Université de Toulouse

Faculté sciences et ingénierie Université de Toulouse

PÉRIODE D'ACCRÉDITATION : 2022 / 2026

UNIVERSITE DE TOULOUSE

SYLLABUS MASTER

Mention Informatique

M1 Sciences du Logiciel

http://www.fsi.univ-tlse3.fr/
https://departement-informatique.univ-tlse3.fr/master-mention-informatique

2026 / 2027

20 NOVEMBRE 2026

SOMMAIRE

SCHÉMA ARTICULATION LICENCE MASTER 3

PRÉSENTATION 4

PRÉSENTATION DE LA MENTION 4

Mention Informatique 4

PRÉSENTATION DE L'ANNÉE DE M1 Sciences du Logiciel 4

RUBRIQUE CONTACTS 5

CONTACTS PARCOURS 5

CONTACTS MENTION 5

CONTACTS DÉPARTEMENT : FSI.Info 5

Tableau Synthétique des UE de la formation 6

LISTE DES UE 9

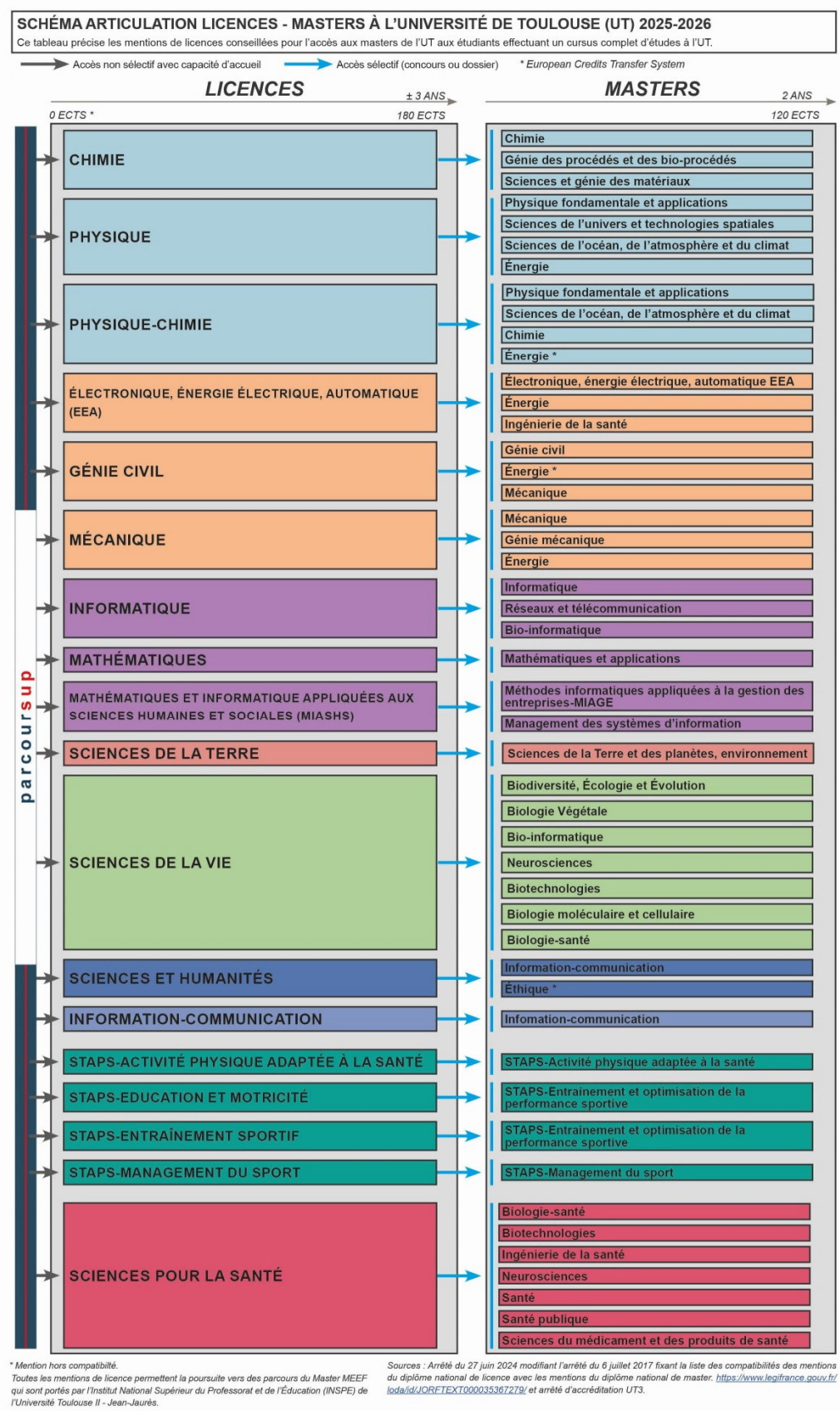
GLOSSAIRE 28

TERMES GÉNÉRAUX 28

TERMES ASSOCIÉS AUX DIPLOMES 28

TERMES ASSOCIÉS AUX ENSEIGNEMENTS 29

SCHÉMA ARTICULATION LICENCE MASTER



PRÉSENTATION DE LA MENTION

MENTION INFORMATIQUE

L'informatique est une discipline scientifique à l'impact sociétal de plus en plus important et partie intégrante de tout métier scientifique.
En première année de ce master, un socle de compétences communes conséquent sert de base à une spécialisation progressive.
En seconde année de ce master, année de spécialisation forte, une formation théorique et technologique de haut niveau est proposée aux étudiants, leur permettant d'accéder aux nombreux débouchés dans l'industrie de l'Informatique et de ses interactions mais aussi de poursuivre leurs études en doctorat.
L'offre de formation est déclinée autour des pôles thématiques suivants :

- Le traitement de l'information et ses infrastructures
- Le génie logiciel comme ensemble de concepts, de méthodes et d'outils de développement.
- La manipulation du contenu selon différents points de vue : analyse/synthèse de l'information, structuration et recherche d'information en intégrant la problématique des données massives.
- La représentation et le traitement des connaissances en intelligence artificielle, liens avec la robotique.
- L'interaction entre l'homme et la machine et les contraintes ergonomiques et cognitives y afférant.

PRÉSENTATION DE L'ANNÉE DE M1 SCIENCES DU LOGICIEL

RUBRIQUE CONTACTS

CONTACTS PARCOURS

RESPONSABLE M1 SCIENCES DU LOGICIEL

