

## Licence mention Informatique

### Parcours Conception et Développement d'Applications

#### Objectifs

Les objectifs de la licence d'informatique sont l'acquisition des compétences fondamentales, méthodes et savoir-faire techniques représentatifs des différentes tâches de la discipline informatique. Cette formation couvre l'ensemble de la discipline informatique : fondements, architecture et matériel, systèmes, méthodes et technologies logicielles, applications informatiques, systèmes d'information.

Le parcours CDA est un parcours d'une année destiné tout particulièrement aux étudiants titulaires d'un BTS SIO ou BTS SN option Informatique qui souhaitent préparer une poursuite d'étude en Master Informatique. Le premier semestre comprend ainsi plusieurs unités d'enseignement spécifiques permettant aux étudiants d'approfondir les matières fondamentales de l'informatique en s'appuyant sur les compétences opérationnelles acquises lors de leur formation précédente. Le parcours CDA peut également s'adresser aux étudiants titulaires de bac+2 dans le domaine des sciences du numérique devant approfondir leurs connaissances dans la conception et le développement logiciel.



#### Compétences acquises

- > Analyse, modélisation d'un problème, conception et mise en œuvre d'une solution informatique adaptée en choisissant des outils adéquats.
- > Conception d'IHM, d'applications Web, de systèmes d'information.
- > Apprentissage de différents langages de programmation et systèmes d'exploitation.

La formation comprend également un stage disciplinaire de 2 mois minimum en fin d'année.

#### Conditions d'accès

Le parcours CDA est accessible sur dossier par validation d'études (BTS SIO, IRIS, SN option Informatique, ou équivalent), directement en troisième année.

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

#### Poursuites d'études

Le parcours Conception et Développement d'Applications permet de poursuivre des études en master à dominante informatique, en particulier dans les différents parcours du master d'informatique de l'université de Brest. Il permet également de candidater dans des formations bac+4 à dominante informatique, comme des formations d'ingénieur.

#### Insertion professionnelle

Le diplômé peut exercer dans des entreprises de services du Numérique ou dans le service informatique d'une entreprise (quelque soit le secteur d'activité de celle-ci).

Il peut exercer les emplois suivants :

- > Développeur informatique (systèmes d'information, applications Web, ...)
- > Assistant fonctionnel/Technicien support
- > Administrateur systèmes et réseaux
- > Administrateur de bases de données

#### Infos pratiques

**Faculté des Sciences et Techniques** à Brest

Ouvert en stage

#### Contacts

**Responsable pédagogique**

LEMARCHAND Laurent

[Laurent.Lemarchand@univ-brest.fr](mailto:Laurent.Lemarchand@univ-brest.fr)

Tel. 02 98 01 69 45

**Responsable Secrétariat pédagogique**

Secrétariat du département informatique

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

Tel. 02 98 01 62 06 / 80 11

## Programme

### Licence 3ème année

#### Semestre 5

**Programmation C avancée** 54h

**C, algorithmique et structures de données**

- Langage C 28h
- Algorithmes et Structures de Données 26h

**Informatique théorique et automates** 54h

**Java et Conception d'Applications**

- Java 2 36h
- Conception d'applications 20h

**Bloc transversal S5 CDA**

- Anglais S5 16h
- Communication S5 12h
- Adaptation : Introduction à l'objet et aux systèmes d'exploitation et microprocesseur 36h

#### Semestre 6

**Systèmes d'information et Web** 54h

**Projet applicatif** 70h

**Architectures et Systèmes 2**

- Architectures 2 28h
- Systèmes 2 28h

**Réseaux IP**

- Réseaux IP : programmation des réseaux 28h
- Réseaux IP : RPC 16h
- Réseaux IP : IHM 27h

**Bloc transversal S6 IFA-CDA**

- Stage (8 semaines) 280h
- Anglais S6 16h
- Communication S6 12h

Dernière mise à jour le 01 mars 2022

## Programmation C avancée

### Présentation

Le but de cette UE est d'approfondir les connaissances des étudiants en C et d'appliquer ces connaissances dans le cadre d'un projet.

