifique, 1 UE à choisir parmi :**

- S7 INF Théorie des programmes 44h
- S7 INF Interface Homme-Machine 44h

**Préparation à la vie professionnelle**

- Communication 22h
- Anglais 22h
- Entreprise 10h

#### Semestre 8

**S8 INF Systèmes temps réel** 44h

**S8 4 UEs à choisir parmi**

- S8 INF Systèmes à objets répartis 44h
- S8 INF Administration des systèmes d'exploitation 44h
- S8 INF Systèmes d'information 44h
- S8 INF Programmation Parallèle Haute Performance 44h
- S8 INF Architectures et accélérateurs parallèles 44h

- S8 INF Méthodes computationnelles 44h

**S8 INF Expérience professionnelle, obligatoire :**

- S8 INF Temps d'accueil en laboratoire TAL 6h
- S8 INF Stage 6h

**Préparation à la vie professionnelle**

- Anglais 22h
- Communication 22h
- Entreprise 10h

### M2

#### Semestre 9

**S9 LSE Modèles et Langages pour le parallélisme** 48h

**S9 LSE Objets Connectés et Sécurité** 48h

**S9 LSE Systèmes d'exploitation pour l'embarqué** 48h

**S9 LSE System On Chip** 48h

**S9 LSE IA embarquée** 48h

**S9 LSE Sûreté de Fonctionnement** 48h

**Préparation à la vie professionnelle**

- Communication 22h
- Anglais 22h
- Entreprise 10h

#### Semestre 10

**S10 LSE Ouverture scientifique LSE** 24h

**S10 LSE Projet de Recherche**

- S10 LSE Atelier de développement collaboratif 20h
- S10 LSE Projet de recherche encadré (6 semaines)

**S10 LSE Stage (4 à 6 mois) ou TAL (temps d'accueil en laboratoire)** 16h

Dernière mise à jour le 13 mars 2025

## S7 INF Compilation

### Présentation

L'objectif de ce cours est d'appréhender le processus de compilation en présentant les modèles formels et algorithmes sous-jacents aux méthodes de compilation, ainsi que des techniques et outils pour les mettre en application de façon à réaliser un compilateur.

Les analyses successives réalisées par un compilateur sont détaillées : analyse lexicale, analyse syntaxique, analyse sémantique et production de code.

La mise en application point par point est réalisée avec des outils Java, et un projet en fin de cours permet de consolider les nouveaux acquis.

#### 4 crédits ECTS

Volume horaire

Travaux Dirigés : 32h

Travaux Pratiques : 12h

### Objectifs

- > Connaissance et compréhension des différentes parties d'un compilateur
- > Savoir mettre en oeuvre les techniques de compilation sur un exemple de langage réduit

### Pré-requis nécessaires

Connaissances de base en algorithmique, en modèles formels (automates à états finis, expressions régulières) et programmation en Java.

### Compétences visées

- > Acquisition des fondements théoriques et des principes méthodologiques nécessaires à la compréhension et à la réalisation des différentes phases constitutives de la compilation, de l'analyse lexicale à la génération de code
- > Capacité à mettre en pratique ces principes dans le cadre d'applications Java
- > Mise en oeuvre d'éléments d'un compilateur (simple) en utilisant les outils Java dédiés
- > Automatisation des traitements sur des fichiers via l'utilisation des méthodes et outils présentés

### Bibliographie

- > Aho, Sethi, Ullman : "Compilateurs : principes, techniques et outils", InterEditions, 1991
- > Wilhelm, Maurer : "Compiler Design", Addison-Wesley, 1995 N. Silverio : "Réaliser un compilateur", Eyrolles, 1995

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/4	
	CT	Ecrit - devoir surveillé	120	3/4	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

## S7 INF Renforcement POO

### Présentation

Ce cours a pour objectif de renforcer la maîtrise de l'algorithmique, des structures de données et de la programmation orientée objet, en particulier avec le langage Java.

Dans un premier temps, les notions élémentaires de structures de données statiques (enregistrements, tableaux) et dynamiques (tables de hachage, chaînes, arbres) sont rappelées. Ces concepts et leurs variantes sont mis en œuvre et outillés en Java.

Dans un deuxième temps, ces structures élémentaires sont utilisées pour concevoir une authentique API orientée objet sur le modèle de la hiérarchie des collections Java (listes, queues, ensembles, ensembles ordonnés). Cette API est ensuite elle-même utilisée pour réaliser des applications exigeantes en termes de performance.

#### 4 crédits ECTS

Volume horaire

Cours Magistral : 22h

Travaux Pratiques : 22h

### Objectifs

- > Consolider les acquis de licence concernant la conception et la programmation orientée objet, l'algorithmique élémentaire et les structures de données.
- > Illustrer la séparation des préoccupations liées à la conception d'une API, à son utilisation et à sa mise en œuvre.
- > Uniformiser les prérequis de programmation Java nécessaires à d'autres UE du M1, quel que soit le parcours d'origine avant l'admission en M1.

### Pré-requis nécessaires

Connaissances élémentaires de la programmation orientée objet en général et du langage Java en particulier.

### Compétences visées

- > Savoir utiliser à bon escient les constructions élémentaires du langage Java.
- > Reconnaître la solution algorithmique adaptée à un besoin standard de programmation.
- > Distinguer l'usage d'une API et sa mise en œuvre.
- > Distinguer le besoin fonctionnel du besoin technique.

### Bibliographie

- > Mickaël Kerboeuf : Fondements de la programmation orientée objet avec Java 8. Références Sciences, Ellipses, décembre 2016, ISBN 9782340014824.
- > Mickaël Kerboeuf : Algorithmique et programmation objet. Références Sciences, Ellipses, mai 2020, ISBN 9782340037984.
- > James Gosling, Bill Joy, Guy Steele et Gilad Bracha : The Java(TM) Language Specification (3rd Edition). Addison-Wesley Professional, 2005, ISBN 0321246780.
- > Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides : Design Patterns, Elements of Reusable Object-oriented Software. Addison-Wesley Longman Publishing Co., 1995, ISBN 0-201-63361-2.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CT	Travaux Pratiques	120	1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

## S7 INF Ingénierie du logiciel

### Présentation

Ce module permet de sensibiliser les étudiants à la rationalisation de la production de logiciels, et à l'amélioration de la productivité du développement. L'enseignement est essentiellement basé sur des études de cas.

### Objectifs

L'objectif pédagogique de ce module est de maîtriser les concepts de base de l'ingénierie logicielle. On y étudie à la fois les méthodes, les modèles UML et les outils.

### Pré-requis nécessaires

Maîtrise des concepts de base de la programmation objet

### Compétences visées

- > savoir spécifier une fonctionnalité logicielle
- > savoir modéliser à l'aide de diagrammes UML les diverses productions des étapes du développement
- > savoir décrire des exigences
- > savoir établir un plan de test

### Bibliographie

- > De Merise à UML » Nasser Kettani, Dominique Mignet, Pascal Paré, Camille Rosenthal-Sabroux, Eyrolles, 2001
- > Pascal Roques, UML2 par la pratique, Eyrolles, 2004
- > Model driven Architecture : Applying MDA to Enterprise Computing, David S. Frankel, John Wiley & Sons

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Ecrit - devoir maison		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

#### 4 crédits ECTS

Volume horaire

Travaux Dirigés : 22h

Travaux Pratiques : 14h

Cours Magistral : 8h

# S7 INF Algorithmes et systèmes distribués

## Présentation

Cette UE porte sur l'étude des principes fondamentaux des applications distribuées et sur la programmation d'algorithmes pour résoudre les problèmes classiques de ce domaine via une approche par processus communicants.