#### 6 crédits ECTS

Volume horaire

Travaux Pratiques : 32h

Travaux Dirigés : 12h

Cours Magistral : 10h

### Objectifs

- > Maîtriser la syntaxe et de la sémantique du langage C.
- > Savoir développer et déboguer une application en langage C.

### Pré-requis nécessaires

Connaissances de l'algorithmique et de la programmation en C, correspondant aux UE [Algorithmique et programmation](#) et [Langages de programmation](#) (parcours IFA), ou à l'UE [C, Algorithmique et Structures de Données](#) (parcours CDA).

### Compétences visées

L'étudiant doit disposer, parmi l'ensemble de ceux qu'il maîtrise, d'un langage adapté à la programmation de systèmes disposant d'un support d'exécution restreint en terme des ressources matérielles et/ou de services logiciels disponibles.

### Descriptif

#### 1. Langage C avancé :

Rappel sur les classes d'allocations.

Codage de structures de données complexes.

Pointeurs de fonctions, fonctions "génériques".

Conversions de types, implicites et explicites, portabilité du code.

Fonctions à nombre variables d'arguments, mécanisme de passage. des arguments à une fonction.

E/S haut niveau, contrôle des périphériques.

#### 2. Environnement de programmation :

Compilation séparée.

Outils make, débogeur symbolique.

Éditions de liens, construction et utilisation de bibliothèques statiques et dynamiques.

Introduction à l'internationalisation.

3. Développement d'une application simple en langage C, intégrant si besoin est, une IHM graphique élémentaire (GTK) et un accès à une base de données.

### Bibliographie

1. B. Kernighan, R. Pike, **La programmation en pratique**, Vuibert informatique, 2001
2. A. Braquelaire, **Méthodologie de la programmation en C (Norme C 99 - API POSIX)**, Dunod, 2005
3. B.W. Kernighan, R. Pike, David Odín, **Programmation Linux avec GTK+**, 2000, Eyrolles

### Modalités de contrôle des connaissances

**Session 1 ou session unique - 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>
UE	CC	Travaux Pratiques		1/2	
UE	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>
UE	CT	Ecrit - devoir surveillé	120	1	Pas de report de CC, même favorable.

# C, algorithmique et structures de données

## Objectifs

---

- > Connaissance des principales structures de données avec leurs algorithmes associés.
- > Compétences initiales en C permettant de suivre l'UE [Programmation C avancée](#)

**5 crédits ECTS**

# Langage C

## Présentation

Remise à niveau en langage C pour suivre l'UE [Programmation C Avancée](#)

- > Eléments syntaxiques de base du langage C,
- > Types simples
- > Structures de contrôle,
- > Fonctions, passage par valeur et par adresse
- > Tableaux et structures.

### 2 crédits ECTS

Volume horaire

Travaux Dirigés : 8h

Travaux Pratiques : 12h

Cours Magistral : 8h

## Bibliographie

**Le langage C**, BW Kernighan et DM Ritchie, 2eme éd., Masson, 1992

## Modalités de contrôle des connaissances

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

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

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	60	1	Pas de report de CC, même favorable.

# Algorithmes et Structures de Données

## Présentation

Le but de cette UE est de rappeler les éléments d'algorithmique essentiels pour la programmation impérative, et ainsi acquérir ou renforcer les pré-requis aux autres UE de l'année.

Il fournit des bases en algorithmique permettant de traiter des collections de données

- > Principes d'induction/récurrance, preuve par récurrence,
- > Application pour la définition de types inductifs (introduction des principales structures de données : listes et arbres, piles, files) et l'écriture d'algorithmes récursifs (parcours, tris),
- > Itération et tableaux : écriture d'algorithmes itératifs sur des tableaux (parcours, tris). Comparaison avec les versions récursives.
- > Introduction à la complexité sur les algorithmes de tri classiques.

### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 6h

Travaux Dirigés : 10h

Cours Magistral : 10h

## Pré-requis nécessaires

Pour aborder ce cours, il est utile d'avoir déjà programmé dans un langage de programmation impérative et d'avoir un minimum de connaissances sur :

- > les types, les variables, les opérateurs
- > les instructions de contrôle (appels, boucles, conditionnelles, fonctions, procédures)

## Compétences visées

- > écrire un algorithme impératif itératif ou récursif répondant à un problème simple
- > tracer l'exécution d'un algorithme/programme
- > évaluer la complexité d'un algorithme/programme

## Modalités de contrôle des connaissances

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

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

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	60	1	Pas de report de CC, même favorable.

## Informatique théorique et automates

### 6 crédits ECTS

Volume horaire

Travaux Pratiques : 18h

Cours Magistral : 18h

Travaux Dirigés : 18h

### Descriptif

- > Logique : raisonnement par induction, logique propositionnelle, systèmes déductifs.
- > Automates : détermination, langages réguliers, minimisation.
- > Graphes : bases de l'algorithmique des graphes et optimisation.

### Bibliographie

1. *Introduction à la logique*, R. DAVID, K. NOUR, C. RAFFALLI, Dunod.
2. *Introduction à la calculabilité*, P. Wolper, Masson.
3. *Optimisation combinatoire T2: programmation discrète*, M. Sakarovitch, Hermann, 1984.
4. *Méthodes d'optimisation*, I. Charon et al., Masson, 1996.

### Modalités de contrôle des connaissances

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CC	Travaux Pratiques		1/4	
UE	CT	Ecrit - devoir surveillé	90	1/2	partie logique et automates
UE	CT	Ecrit - devoir surveillé	60	1/4	partie graphes

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	120	1	Pas de report de CC, même favorable.



# Java et Conception d'Applications

## Présentation

---

UE constituée de deux EC obligatoires : Java 2 et Conception d'Applications.

**6 crédits ECTS**

## Java 2

### Présentation

Ce cours revient sur les concepts fondamentaux de la programmation objet : l'encapsulation, la composition et l'héritage. En complément, ce cours aborde les concepts de classe enfouie, de classe abstraite, d'interface, d'introspection, de clonage, d'exception, de généricité et de lambda-expression. Le langage Java sert de support à l'apprentissage de ces concepts. Le cours aborde également les éléments indispensables de l'API pour le développement d'applications en Java comme les collections, les flots de données ou les fils d'exécution. Enfin, le cours présente les principaux outils de développement associés à Java : outils de tests unitaires et de non-régression, générateurs de documentation, outils de contrôle de la qualité du code, et outils de travail collaboratif.

L'UE est organisée en 10 leçons comportant chacune un cours, une à deux séances de travaux dirigés et une à deux séances de travaux pratiques :

1. Rappels de Java : fondements impératifs, fondements Objet (encapsulation, composition, héritage)
2. Classes enfouies, classes abstraites, interfaces
3. Introspection, clonage, égalité, représentation littérale
4. Exceptions : principes généraux et mise en oeuvre en Java
5. Généricité : polymorphisme et types génériques
6. Objets fonctionnels : lambda-expressions
7. Collections : API standard, collections simples et tableaux associatifs
8. Flots de données : flots d'octets et de caractères, gestion de fichiers et analyse lexicale
9. Fils d'exécution : principes élémentaires de programmation concurrente
10. Outils pour le développement d'applications : JUnit, Javadoc, Doxygen, Checkstyle, SVN, Git

### Objectifs

L'objectif est quadruple :

- > savoir tirer parti de la programmation objet pour la conception et la maintenance de programmes modulaires, réutilisables, évolutifs et sûrs
- > compléter les éléments méthodologiques d'analyse, de conception et de programmation orientées objet
- > renforcer la maîtrise du langage Java, de ses constructions élémentaires, et de son API standard
- > découvrir l'outillage essentiel au développement d'une application d'envergure
  