## Objectifs

Cette UE présente globalement les différents modèles de systèmes distribués (notamment en termes de communication). L'accent est mis sur la nécessité d'appuyer la conception d'un algorithme distribué sur un modèle synchrone de processus communicants permettant de garantir sa robustesse, en faisant abstraction de l'environnement technologique d'implémentation.

Ces notions sont illustrées par des exemples significatifs d'algorithmes distribués mis en œuvre dans les réseaux informatiques, au niveau des couches liaison, réseau et transport, et par des algorithmes classiques du domaine. Les algorithmes distribués abordés sont notamment : horloges de lamport, exclusion mutuelle, election, arbre de recouvrement, calculs globaux adaptés à une topologie, commutation de messages, routage de messages.

Le langage support pour l'expression et l'expérimentation des algorithmes est le langage Go. On étudie plus particulièrement dans ce langage la capacité de conception de processus poids-léger communicants, interagissant via des canaux synchrone ou asynchrone de type CSP (communicating sequential processes)

### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 16h

Travaux Dirigés : 16h

Cours Magistral : 12h

## Pré-requis nécessaires

- > "Systèmes d'exploitation
- > "Algorithmique des graphes "
- > "Réseaux IP : programmation réseaux"

## Compétences visées

- > Savoir concevoir des algorithmes distribués fiables, via l'utilisation du modèle de programmation par processus communicants
- > Savoir programmer des algorithmes distribués en langage Go

## Bibliographie

Distributed algorithms, N.Lynch ; Morgan Kaufmann, 96

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CT	Travaux Pratiques	60	1/3	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

## Langue d'enseignement

Français

**S7 INF Complément tronc commun, 1 UE à choisir parmi :****2 crédits ECTS**

## S7 INF Langage objet scientifique, C++

### Présentation

Le but de cet enseignement est d'acquérir une bonne compréhension et une bonne connaissance des mécanismes de base du langage orienté objets C++.

Contenu détaillé :

- > Du C au C++ en passant par le C+
- > Les classes
- > L'héritage
- > La liaison dynamique

#### 2 crédits ECTS

Volume horaire

Travaux Pratiques : 12h

Cours Magistral : 10h

### Objectifs

Maîtrise approfondie du langage objets C++

### Pré-requis nécessaires

programmation en C et Java

### Compétences visées

Compréhension et acquisition de bonnes pratiques de programmation du C++.

### Bibliographie

Bjarne Stroustrup, "Le langage C++". Addison-Wesley.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	60	1/2	
	CT	Travaux Pratiques	120	1/2	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Travaux Pratiques	120		

### Langue d'enseignement

Français

## S7 INF Sécurité logicielle

### Présentation

Dans cette UE, une introduction générale aux problématiques et solutions de sécurité est effectuée avec un focus sur les aspects logiciels, avec mise en application. Les services de confiance, la signature électronique et les outils permettant leur mise en œuvre (bibliothèque de fonctions cryptographiques Openssl, ...) sont ensuite abordés, ainsi que les outils et solutions en java pour la sécurité.

#### 2 crédits ECTS

Volume horaire

Travaux Dirigés : 8h

Travaux Pratiques : 8h

Cours Magistral : 6h

### Objectifs

L'objectif de cette UE est de former aux bases de la sécurité logicielle.

### Pré-requis nécessaires

Langage C, Bases de données SQL, langage Java

### Compétences visées

- > connaissance des principales problématiques liées à la sécurité logicielle
- > compétences de base en matière de détection de failles logicielles
- > compétences de base en matière de programmation sécurisée

### Bibliographie

Sécurité informatique. Gildas Avoine, Pascal Junod, Philippe Oechslin. Vuibert, 2ème édition, 2009

Cryptographie appliquée, Bruce Schneier, Vuibert 2001.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		2/3	
	CT	Travaux Pratiques	35	1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Oral	15		

### Langue d'enseignement

Français

## S7 INF Architectures logicielles

### Présentation

Ce module permet de sensibiliser les étudiants à l'importance de l'architecture logicielle dans la qualité du développement logiciel, en particulier pour la maîtrise du changement. L'enseignement est essentiellement basé sur des études de cas.

### Objectifs

L'objectif de ce module est l'apprentissage des concepts de base des architectures logicielles au travers l'étude des Design patterns.

### Pré-requis nécessaires

Maîtrise des concepts de base de la programmation objet et la modélisation à l'aide de diagrammes UML

### Compétences visées

- Connaître les principes et l'utilisation des principaux Design Pattern du développement logiciel
- Mettre en place les principaux Design-Pattern dans un langage de programmation objet

### Bibliographie

« Design Patterns: Elements of Reusable Object-Oriented Software” Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison-Wesley, 1994

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

#### 2 crédits ECTS

Volume horaire

Travaux Dirigés : 8h

Travaux Pratiques : 12h

Cours Magistral : 2h

## S7 INF Orientation

**2 crédits ECTS**

Volume horaire

Cours Magistral : 5h

Travaux Pratiques : 17h

# Initiation à la lecture d'articles

## Présentation

Ce module permet de sensibiliser les étudiants à l'activité de recherche scientifique au travers d'une présentation générale du déroulement d'une activité de recherche dans le domaine des sciences. Il comporte aussi une initiation à la présentation d'articles scientifiques.

### 1 crédits ECTS

Volume horaire

Travaux Pratiques : 6h

Cours Magistral : 5h

## Objectifs

L'objectif est de comprendre l'objectif du métier de chercheur et les activités principales liées à la recherche académique ou en entreprise. En particulier, l'activité de publication scientifique est illustrée au travers d'un exercice de revue d'articles scientifiques.

## Pré-requis nécessaires

Connaissances de base en développement et déploiement de logiciel.

## Compétences visées

- savoir évaluer un article scientifique
- savoir décrire l'objectif d'une activité de recherche

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Ecrit - devoir maison		1/1	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Ecrit - devoir maison		100	

## Langue d'enseignement

Français avec aide ponctuelle en anglais

## 1 UE préparation à choisir parmi

1 crédits ECTS

## Préparation à l'alternance

### Présentation

Cette UE permet d'accompagner les étudiants dans leur projet de formation professionnelle de Master (stage de M1 et alternance en M2).

**1 crédits ECTS**

Volume horaire

Travaux Pratiques : 11h

### Objectifs

- affiner son projet professionnel
- obtenir un stage de M1 en vue d'obtenir un contrat de professionnalisation ou d'apprentissage.

### Compétences visées

- > savoir mobiliser ses capacités pour mener à bien son projet de formation professionnelle
- > savoir défendre son parcours au travers d'un CV et d'entretiens professionnels
- > savoir présenter un travail d'ingénierie informatique

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Dossier		100	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Dossier		100	

### Langue d'enseignement

Français

## Préparation à la mobilité internationale

### Présentation

Cette UE permet d'accompagner les étudiants dans leur projet de formation professionnelle de Master (M1 et M2).

**1 crédits ECTS**

Volume horaire

Travaux Pratiques : 11h

### Objectifs

- affiner son projet professionnel
- obtenir une acceptation dans une université étrangère.