- > maîtriser le langage Java : sa syntaxe et ses fondements sémantiques
- > connaître les différentes phases de la compilation d'un programme Java
- > connaître le mécanisme d'exécution d'un programme Java et le fonctionnement de la machine virtuelle
- > maîtriser les rudiments de la programmation objet
- > connaître quelques bonnes pratiques de conception objet

D'autre part, concernant l'outillage pour la conduite des projets, les étudiants sont amenés à expérimenter et à comprendre l'intérêt et les principes des outils :

- > de validation par tests unitaires
- > de documentation logicielle
- > de contrôle de la qualité du code
- > de gestion de versions et travail collaboratif

### Pré-requis nécessaires

Pour aborder ce cours, il est nécessaire d'avoir déjà programmé en Java ou dans un langage de programmation Objet similaire, et d'avoir un minimum de connaissances sur :

- > les aspects procéduraux de Java (types, variables, opérateurs, appels, boucles, conditionnelles) et l'algorithmique élémentaire (tri, recherche de max, calcul, etc.)
- > les fondements de la programmation Objet (encapsulation, composition, héritage)

#### 4 crédits ECTS

Volume horaire

Travaux Pratiques : 12h

Cours Magistral : 12h

Travaux Dirigés : 12h

- > les principes élémentaires de conception Objet (identification des classes, assignation de responsabilités, structure, composition et comportement des objets)

L'UE Java 1 de L2 et l'UE Introduction à l'objet et aux systèmes d'exploitation et microprocesseur en L3 fournissent ces pré-requis.

## Compétences visées

---

Traduire la spécification d'un programme en système d'objets en assignant des responsabilités.

Concevoir un programme à la fois ouvert à l'extension et fermé à la modification.

Mobiliser à bon escient les constructions du langage Java et de son API.

Savoir utiliser les outils de tests unitaires, les générateurs de documentation, les outils de contrôle de la qualité du code, et les outils de travail collaboratif.

## 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
UE	CT	Ecrit - devoir surveillé	120	2/3	
UE	CC	Travaux Pratiques		1/3	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	120	1	Pas de report de CC, même favorable.

# Conception d'applications

## Présentation

Cette UE a pour objectif la maîtrise des outils du développeur (codage, test et déploiement) pour réaliser une application déployable et utilisable par un utilisateur final. L'application visée couvre plusieurs technologies (développement objet, développement Web, base de données).

L'UE est organisée sous forme de projet à réaliser en groupe de 4 à 5 étudiants selon une méthode itérative basée sur plusieurs Sprints.

### 2 crédits ECTS

Volume horaire

Cours Magistral : 2h

Travaux Dirigés : 6h

Travaux Pratiques : 12h

## Pré-requis nécessaires

Programmation Java. Programmation Web. Programmation Base de données. Mise en place de tests unitaires.

## Compétences visées

Savoir développer une application. Savoir spécifier et mettre en place un IHM. Savoir décrire et mettre en place une base de données. Savoir décrire et documenter le code. Maîtriser la définition et la mise en place de tests unitaires. Maîtriser la qualité du code. Maîtriser les outils du développeur (IDE, suivi de version, vérification de la qualité du code, spécification du code, mise en œuvre de tests unitaires).

## 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	Travaux Pratiques		1	note reportée en session 2

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	Report de notes	Travaux Pratiques		1	report de session 1

## Bloc transversal S5 CDA

**7 crédits ECTS**

## Anglais S5

**2 crédits ECTS**

Volume horaire

Travaux Dirigés : 16h

### Modalités de contrôle des connaissances

---

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
Travaux Dirigés	CC	Ecrit et/ou Oral		30/100	
Travaux Dirigés	CT	Ecrit - devoir surveillé	90	70/100	

#### Session 2 : Contrôle de connaissances

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

## Communication S5

**2 crédits ECTS**

Volume horaire

Travaux Dirigés : 12h

### Modalités de contrôle des connaissances

---

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

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

#### Session 2 : Contrôle de connaissances

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

# Adaptation : Introduction à l'objet et aux systèmes d'exploitation et microprocesseur

## Présentation

Pour la partie **Introduction à l'objet**, ce cours permet de rattraper l'UE Java 1 de L2, pré-requis de l'UE Java 2 de L3. Ce cours d'initiation aborde trois concepts fondamentaux de la programmation objet : l'encapsulation, la composition et l'héritage. Le langage Java sert de support à l'apprentissage par la mise en pratique de ces concepts.

L'UE est organisée en 3 leçons comportant chacune une séance de travaux dirigés et une à deux séances de travaux pratiques :

1. Encapsulation : modèle de données structurées, enfouissement, initialisation, consultation, mise à jour, interface d'interactions
2. Composition : références d'objet, changement d'état, interaction, délégation
3. Héritage : extension de structure, extension de comportement, mise à jour de comportement, liaison dynamique, polymorphisme

Pour la partie **systèmes d'exploitation et microprocesseurs**,

Le cours débute par une description de l'architecture des ordinateurs et du fonctionnement d'un processeur basique. Dans un second temps, le rôle et les fonctionnalités d'un système d'exploitation sont précisés, puis la notion de processus en tant que support de l'exploitation multi-tâches, est présentée.

Ce cours couvre les pré-requis nécessaires pour suivre l'UE "Architecture et Système 2" du niveau L3 :

- > Machine à programme enregistré, processeurs séquentiels
- > Langage machine et assembleur (Intel64)
- > Fonctions et structure générale des systèmes d'exploitation
- > Notion de processus, gestion des ressources partagées par sémaphore

## Objectifs

Partie **Introduction à l'objet**

L'objectif est triple :

- > mettre en évidence l'intérêt essentiel de la programmation objet, i.e. sa nature intrinsèquement modulaire et extensible
- > donner les premiers éléments méthodologiques de conception qui permettent de déterminer les objets utiles à la réalisation d'un programme en précisant leurs responsabilités et leurs interactions
- > consolider la maîtrise du langage Java, de ses constructions élémentaires, et de son API standard

## Pré-requis nécessaires

Pour aborder ce cours, il est utile d'avoir déjà programmé dans un langage de programmation procédural et d'avoir un minimum de connaissances sur :

- > les types, les variables, les opérateurs
- > les instructions de contrôle (appels, boucles, conditionnelles)
- > l'algorithmique élémentaire (tri, recherche de max, calcul, etc.)
- > les éléments constitutifs d'un ordinateur

## Compétences visées

Partie **Introduction à l'objet**

- > Traduire la spécification d'un programme en système d'objets en assignant des responsabilités.
- > Concevoir un programme à la fois ouvert à l'extension et fermé à la modification.
- > Mobiliser à bon escient les constructions du langage Java et de son API.

### 3 crédits ECTS

Volume horaire

Travaux Dirigés : 20h

Travaux Pratiques : 8h

Cours Magistral : 8h



Partie **Systèmes d'exploitation et microprocesseurs**

- > Comprendre comment les services fournis par le matériel d'un ordinateur sont rendus accessibles aux applications numériques.
- > Etre sensibilisé à la problématique du partage des ressources matérielles d'un ordinateur entre plusieurs applications.

## Bibliographie

---

Partie **Introduction à l'objet**

- > 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
UE	CC	Travaux Pratiques		1	pour la partie objet
UE	CC	Travaux Pratiques		1/4	pour la partie microprocesseurs
UE	CT	Ecrit - devoir surveillé	60	1/4	pour la partie microprocesseurs

## Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Oral	30	1/2	

# Systèmes d'information et Web

## Présentation

### 5 crédits ECTS

Volume horaire

Travaux Dirigés : 18h

Travaux Pratiques : 18h

Cours Magistral : 18h

- > Rappels : BD et SGBD.
- > Respect d'un cahier des charges.
- > Introduction aux méthodologies AGILE.
- > Introduction à l'architecture client/serveur.
- > Conception d'une base de données relationnelle (diagramme UML de classes) puis passage du modèle conceptuel au modèle logique (schéma relationnel).
- > Langage SQL
  
- > Langage de Définition de Données (LDD)
- > Langage de Manipulation de Données (LMD)
- > SQL avancé : vue, procédure, trigger.
  