### Compétences visées

- > savoir mobiliser ses capacités pour mener à bien son projet de formation professionnelle
- > savoir défendre son parcours au travers d'un CV et de dossier de candidature
- > savoir présenter un travail de recherche ou d'ingénierie informatique

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Dossier		100	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Dossier		100	

### Langue d'enseignement

Français avec aide ponctuelle en anglais

## Préparation stage / temps d'accueil en laboratoire

### Présentation

Cette UE permet d'accompagner les étudiants dans leur projet de formation professionnelle de Master (M1 et M2).

**1 crédits ECTS**

Volume horaire

Travaux Pratiques : 11h

### Objectifs

- affiner son projet professionnel
- obtenir un stage en entreprise ou en laboratoire

### Compétences visées

- > savoir mobiliser ses capacités pour mener à bien son projet de formation professionnelle
- > savoir défendre son parcours au travers d'un CV et d'entretiens professionnels
- > savoir présenter un travail de recherche ou d'ingénierie informatique

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Dossier		100	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Dossier		100	

### Langue d'enseignement

Français

**S7 INF Complément spécifique, 1 UE à choisir parmi :****4 crédits ECTS**

## S7 INF Théorie des programmes

### Présentation

Ce cours étudie différents aspects des fondements de la programmation en termes de calculabilité et de complexité des calculs. On définit notamment une sémantique de langage comme support formel pour la représentation de programmes. On introduit également le lambda-calcul comme objet d'étude dans le cadre de la programmation.

#### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 12h

Cours Magistral : 16h

Travaux Dirigés : 16h

### Objectifs

L'objectif est de sensibiliser les étudiants aux diverses notions théoriques liées aux programmes telles que la calculabilité, la complexité, et la modélisation et vérification de programmes.

### Pré-requis nécessaires

Connaissances, niveau licence, dans le domaine des fondements des langages et de la programmation et des systèmes formels.

### Compétences visées

Acquisition de fondements théoriques nécessaires à la compréhension de la programmation.

Etre capable de pouvoir analyser la complexité des programmes.

Être capable de concevoir et vérifier différents schémas de programmes.

### Bibliographie

1. Lambda-calcul, types et modèles, Jean-Louis Krivine, Editions Masson.
2. The Lambda-calculus. H.P. Barendregt. Volume 103, Elsevier Science Publishing Company.
3. Logique, réduction, résolution. R. Lalement. Éditions Masson.
4. Logique mathématique (Tome 2). René Cori et Daniel Lascar. Editions Dunod.
5. Introduction à la calculabilité. Pierre Wolper. InterEditions.
6. Introduction à l'algorithmique. T.H Cormen, C.E Leiserson, R.L Rivest, C. Stein. Edition Dunod.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CT	Travaux Pratiques		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

## S7 INF Interface Homme-Machine

### Présentation

Cette UE permet d'introduire les concepts de base de la programmation des interfaces graphiques évoluées et avancées pour le Web et Android. L'objectif est de pouvoir développer des interfaces de qualité respectant la norme MVC (Model-View-Controller) ainsi que les outils existants permettant de mettre en place ce type de modèles.

La partie Web est focalisée sur l'utilisation des langages HTML / CSS / TypeScript. L'API Flexbox est privilégiée. Nous mettons en pratique nos connaissances au travers d'exemples guidés et d'un projet à la thématique libre mais devant respecter plusieurs critères de qualité.

La partie Android, elle permet d'introduire les concepts de base de l'environnement de programmation mobile Android Studio et le SDK Android. Elle est composée de 4 parties. La première partie présente l'environnement de développement Android dont le système, le SDK et la plateforme Android Studio. La deuxième partie présente les concepts fondamentaux et les composants d'un projet et d'une application. La partie 3 présente les techniques de navigation entre plusieurs activités d'une application. La dernière partie sera consacrée à la partie IHM sous Android.

#### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 16h

Travaux Dirigés : 12h

Cours Magistral : 16h

### Objectifs

Maîtriser les langages et environnements suivants :

1. HTML, CSS, Flexbox et TypeScript
2. Programmation Android (SDK et Android Studio)
3. Modélisation MVC et techniques de programmation d'interfaces graphiques ergonomiques

### Pré-requis nécessaires

Java, HTML, CSS, JavaScript

### Compétences visées

- savoir placer l'utilisateur au centre d'une application
- savoir séparer le noyau et l'ihm d'une application
- connaître les mécanismes pour garder la cohérence noyau <=> ihm (Model-View-Update (MVU) )
- connaître les critères externes d'une application (point de vue de l'utilisateur)
- connaître les critères internes (point de vue développeur)
- distinguer les approches ihm pour le desktop et pour le web
- avoir des notions de Threads et de tâches (Task) et leurs utilités pour les IHMs (Java/JavaFX et Android)
- savoir programmer pour Android (SDK) grâce à l'environnement de développement Android Studio
- connaître les Activités et les Intents
- être capable de mettre au point des IHMs pour Android
- programmer des composants personnalisés (Android)

### Bibliographie

<https://developer.android.com/training/index.html>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://www.w3schools.com/>

<https://deveppaper.com/ten-year-changes-in-gui-application-architecture-mvc-mvp-mvvm-unidirectional-clean/>

### Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	1/2	
	CC	Travaux Pratiques		1/2	

## Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

## Langue d'enseignement

---

Français

## Préparation à la vie professionnelle

**6 crédits ECTS**

## Communication

### Présentation

L'UE communication est destinée aux étudiants de master au semestre 7.

#### 2 crédits ECTS

Volume horaire

Cours Magistral : 7h

Travaux Dirigés : 15h

### Objectifs

L'objectif est de maîtriser une culture scientifique de spécialité, d'initier une réflexion sur l'épistémologie et l'histoire des sciences et de savoir communiquer sur ses travaux de recherche en les vulgarisant.

Un autre objectif est celui de l'insertion professionnelle par le biais de la rédaction d'une fiche de poste dans leur spécialité et la sélection de candidats aptes à y répondre.

### Compétences visées

Les étudiants développent leurs capacités rédactionnelles par la rédaction d'un article scientifique ou de médiation scientifique.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CC	Ecrit et/ou Oral		1/1	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	Contrôle ponctuel	Oral	15	1/1	

# Anglais

## Objectifs

Intégration dans le monde du travail.

### 3 crédits ECTS

Volume horaire

Cours Magistral : 8h

Travaux Dirigés : 14h

## Compétences visées

Préparer une candidature à l'embauche / Comprendre l'organisation d'une entreprise / Participer et animer une réunion / Prendre des notes / Rédiger un compte-rendu

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Ecrit - devoir surveillé	90	1/2	
EC	CC	Ecrit et/ou Oral		1/2	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Ecrit - devoir surveillé	30	1/1	

## Entreprise

**1 crédits ECTS**

Volume horaire

Autres : 10h

### Modalités de contrôle des connaissances

---

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
Autres	Autre modalité	Ecrit et/ou Oral		1/1	validation par "badge" - Pas de session 2 -

## S8 INF Systèmes temps réel

### Présentation

Méthodes et technologies pour spécifier, concevoir, vérifier et réaliser des systèmes temps-réel critiques.

### Objectifs

L'objectif de ce module est de présenter aux étudiants les méthodes et technologies qui peuvent être utilisées pour spécifier, concevoir et réaliser les systèmes temps-réel critiques. Après une introduction qui présente les objectifs et contraintes de ces systèmes, une ou plusieurs méthodes de spécification, de conception et de vérification adaptées aux systèmes temps réel sont exposées. Enfin, un exemple de technologie permettant d'implanter ces systèmes est abordé (ex : C et VxWorks, ou Esterel/SCADE, Ada, ...).

#### 4 crédits ECTS

Volume horaire

Cours Magistral : 16h

Travaux Dirigés : 16h

Travaux Pratiques : 12h

### Pré-requis nécessaires

- > Programmation concurrente, systèmes d'exploitation
- > Réseaux
- > Recherche opérationnelle, théorie des graphes

### Compétences visées

- > Comprendre les contraintes liées aux systèmes temps réel
- > Savoir spécifier, concevoir et vérifier un système concurrent contraint par le temps
- > Maîtriser au moins une technologie permettant d'implanter un système temps réel concurrent

### Bibliographie

- > Real time systems and programming language, A. BURNS, A. Welling. ADDISON, WESLEY, 2009
- > Building Parallel, Embedded and Real-Time Applications with Ada. J.W. McCormick, F. Singhoff, J. Hugues. Cambridge University Press. 365 pages. Juillet 2010

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/4	
	CT	Ecrit - devoir surveillé	120	3/4	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	1/1	

### Langue d'enseignement

Français

## S8 4 UEs à choisir parmi

## S8 INF Systèmes à objets répartis

### Présentation

Etude des applications réparties hétérogènes.

### Objectifs

L'objectif de cette UE est de connaître et savoir utiliser différentes technologies permettant de construire des applications réparties hétérogènes.

#### 3 crédits ECTS

Volume horaire

Cours Magistral : 12h

Travaux Pratiques : 32h

### Pré-requis nécessaires

système d'exploitation et réseaux, langage et algorithmique, architecture des systèmes distribués, applications réparties

### Compétences visées

- > Connaître les mécanismes mis en oeuvre dans les systèmes et les services à objets répartis.
- > Connaître les modèles d'interaction dans les applications réparties (client/serveur, agents mobiles)
- > Savoir programmer en objet/service répartis
- > Comprendre les enjeux de l'intégration de service, les problèmes d'interopérabilité (systèmes, langages, protocoles réseaux)
- > Connaître les principaux mécanismes et patrons de conception exploités dans les plates-formes offrant un support pour l'intégration de systèmes et l'interopérabilité : patrons service de nommage (naming), service cycle de vie (factory), proxy, service d'événements, service transactionnel ...

### Descriptif

#### 1) Introduction aux systèmes répartis

Problématiques et notions de transparence. Modèles d'interaction. Mécanismes et patrons usuels. Exemples de systèmes répartis

#### 2) Les approches objets répartis

- > Le standard CORBA. L'architecture OMA, notion de bus à objets. Mapping Java et C++. Services standards CORBA (naming et event services)
- > Java RMI. Le modèle agent mobile : migration, services web. Architecture et fonctionnement de java RMI
- > La plate-forme C#/ .NET et les web services . L'architecture .Net, XML et ses outils, C#

#### 3) Les approches services

- > L'architecture JEE
- > Accès aux bases de données distantes, JDBC
- > Gestion des requêtes HTTP, objets Servlet et HttpSession, validation des requêtes
- > Conception d'une application MVC basée sur les Servlets
- > L'architecture NodeJS
- > Programmation asynchrone, gestion des événements, promesses (promises)
- > Échanges de données en JSON (Javascript Object Notation)
- > Conception d'un serveur Web, le framework express, l'API REST, les routes
- > Conception d'une application Web répartie, le framework React

Répartition approximative : 2h introduction, 10h CORBA, 5h RMI, 5h .NET, 12h JEE, 12h NodeJS

### Bibliographie

- > Learning Node.js, A Hands-On Guide to Building Web Applications in JavaScript. [Marc Wandschneider](#). Addison Wesley editor. 2017.
- > Java network programming and distributed computing. David Reilly, Addison Wesley editor. 2003.
- > .NET Application Development with C#, ASP.NET, ADO.NET and Web Services. Hanspeter Mossenbock, Dietrich Birngruber, Albrecht Woss, Wolfgang Beer. Addison Wesley editor. 2004.

### Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	3/4	
	CC	Travaux Pratiques		1/4	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

---

Français

## S8 INF Administration des systèmes d'exploitation

### Présentation

Cette UE aborde le rôle de l'administrateur et ses principales tâches, la méthodologie d'administration système, la procédure d'installation d'un système Linux, la gestion des identités et des annuaires, la gestion de l'espace disque (partition, systèmes de fichiers, organisation, sauvegarde), les services réseaux (NIS, DNS, NFS), les pare-feux et le filtrage réseau, le démarrage des services, l'automatisation des tâches d'administration, les outils de supervision.

#### 3 crédits ECTS

Volume horaire

Travaux Dirigés : 4h

Cours Magistral : 12h

Travaux Pratiques : 28h

### Objectifs

Le but de cette UE est de former à l'administration système et réseau sous Linux.

### Pré-requis nécessaires

- > Connaissance des fonctions d'un système d'exploitation et des principes de mise en œuvre.
- > Connaissance de base en programmation shell.
- > Connaissance de base sur les réseaux informatique

### Compétences visées

- > Connaître le rôle et les tâches de l'administrateur système
- > Maîtriser la procédure d'installation d'un système Linux
- > Maîtriser les bases de l'administration d'un système
- > Savoir configurer les services réseaux, les pare-feux et le filtrage réseau
- > Savoir gérer les identités et les annuaires

### Bibliographie

Linux administration, Jean-François Bouchaudy, Gilles Goubet, Eyrolles, 2007

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	60	1/3	
	CC	Travaux Pratiques		2/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	60	100%	

### Langue d'enseignement

Français

## S8 INF Systèmes d'information

### Présentation

Au travers de cette UE, les étudiants considèrent les architectures et technologies utiles à la mise en place d'un système d'information.

### Objectifs

le premier objectif est de présenter les architectures à base de services web pour les systèmes d'information.

L'objectif de cette UE est aussi d'illustrer la diversité des systèmes de gestion de bases de données en mettant en œuvre plusieurs types de SGBD qualifiés de NoSQL.

Par ailleurs, les environnements de développement utilisant des outils de travail collaboratif, leur usage est ici illustré via des projets.

### Pré-requis nécessaires

Cette UE s'appuie sur les connaissances des langages Java et Javascript.

Les bases du développement web sont supposées connues

### Compétences visées

Etre autonome dans la mise en œuvre d'un site web s'appuyant sur une structure à base de services

- > Mise en œuvre du développement web à base de services REST en utilisant Java avec le Framework Spring Boot
- > Mise en œuvre du développement web à base de services REST en utilisant NodeJS

Savoir choisir un système de base de données adapté aux besoins entre

- > Base de données relationnelle
- > Base de données orientée document (MongoDB)
- > Base de données graphe (Neo4j)

Être autonome dans l'usage d'un logiciel de versionning et travail collaboratif, utilisation de gitlab.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

#### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 32h

Cours Magistral : 6h

Travaux Dirigés : 6h

# S8 INF Programmation Parallèle Haute Performance

## Présentation

Cette UE porte sur la programmation parallèle multi-threads et multi-processeurs. Elle aborde les principaux modèles d'exploitation du parallélisme, la mesure de la performance, et se focalise sur la programmation openMP et PVM.

### 3 crédits ECTS

Volume horaire

Travaux Dirigés : 16h

Travaux Pratiques : 14h

Cours Magistral : 14h

## Objectifs

Cette UE est composée de quatre parties :

- > Introduction au parallélisme : modèles de parallélisme (données, contrôle, flot), modèles de programmation parallèle, mesures et limites de performance.
- > Programmation parallèle multi-threads en mémoire partagée avec openMP : parallélisation de boucles, ordonnancement statique et dynamique, création de tâches dynamiques.
- > Programmation parallèle multi-processus à grande échelle en mémoire distribuée avec PVM (Parallel Virtual Machine) : passage de messages, création/destruction de processus, paradigme maître/esclave.
- > Algorithmes et techniques pour le parallélisme massif : ordonnancement, analyse de dépendances, réécriture de nids de boucle, calcul systolique.