- > Programmation, côté serveur, d'une application Web en utilisant un framework MVC.
- > Réalisation de tests unitaires et de tests de validation.

## Pré-requis nécessaires

- > Création de pages Web statiques (HTML / CSS / Javascript)
- > Création d'une base de données relationnelle, insertion du jeu de données et manipulation des données

## Compétences visées

- > Concevoir et implanter une base de données sur un SGBD relationnel (mysql, MariaDB, Oracle ...) en utilisant le langage SQL.
- > Manipuler les données de la base de données relationnelle en utilisant le langage SQL.
- > Développer, côté serveur, l'application Web permettant de manipuler les données de la base de données (utilisation d'un framework MVC).
- > Travailler en mode projet.

## Bibliographie

- > Jean-Luc HAINAUT, Bases de données, Dunod 2009
- > F. BROUARD, R. BRUCHEZ, C. SOUTOU, "SQL", Pearson, 2012
- > Christian SOUTOU, Modélisation des Bases de Données, Eyrolles, 2015
- > Pascal ROQUES, UML2 par la pratique, Eyrolles, 2005
- > Signataires du manifeste Agile, "<https://agilemanifesto.org/>", consulté le 02/02/2022

## Modalités de contrôle des connaissances

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CC	Travaux Pratiques		40%	
UE	CT	Ecrit - devoir surveillé	120	60%	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	120	1	Pas de report de CC, même favorable.

## Projet applicatif

### Présentation

Cette UE consiste en un ou deux projets développés en intensif, par groupe de 2 à 4, pendant deux semaines en continu. L'objectif pédagogique est d'apprendre à élaborer un développement applicatif adéquat d'un système et de savoir le présenter. L'objectif pratique du problème à résoudre est ambitieux. Il s'agit d'un projet transversal, tourné vers la maturation et le croisement des connaissances acquises pendant la licence. Le problème peut potentiellement être résolu de façon diverse pour valoriser la créativité. La démarche pédagogique consiste à placer les étudiants dans une situation de recherche, dans laquelle ils doivent s'investir, se poser des questions, faire des choix, trouver des solutions et apprendre à les argumenter.

Le cahier des charges est présenté en début de semaine par un énoncé général complété par un ensemble de sous-problèmes à résoudre et la présentation des artefacts à produire. Cette présentation peut éventuellement introduire une chronologie à respecter pour la résolution des sous-problèmes, des exemples et des démonstrations fonctionnelles.

En fin de projet, les étudiants doivent être capables de présenter leur mise en œuvre au cours d'un exposé de 15 à 20 minutes avec démonstration.

#### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 48h

Autres : 22h

### Objectifs

- > Travailler en équipe et en réseau ainsi qu'en autonomie et responsabilité au service d'un projet;
- > Choisir, sur des critères objectifs, les structures de données et construire les algorithmes les mieux adaptés à un problème donné;
- > Analyser et interpréter les résultats produits par l'exécution d'un programme;
- > Appliquer des approches raisonnées de résolution de problèmes complexes par décompositions et/ou approximations successives et mettre en œuvre des méthodes d'analyse pour concevoir des applications et algorithmes à partir d'un cahier des charges partiellement donné;
- > Concevoir le traitement informatisé d'informations de différentes natures, telles que des données, des images et des textes;
- > Expliquer et documenter la mise en œuvre d'une solution technique.

### Pré-requis nécessaires

Pour aborder cette UE, il faut avoir suivi toutes les UE de la licence. En particulier celles qui visent directement le développement d'applications :

- > L'algorithmique et les structures de données;
- > Les langages de programmation comme le C, Python et Java;
- > L'ingénierie des systèmes d'information et la conception d'application;
- > Le développement des IHM.