## Pré-requis nécessaires

- > Systèmes d'exploitation
- > Architecture des ordinateurs

## Compétences visées

- > Capacité d'identification des différents types de parallélisme exploitables dans une application
- > Parallélisation multi-threads de programmes séquentiels en openMP
- > Savoir concevoir des programmes multi-processus pour le calcul parallèle
- > Amélioration des performances parallèles par réécriture de code

## Bibliographie

- > Algorithmes et architectures systoliques - P.Quinton et Y. Robert, Masson, 1989
- > Algorithmique parallèle - A. Legrand et Y. Robert, Dunod, 2003
- > PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing - A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam, MIT Press, 1994

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

## Langue d'enseignement

Français

## S8 INF Architectures et accélérateurs parallèles

### Présentation

Cette UE porte sur la connaissance et la programmation des principaux modèles d'architectures et d'accélérateurs parallèles. Elle se focalise plus particulièrement sur la programmation des accélérateurs FPGA et GPU.

### Objectifs

Cette UE présente globalement les différents modèles d'architectures parallèle et leur mise en œuvre matérielle :

- > Architectures super-scalaires, VLIW, super-pipeline, vectorielle pour des cœurs de processeurs
- > Processeurs multicores et manycores et leur hiérarchie mémoire
- > Accélérateurs graphiques GPU
- > Architectures reconfigurables FPGA
- > Supercalculateurs

D'un point de vue conceptuel et pratique, elle se concentre sur la programmation des deux familles d'accélérateurs parallèles les plus répandus pour le calcul intensif (simulation, traitement d'image, IA, data mining), que sont les GPU et FPGA :

- > Dans la partie GPU, on apprend à programmer des kernels de calcul intensif pour GPU. Les langages OpenCL et Compute Shader sont étudiés à l'aide d'environnements de programmation comme [MatrixStudio](#) ou [ShaderToy](#). Divers sujets sont alors abordés comme le traitement d'image, la simulation de drones, la simulation de systèmes biologiques et plus généralement le calcul scientifique intensif.
- > Dans la partie FPGA, on introduit la synthèse de haut-niveau (HLS), permettant la conception de traitements parallélisés à partir d'une description C ou C++. On apprend comment caractériser et comparer les performances, en termes de fréquence d'horloge, de cycles de fonctionnement et de ressources utilisées. L'environnement utilisé est Vitis, pour la programmation des FPGAs du constructeur Xilinx/AMD.

### Pré-requis nécessaires

- > Systèmes d'exploitation
- > Architecture des ordinateurs

### Compétences visées

- > Savoir concevoir et programmer des cœurs de traitement pour FPGA par synthèse de haut niveau
- > avoir concevoir et programmer des kernels de calcul intensif pour GPU avec OpenCL

### Bibliographie

- > Computer Organization and Design - David A.Patterson and John L.Hennessy, Morgan Kaufmann Publishers, 1997
- > Reconfigurable Computing: The Theory and Practice of Fpga-based Computation - S. Hauck et A. DeHon, Morgan Kaufmann, 2007
- > OpenCL Programming by Example. Ravishekhar Banger, Koushik Bhattacharyya. Packt Publishing. ISBN-13: 978-1849692342. 2013.
- > OpenCL Programming Guide. Aaftab Munshi. Addison-Wesley. ISBN-13: 978-0-321-74964-2. 2013.
- > Using OpenCL – Programming Massively Parallel Computers. Janusz Kowalik et Tadeusz Puzniakowski. IOS Press. ISBN 978-1-61499-029-1. 2012.

### Modalités de contrôle des connaissances

#### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 30h

Cours Magistral : 14h

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

---

Français

## S8 INF Méthodes computationnelles

### Présentation

L'objectif de cette UE est de connaître les principales méthodes du soft computing (méthodes computationnelles) de façon à en cerner les intérêts et à en apprécier les limites pratiques. Une définition du soft-computing: "Based on fuzzy logic, Evolutionary Computing, Neural networks, Genetic Algorithms, or their combinations to yield synergetic effects, Soft Computing is aimed to realize intelligent systems for different applications"

Contenu susceptible d'évoluer :

- > Logique floue : application au contrôle
- > Systèmes Immunitaires Artificiels pour l'optimisation et la détection d'anomalies,
- > Méta-heuristiques pour l'optimisation (recuit simulé, Algorithmes Génétiques, etc.),
- > Programmation Multi-Objectifs (branch and bound)

#### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 26h

Travaux Dirigés : 2h

Cours Magistral : 16h

### Objectifs

L'objectif est de connaître les principales méthodes du soft computing (méthodes computationnelles) de façon à en cerner les intérêts et à en apprécier les limites pratiques.

### Pré-requis nécessaires

programmation en C

### Compétences visées

Connaissances des principales méthodes théoriques de soft computing. Maîtrise de ces méthodes à travers de nombreux exercices pratiques.

### Bibliographie

- > Logique floue. Exercices corrigés et exemples d'applications. Bernadette Bouchon-Meunier, Laurent Foulloy, Mohammed Ramdani. Cépaduès, 1998.
- > Métaheuristiques pour l'optimisation difficile. Patrick Siarry. Eyrolles, 2005.
- > Artificial Immune Systems: A new computational Intelligence Approach. Springer-Verlag, 2002.
- > Algorithmes génétiques et réseaux de neurones. Jean-Michel Renders. Hermès, 1995.

Et la vision du soft computing de l'Univ. de Berkeley: <http://www.cs.berkeley.edu/~zadeh>

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	2/3	
	CC	Travaux Pratiques		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120		

### Langue d'enseignement

Français

**S8 INF Expérience professionnelle, obligatoire :****8 crédits ECTS**

## S8 INF Temps d'accueil en laboratoire TAL

### Présentation

Développement d'un projet informatique R&D au sein d'un laboratoire de recherche.

### Objectifs

- > Découvrir les métiers de la recherche
- > Mener un projet au sein d'un laboratoire de recherche

**8 crédits ECTS**

Volume horaire

Projet tutoré : 6h

### Pré-requis nécessaires

Maîtriser les outils et environnement du développement logiciel.

### Compétences visées

- > Savoir mener un travail de développement logiciel
- > Capacité à intégrer une équipe de recherche
- > Capacité à mobiliser ses connaissances théoriques et pratiques pour résoudre un problème d'ingénierie informatique.
- > Capacité à faire un état de l'art ou une veille technologique

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Autre nature		1/2	
	CC	Oral - soutenance	15	1/4	
	CC	Ecrit - rapport		1/4	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Oral - soutenance		1/4	
	Report de notes	Ecrit - rapport		1/4	
	Report de notes	Autre nature		1/2	

## S8 INF Stage

### Présentation

Développement d'un projet informatique au sein d'une entreprise.

### Objectifs

- > Découvrir les métiers du développement logiciel
- > Mener un projet au sein d'une entreprise

**8 crédits ECTS**

Volume horaire

Stages : 6h

### Pré-requis nécessaires

Maîtriser les outils et environnement du développement logiciel.

### Compétences visées

- > Savoir mener un travail de développement logiciel
- > Capacité à intégrer une équipe de développement.
- > Capacité à mobiliser ses connaissances théoriques et pratiques pour résoudre un problème d'ingénierie informatique.
- > Capacité à faire un état de l'art ou une veille technologique.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Autre nature		1/3	
	CC	Oral - soutenance	15	1/3	
	CC	Ecrit - rapport		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Oral - soutenance		1/3	
	Report de notes	Ecrit - rapport		1/3	
	Report de notes	Autre nature		1/3	

## Préparation à la vie professionnelle

**6 crédits ECTS**

# Anglais

## Objectifs

Intégration dans le monde du travail.

### 3 crédits ECTS

Volume horaire

Cours Magistral : 8h

Travaux Dirigés : 14h

## Compétences visées

Présenter des résultats et argumenter / Conseiller / Simplifier / Vulgariser / Rédiger une note de synthèse / Présentation orale d'un objet technique ou d'une campagne d'information

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Ecrit - devoir surveillé	90	1/2	
EC	CC	Autre nature		1/2	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Oral	10		

## Communication

### Présentation

L'UE communication est destinée aux étudiants de master au semestre 8.

### Objectifs

L'objectif est de développer une analyse critique des médias par l'étude du monde de l'édition scientifique et de la presse généraliste et de spécialité.

### Compétences visées

Les étudiants sont amenés à développer leurs capacités rédactionnelles à travers la réalisation d'une revue de presse puis d'un dossier de presse mais également d'expression orale par la conception de podcasts.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CC	Ecrit et/ou Oral		1/1	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Oral	15	1/1	

#### 2 crédits ECTS

Volume horaire

Travaux Dirigés : 15h

Cours Magistral : 7h

## Entreprise

**1 crédits ECTS**

Volume horaire

Travaux Dirigés : 10h

### Modalités de contrôle des connaissances

---

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CC	Autre nature		1	Pas de session 2

## S9 LSE Modèles et Langages pour le parallélisme

### Présentation

Cette UE porte sur les modèles, les langages et la compilation pour la programmation ou la conception de systèmes parallèles :

- > au niveau modèles et langages, l'objectif est de connaître les différents modèles de programmation pour architectures parallèles et d'étudier leur adéquation avec la spécification d'application et la problématique de leur exploitation pour un modèle d'exécution
- > au niveau conception et compilation, l'objectif est de passer en revue les principales techniques de compilation exploitant au mieux les ressources d'une architecture parallèle fixe ou d'un circuit spécialisé.

#### 4 crédits ECTS

Volume horaire

Travaux Dirigés : 16h

Travaux Pratiques : 16h

Cours Magistral : 16h

### Objectifs

Etudier les modèles d'exécution, modèles de programmation et langages

- > Streams, Graphes flot de données
- > Processus communicants, FSM
- > modèle de calcul: réseau de kahn
- > MIMD (SM, MP)
- > SIMD, SPMD
- > Équations récurrentes, modèle systolique

Compilation pour le calcul à haute performance

- > Analyse de dépendance, transformation de boucles
- > Parallélisation de nids de boucles : algorithme d'Allen Et Kennedy, approches polyédrales
- > Structures de données et options des compilateurs
- > Analyse de performances
- > Granularité de parallélisme
- > Optimisation de la localité des données en mémoire
- > Optimisation de l'allocation de mémoire
- > Ordonnement et allocation de tâches

### Pré-requis nécessaires

Architectures parallèles

Algorithmique parallèle et distribuée

### Compétences visées

A l'issue de cette UE :

- > On sait identifier le modèle de programmation sous-jacent à un langage à parallélisme explicite
- > On sait exprimer une application dans différents modèles de programmation parallèle
- > On sait analyser les avantages et inconvénients d'un modèle de programmation pour l'expression d'une application parallèle
- > On sait appliquer les principales techniques de compilation pour un modèle d'exécution parallèle fixé
- > On sait utiliser un compilateur pour le calcul haute performance (HPC)

### Bibliographie

- > Algorithmes et architectures systoliques - P.Quinton et Y. Robert, Masson, 1989
- > Algorithmique parallèle - Arnaud Legrand et Yves Robert, Dunod, 2003
- > Multiprocessor Systems-on-chips - Ahmed Jerraya et Wayne Wolf, 2005

### Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/3	
	CT	Ecrit - devoir surveillé	120	2/3	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	100%	

### Langue d'enseignement

---

Français

## S9 LSE Objets Connectés et Sécurité

### Présentation

Réseaux de capteurs sans fil (concepts de base et la gestion de la radio)  
 Programmation de cartes embarquées (Arduino, Raspberry, BeagleBone)  
 Programmation sans fil à base du protocole 802.15.4  
 Développement Android avancé (utilisation des capteurs, Google Maps, etc.)

#### 4 crédits ECTS

Volume horaire  
 Travaux Dirigés : 16h  
 Cours Magistral : 16h  
 Travaux Pratiques : 16h

### Objectifs

Comprendre l'architecture d'une plateforme d'Internet des objets.  
 Comprendre le fonctionnement d'une communication réseau formée de d'objets connectés (Smartphone, tablette, montre, robot, etc.)  
 Comprendre le fonctionnement des réseaux de capteurs sans fil (capteurs, cartes embarquées, modules radios et standard de communication, etc.)  
 Comprendre le fonctionnement du module radio

### Pré-requis nécessaires

Programmation Java et C  
 Programmation de base Android

### Compétences visées

Savoir programmer une carte Arduino  
 Savoir programmer avec une communication sans fil (802.15.4, ZigBee, ...) et des capteurs  
 Savoir programmer Android en mode avancé

### Bibliographie

L'Internet des objets - Les principaux protocoles M2M et leur évolution vers IP - Olivier Hersent  
 Arduino - Maîtrisez sa programmation et ses cartes d'interface (shields) - Christian Tavernier  
 Raspberry Pi 2 - Exploitez tout le potentiel de votre nano-ordinateur (compatible Raspberry Pi 3) - François MOCQ  
 Programmation Android - Damien Guignard  
 Arduino : Applications avancées - Claviers tactiles, télécommande par Internet, géolocalisation - Christian Tavernie

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/3	
	CT	Ecrit - devoir surveillé	120	2/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	100%	

### Langue d'enseignement

Français

## S9 LSE Systèmes d'exploitation pour l'embarqué

### Présentation

Cette UE s'intéresse à la conception et à la mise en place de systèmes d'exploitation, critiques et non critiques, pour l'embarqué

### Objectifs

- Connaître les contraintes liées à l'implémentation des systèmes d'exploitation pour l'embarqué par rapport aux systèmes d'exploitation classiques
- Apprendre le développement Bare-metal
- Approfondir les compétences en programmation système
- Apprendre à appréhender un système d'exploitation existant et en tirer les concepts de bases utilisés
- Apprendre la méthodologie de conception pour créer un système d'exploitation Linux pour l'embarqué
- Prendre en main un ensemble d'outils liés au développement de systèmes d'exploitation pour l'embarqué
- Programmer les systèmes critiques/non critiques avec des contraintes temporelles mous/durs

#### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 32h

Cours Magistral : 16h

### Pré-requis nécessaires

Maîtriser les concepts de bases dans le domaine des systèmes d'exploitation

Programmation C

Outils et frameworks de compilation

### Compétences visées

- Comprendre des différences alternatifs pour mettre en œuvre un système embarqué avec ou sans systèmes d'exploitation
- Comprendre les impacts de la criticité et des contraintes temporelles sur les systèmes d'exploitation