### Modalités de contrôle des connaissances

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CC	Travaux Pratiques		1	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	Report de notes	Autre nature		1	

## Architectures et Systèmes 2

**6 crédits ECTS**

## Architectures 2

### Présentation

Le cours décrit deux structures fondamentales utilisées dans les processeurs actuels : le pipeline et la hiérarchie mémoire. La problématique de la mesure des performances d'un processeur est introduite.

- > technique de pipeline appliquée à la réalisation d'un processeur RISC de première génération : principe, aléas de données, de contrôle, structurelle, prédiction de branchement
- > hiérarchie mémoire : cache et mémoire virtuelle
- > mesures de performances

Au cours des travaux pratiques, les étudiants sont invités à modéliser une partie d'un processeur RISC *pipeliné* en utilisant le langage de description de matériel VHDL, puis à simuler le comportement de ces modèles.

#### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 10h

Travaux Dirigés : 8h

Cours Magistral : 10h

### Compétences visées

- > Comprendre la signification des caractéristiques techniques et de performances d'un ordinateur
- > Evaluer l'adéquation d'un algorithme et une architecture matérielle de processeur

### Bibliographie

1. D. Patterson, J. Hennessy, *Organisation et conception des ordinateurs (l'interface matériel/logiciel)*, Dunod
2. *Operating Systems : Internals and Design Principles*, William Stallings - Prentice Hall
3. *Programmation systèmes en C sous Linux: Signaux, Processus, threads, IPC et sockets*, Christophe Blaess, Eyrolles
4. *Unix, programmation et communication*, Jean-Marie Rifflet, Dunod

### Modalités de contrôle des connaissances

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CC	Travaux Pratiques		1/3	
UE	CT	Ecrit - devoir surveillé	90	2/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	60	1	Pas de report de CC (même favorable).

## Systèmes 2

### Présentation

- > programmation *multi-threads*
- > ordonnancement,
- > systèmes de fichiers et gestion des ressources de mémoire
- > utilisation en langage C des primitives systèmes

#### 3 crédits ECTS

Volume horaire

Cours Magistral : 10h

Travaux Dirigés : 10h

Travaux Pratiques : 8h

### 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	Travaux Pratiques		1/3	
EC	CT	Ecrit - devoir surveillé	90	2/3	

#### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Ecrit - devoir surveillé	60	1	Pas de report de CC (même favorable)

## Réseaux IP

**8 crédits ECTS**

# Réseaux IP : programmation des réseaux

## Présentation

### Programmation réseau en C

- > architecture des réseaux informatiques, adressage des machines
- > modèles réseau en couches : principes, couches et protocoles réseau,
- > protocoles TCP et UDP, notion de socket
- > programmation en C d'applications clientes et serveur à l'aide des primitives système et réseau (exemple de champ d'application possible : échange de données avec des objets connectés).

### 3 crédits ECTS

Volume horaire

Travaux Pratiques : 9h

Travaux Dirigés : 9h

Cours Magistral : 10h

## Pré-requis nécessaires

- > Architecture et système (S4)
- > Programmation C avancée (S5)

## Compétences visées

- > connaître le modèle client-serveur utilisé dans le cadre des SE distribués,
- > connaître les caractéristiques des protocoles TCP et UDP et être capable de programmer des applications de type client/serveur,
- > être sensibilisé à l'architecture d'un réseau local et à l'interconnexion des réseaux (ex : Internet).

## Bibliographie

1. "Unix, programmation et communication", Jean-Marie Rifflet, Dunod
2. "TCP/IP : architecture, protocoles, applications" - D. Comer - Dunod
3. "Cours Réseaux et télécoms" - G. Pujolle - Eyrolles
4. "Cours Réseaux" - V. Ribaud et P. Le Parc - Polycoopié de cours du département Informatique, UBO, Brest.

## Modalités de contrôle des connaissances

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

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

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	60	1	Pas de report de CC (même favorable).



# Réseaux IP : RPC

## Présentation

Introduction aux systèmes répartis

- > Définitions
- > Caractéristiques des systèmes répartis
- > Systèmes répartis vs. systèmes centralisés
- > Mise en oeuvre des applications réparties

Introduction à RPC

- > Quelques rappels
- > Présentation de RPC
- > Fonctionnement général de RPC

Le protocole RPC

- > RPC & XDR
- > Le langage RPC
- > Fonctionnement de RPC
- > Gestion des erreurs
- > Données transmises
- > Accès aux serveurs et aux services

### 2 crédits ECTS

Volume horaire

Cours Magistral : 4h

Travaux Dirigés : 4h

Travaux Pratiques : 8h

## Pré-requis nécessaires

Programmation C Avancée

## Compétences visées

- > Introduction à RPC : Remote Procedure Call
- > Développement d'une application répartie client/serveur en utilisant le protocole RPC
- > Mise en œuvre de l'application avec l'outil RPCgen (en langage C sous linux).

## Bibliographie

- > aux systèmes répartis, Frank Singhoff, UBO, 2011
- > Technologie et architectures Internet, Pierre-Yves Cloux & al., DUNOD, 2002
- > Algorithmes distribués e

## 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	Travaux Pratiques		1/2	
EC	CT	Ecrit - devoir surveillé	60	1/2	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CT	Ecrit - devoir surveillé	60	1	pas de report de CC

# Réseaux IP : IHM

## Présentation

L'accent est mis sur les points suivants :

1. Les flots de données, pré-requis à l'utilisation des sockets pour la communication via des fichiers le réseaux ainsi que la communication série (port RS232, bluetooth, sans fil, etc.)
2. La programmation des sockets en Java (comparaison/différence avec le C),
3. Les threads et la synchronisation. Cette partie est indispensable dans toute application Java. Aussi, elle permet de gérer les accès aux fichiers/ressources (plusieurs utilisateurs accédant en parallèle à une même variable ou via le réseau au même fichier),
4. La création d'une interface utilisateur : nécessite des notions sur la structuration des applications en packages et les IHM avec Swing et Java FX,
5. Introduction à la sérialisation et au protocole de communication MQTT (API Paho de Eclipse)

### 3 crédits ECTS

Volume horaire

Cours Magistral : 9h

Travaux Dirigés : 9h

Travaux Pratiques : 9h

## Objectifs

Cet EC permet d'introduire les concepts de programmation Java permettant de développer des applications complètes intégrant à la fois des aspects orientés réseaux (Web et Sockets) et la conception d'interfaces utilisateur (Swing et JavaFX). Elle doit permettre à un étudiant de développer des applications clients/serveurs de type Chat, client FTP, etc.

## Pré-requis nécessaires

programmation Java

## Compétences visées

Java réseau et IHM

## 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	60	1/2	
Travaux Pratiques	CC	Travaux Pratiques		1/2	

### Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	Autre modalité	Ecrit - devoir surveillé	60	1	

## Bloc transversal S6 IFA-CDA

**8 crédits ECTS**

## Stage (8 semaines)

### Présentation

Un stage de 8 semaines minimum complète le dernier semestre de licence informatique. Ce stage (essentiellement en entreprise) doit permettre à l'étudiant d'appliquer les enseignements suivis au cours de l'année.

Le stage est évalué à la fois par le tuteur en entreprise et par la rédaction d'un rapport ainsi qu'une soutenance orale.

**4 crédits ECTS**

Volume horaire

Stages : 280h

### Modalités de contrôle des connaissances

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CC	Travaux Pratiques		1/3	Note de travail en entreprise
UE	CT	Ecrit - rapport		1/3	
UE	CT	Oral - soutenance	15	1/3	

## Anglais S6

**2 crédits ECTS**

Volume horaire

Travaux Dirigés : 16h

### Modalités de contrôle des connaissances

---

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

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

#### Session 2 : Contrôle de connaissances

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

## Communication S6

### 2 crédits ECTS

Volume horaire

Travaux Dirigés : 12h

### Modalités de contrôle des connaissances

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

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CC	Ecrit et/ou Oral		50/100	
UE	CT	Ecrit - devoir surveillé	60	50/100	

#### Session 2 : Contrôle de connaissances

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