### Descriptif

Partie 1: Systèmes d'exploitation non-critique pour l'embarqué

- Développement Bare-metal
- Systèmes d'exploitation Linux pour l'embarqué
  - + Structure des tâches: processus, threads, etc.
  - + Ordonnancement des tâche, Linux temps réel
  - + Communications inter processus
  - + Une étude de cas sera faite sur le système d'exploitation Linux pour l'embarqué avec manipulation des outils et développement noyau
  - + Construction de noyau linux embarqué et prise en main des outils permettant d'y parvenir

Partie 2: Systèmes d'exploitation critique pour l'embarqué

- Systèmes d'exploitation temps réel
- Partitionnement temporel et spatial
- Standard de certification pour les systèmes embarqués critiques

### Bibliographie

Gerald Coley. BeagleBone Black System Reference Manual, Revision C.3, 2021

### Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/2	
	CT	Ecrit - devoir surveillé	120	1/2	

**Session 2 : Contrôle de connaissances**

<b>Nature de l'enseignement</b>	<b>Modalité</b>	<b>Nature</b>	<b>Durée (min.)</b>	<b>Coefficient</b>	<b>Remarques</b>
	CT	Ecrit - devoir surveillé	120	100%	

**Langue d'enseignement**

---

Français

## S9 LSE System On Chip

### Présentation

Cette UE permet d'aborder la programmation d'un *system-on-chip*

### Objectifs

- Savoir configurer un *system-on-chip*
- Savoir modéliser et implanter une fonction matérielle dans un circuit configurable de type FPGA
- Savoir définir et mettre en œuvre des systèmes de communication entre des blocs fonctionnels disponibles au sein d'un *system-on-chip*
- Savoir écrire un pilote logiciel dédié à la communication avec une fonction matérielle

#### 4 crédits ECTS

Volume horaire

Cours Magistral : 12h

Travaux Dirigés : 12h

Travaux Pratiques : 24h

### Compétences visées

Savoir mettre en œuvre un circuit *system-on-chip*, éventuellement configurable, dans le cadre de systèmes embarqués.

### Descriptif

- > Systèmes sur Puce (SoC): concept, fonctions, interconnexions internes, (bus, NoC)
- > SoC Programmable (configurable) type Xilinx Zynq, technologie FPGA et chaîne de développement, algorithmes de placement/routage
- > Interface logiciel/matériel : échange de données (entrées/sorties), pilotes de périphériques en relation avec une réalisation *hardware* implantée dans la partie FPGA
- > Langage de modélisation matériel VHDL

### Bibliographie

ARM System-on-chip architecture, Steve Furber, Addison-Wesley

The Zynq Book Tutorials, L. Crockett, R. Zlliot, M. Enderwitz, R. Stewart, 2014

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/2	
	CT	Ecrit - devoir surveillé	120	1/2	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	100%	

### Langue d'enseignement

Français

## S9 LSE IA embarquée

### Présentation

Ce cours permet d'aborder les problèmes logiciels liés à la synthèse d'une application IA sur un FPGA-SOC. Les transformations à partir de spécifications de haut niveau jusqu'à une réalisation architecturale sont abordées sous des aspects théoriques (ordonnancement, allocation, complexité, structure à modéliser) et sous des aspects plus pragmatiques avec des études et des utilisations de chaînes de conception commerciales (ou universitaires) permettant d'aborder les techniques de co-design et de synthèse matérielle. Des applications de classification reposant sur l'apprentissage profond (ex CNN) pourront être développées et implémentées sur FPGA à titre d'exemple.

#### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 16h

Cours Magistral : 16h

Travaux Dirigés : 16h

### Objectifs

Comprendre le flot global permettant de migrer d'une spécification comportementale de haut niveau vers une implémentation mixte matérielle/logiciel sur support hybride. Comprendre la nature des contraintes possibles, exprimer les contraintes fonctionnelles et non fonctionnelles. et élaborer des schémas de mise en œuvre d'une implémentation conjointe logicielle/matérielle

Cas d'études issus du domaine de l'IA. Comprendre la mise en place d'algorithmes de décision, de classification, d'apprentissage et diagnostic et comprendre les défis pour les embarquer.

### Descriptif

Techniques de compilation spécifiques pour la mise en place d'accélérateurs matériels pour des applications d'IA:

- > Synthèse système et co-design: partitionnement logiciel/matériel et mapping sur un SOC, exploration architecturale . Synthèse de haut-niveau, les principes: techniques de compilation, CFG, algorithmes d'ordonnancement, allocation, traitement des boucles
- > Mise en relation avec des champs applicatifs de l'IA (Etude des mises en place de diagnostic, de prise de décision et de classifieurs, d'apprentissage profond). Etude de quelques mises en oeuvres matérielles spécifiques (ex CNN).

Illustration sur des projets/outils permettant l'implémentation sur plateforme hétérogène type MPSoC (ex carte ZedBoard de Xilinx ) et interface avec des logiciels de mises en place d'algorithme IA

### Bibliographie

Synthesis and Optimization of Digital Circuits, Giovanni De Micheli, McGraw-Hill, 1994

Embedded System Design, Modeling, Synthesis and Verification, Daniel Gajski and al. , Springer 2009

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/3	
	CT	Ecrit - devoir surveillé	120	2/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	100%	

### Langue d'enseignement

Français avec aide ponctuelle en anglais

## S9 LSE Sûreté de Fonctionnement

### Présentation

Ce cours aborde le problème de la validation des applications embarquées en proposant des méthodes/techniques pour vérifier les propriétés temps-réel, pour évaluer la fiabilité du système et pour mettre en place des mécanismes de sécurité.

### Objectifs

Etre sensibilisé à certaines bonnes pratiques, à certaines propriétés fondamentales de la cryptographie appliquée, à l'utilisation des bibliothèques cryptographiques, à l'analyse des protocoles de sécurisation d'une communication.

Comprendre les techniques et mécanismes pour mettre en place des modules fiables au sein d'un système embarqué.

Savoir modéliser l'architecture du système avec un standard (ex AADL),

Comprendre les problèmes de contraintes de temps dans le cadre du temps réel et savoir réaliser la vérification de ces contraintes.

#### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 16h

Travaux Dirigés : 16h

Cours Magistral : 16h

### Pré-requis nécessaires

Systèmes temps-réel

### Descriptif

Ce cours est constitué de trois parties:

partie I: vérification de propriétés, application au temps réel, modélisation AADL

Pour cette première partie, nous regarderons comment décrire l'architecture du système, tant du point de vue du logiciel que de la plateforme d'exécution. Le langage employé pour ce faire sera AADL, un exemple de standard appliqué par les acteurs du domaine dans le cadre de systèmes embarqués critiques temps réel. Puis, nous regardons comment réaliser la vérification des contraintes de temps (analyse d'ordonnancement temps réel) pour des architectures monoprocesseurs et multiprocesseur).

partie II: Modèles pour l'analyse de fiabilité et techniques de tolérance aux fautes

Dans cette partie, on s'intéresse au monitoring du système et à l'analyse de sa fiabilité. Des techniques de tolérance aux fautes sont abordées pour rendre les architectures matérielles plus robustes (ECC, TMR). Des exemples d'application de ces techniques sont développés.

partie III: Sécurité des systèmes embarqués

Dans cette dernière partie, nous allons étudier certains aspects de la sécurité informatique dans le cadre des systèmes embarqués. Nous reviendrons sur les propriétés de base de la sécurité, puis nous nous intéresserons à la cryptographie utilisée dans ce contexte (en particulier en présentant des attaques spécifiques à l'embarqué, et les protections pour les contrecarrer). Enfin il sera présenté un exemple d'application bien connu, le standard EMV, utilisé lors des paiements via carte à puce.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/3	
	CT	Ecrit - devoir surveillé	120	2/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Ecrit - devoir surveillé	120	100%	

### Langue d'enseignement

Français avec aide ponctuelle en anglais

## Préparation à la vie professionnelle

**6 crédits ECTS**

# Communication

## Présentation

Deux programmes distincts en fonction du choix de la voie orientée professionnalisante ou recherche.

## Objectifs

### Master professionnel

L'objectif est de cibler le marché du travail afin d'affiner son projet professionnel pour trouver l'offre de stage de fin d'études la plus adaptée au cursus et aux objectifs de carrière. Les étudiants sont amenés à se créer un réseau professionnel, à valoriser leur profil universitaire afin de postuler auprès des entreprises.

### Master recherche

L'objectif est d'acquérir une connaissance de soi, des métiers et de l'environnement de la recherche, des débouchés du master, du doctorat ou post-doctorat afin de candidater efficacement, de construire son insertion professionnelle ou sa poursuite d'études dans un contexte concurrentiel.

### 2 crédits ECTS

Volume horaire

Travaux Dirigés : 15h

Cours Magistral : 7h

## Compétences visées

### Master professionnel

Les étudiants sont amenés à se créer un réseau professionnel, à valoriser leur profil universitaire afin de postuler auprès des entreprises. Ils affinent leurs compétences à l'oral pour maîtriser leur entretien de recrutement. Ils développent également leurs compétences rédactionnelles par une préparation à l'élaboration du rapport de stage. Des notions de management et de gestion de projet leur sont dispensées afin qu'ils puissent s'insérer rapidement dans leur équipe professionnelle.

### Master recherche

Les étudiants sont amenés à construire leur projet de doctorat et à appréhender la méthodologie de la thèse par l'utilisation d'outils de recherche et de communication. Ils travaillent à organiser une réflexion personnelle objective à partir d'une recherche bibliographique et d'hypothèses scientifiques et développent leur esprit critique. Ce travail est complété par une réflexion sur la propriété intellectuelle, la fraude ou l'erreur scientifique, l'analyse des mécanismes de l'innovation, les enjeux des rapports entre scientifiques et société et l'éthique de la recherche dans un contexte compétitif. Les travaux comportent des exposés sur le monde de la recherche, la présentation d'un état de l'art en lien médiatisant leur sujet de stage de recherche et une présentation de leur projet professionnel.

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CC	Ecrit et/ou Oral		1/1	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Oral	15	1/1	

# Anglais

**3 crédits ECTS**

Volume horaire

Cours Magistral : 8h

Travaux Dirigés : 14h

## Modalités de contrôle des connaissances

---

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Ecrit et/ou Oral		100	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CT	Oral	15	100	

## Entreprise

**1 crédits ECTS**

Volume horaire

Autres : 10h

### Modalités de contrôle des connaissances

---

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
Autres	Autre modalité	Ecrit et/ou Oral		1/1	validation par "badge" - Pas de session 2 -

## S10 LSE Ouverture scientifique LSE

### Présentation

Ce module permet d'aborder des aspects complémentaires de la formation sous forme de conférences thématiques faisant intervenir des experts (industriels ou académiques) du domaine.

### Objectifs

Aborder les nouveautés dans le domaine de l'informatique ou des domaines connexes. Éventuellement approfondir des thèmes abordés dans la formation.

Permettre aux étudiants d'assister à des conférences, dispensées dans ou hors de l'Université, et ayant trait aux nouvelles technologies en informatique pour le traitement de la donnée environnementale.

Renforcer la connaissance du domaine métier et rester au contact des évolutions de la discipline.

**4 crédits ECTS**

Volume horaire

Travaux Dirigés : 24h

### Descriptif

Conférences principalement dispensées sur une journée ou ½ journée.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Ecrit et/ou Oral		100%	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Ecrit et/ou Oral		100%	Pas de session 2

### Langue d'enseignement

Français avec aide ponctuelle en anglais

## S10 LSE Projet de Recherche

**6 crédits ECTS**

# S10 LSE Atelier de développement collaboratif

## Présentation

Développement d'un pilote de périphérique avec gestion collaborative des versions.

**2 crédits ECTS**

Volume horaire

Travaux Pratiques : 20h

## Pré-requis nécessaires

Maitrise du langage C

## Compétences visées

Etre en mesure d'intégrer rapidement une entreprise utilisant un outil de gestion de versions pour le développement ou la maintenance en équipe d'un logiciel.

Etre capable d'utiliser un outil de gestion de version dans le cadre d'un développement logiciel collaboratif

Connaître le rôle d'un pilote de périphérique, et être capable d'en assurer le développement

## Descriptif

Outil de gestion de version (GIT)

- > Commandes de base pour la gestion de version
- > Documentation des évolutions d'un logiciel
- > Création et fusion de branches de développement

Pilotes de périphérique

- > Rappel sur la notion de pilote (rôle, situation dans le système, structure)
- > Développement en langage C d'un pilote de périphérique simple

## Modalités de contrôle des connaissances

### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Autre nature		100%	projet

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Autre nature		100%	

## Langue d'enseignement

Français

## S10 LSE Projet de recherche encadré (6 semaines)

### Présentation

Développement d'un projet en relation avec les activités de recherche du domaine et avec les enseignements de la formation (projet sur 8 semaines).

4 crédits ECTS

### Objectifs

S'approprier un sujet de projet et mettre en place des solutions techniques.

### Compétences visées

Savoir s'organiser sur un projet de 8 semaines. Etre capable de mobiliser ses compétences pour proposer des solutions et les mettre en œuvre.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/2	
	CC	Oral - soutenance	30	1/4	
	CC	Ecrit - rapport		1/4	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Autre nature			Pas de session 2

### Langue d'enseignement

Français

## S10 LSE Stage (4 à 6 mois) ou TAL (temps d'accueil en laboratoire)

### Présentation

Mise en situation réelle au sein d'une entreprise ou d'un laboratoire à travers le développement d'un projet informatique proposé par une entreprise ou un laboratoire et mené au sein de l'entité.

**20 crédits ECTS**

Volume horaire

Stages : 16h

### Objectifs

Éprouver les connaissances acquises au cours de la formation en situation réelle d'entreprise à travers un projet informatique proposé par une entreprise et mené au sein de cette dernière.

Le stage de fin d'études doit permettre aux étudiants de commencer leur insertion dans le monde du travail. Il agit comme une période de transition entre le monde universitaire et le monde professionnel. Pour certains étudiants, ce stage sert de stage de pré-embauche.

### Pré-requis nécessaires

Maîtriser les outils et méthodes du développement logiciel pour l'embarqué.

### Compétences visées

Savoir mener un travail de développement logiciel en tant qu'ingénieur informatique. :

- Capacité à appréhender le cycle de vie d'un projet de développement conséquent et réaliste.
- Capacité à intégrer une équipe de développement.
- Capacité à mobiliser ses connaissances théoriques et pratiques pour résoudre un problème d'ingénierie informatique.
- Capacité à faire un état de l'art ou une veille technologique.

### Modalités de contrôle des connaissances

#### Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	CC	Travaux Pratiques		1/3	
	CC	Oral - soutenance	30	1/3	
	CC	Ecrit - rapport		1/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
	Report de notes	Autre nature			pas de session 2

### Langue d'enseignement

Français