MIGEON Frédéric
Email : frederic.migeon@univ-tlse3.fr Téléphone : 05 61 55 62 46

MENGIN Jérôme
Email : mengin@irit.fr

ROCHANGE Christine
Email : christine.rochange@irit.fr Téléphone : 05 61 55 84 25

SECRÉTAIRE PÉDAGOGIQUE

DOSSANT Sabine
Email : sabine.dossant@univ-tlse3.fr Téléphone : 0561558639

CONTACTS MENTION

RESPONSABLE DE MENTION INFORMATIQUE

MENGIN Jérôme
Email : mengin@irit.fr

ROCHANGE Christine
Email : christine.rochange@irit.fr Téléphone : 05 61 55 84 25

CONTACTS DÉPARTEMENT: FSI.INFO

DIRECTEUR DU DÉPARTEMENT

GASQUET Olivier
Email : olivier.gasquet@univ-tlse3.fr

SECRETARIAT DU DÉPARTEMENT

RODRIGUES Manuella
Email : manuella.rodrigues@univ-tlse3.fr Téléphone : 05 61 55 73 54

Université Paul Sabalier
1TP1, bureau B13
118 route de Narbonne
31062 TOULOUSE cedex 9

TABLEAU SYNTHÉTIQUE DES UE DE LA FORMATION

page	Code	Intitulé UE	semestre*	ECTS	Obligatoire Facultatif	Cours	Master Class	Cours-TD	TD	TP	Projet	Stage*
Premier semestre												
	KINB7AAU	MODÉLISATION, CONCEPTION, DÉVELOPPEMENT COLLABORATIF	I	6	O							
10	KINX7AA1	Modélisation, Conception, Développement Collaboratif				22			18	14		
	KINX7AA2	Modélisation, Conception, Développement Collaboratif - mc (MCDC)					3					
	KINB7ABU	THÉORIE DES LANGAGES	I	6	O							
12	KINX7AB1	Théorie des langages (TL)				22			14	18		
	KINX7AB2	Théorie des langages - mc (TL)					3					
	KINB7ACU	ALGORITHMIQUE AVANCÉE	I	6	O							
14	KINX7AC1	Algorithmique avancée (AA)				20			24	10		
	KINX7AC2	Algorithmique avancée - mc (AA)					3					
	KINB7ADU	PARALLÉLISME	I	6	O							
15	KINX7AD1	Parallélisme (PARA)				18			20	16		
	KINX7AD2	Parallélisme - mc (PARA)					3					
	KINB7AEU	UI/UX ET APPLICATIONS FRONTALES	I	6	O							
16	KINX7AE1	UI/UX et applications frontales (UIUX)				20			10	26		
	KINX7AE2	UI/UX & AF - mc (UIUX)					4					
	KINB7FRU	MISE À NIVEAU	I	0	F			24				
Second semestre												
	KINB8AAU	GPRIA	II	9	O							
18	KINX8AA1	Travaux d'initiation à la recherche (TIR)				6						
19	KINX8AA2	Projet				8			24			
	KINX8AA3	Projet - proj									50	

* **AN** :enseignenents annuels, **I** : premier semestre, **II** : second semestre
Stage: en nombre de mois

page	Code	Intitulé UE	semestre*	ECTS	Obligatoire Facultatif	Cours	Master Class	Cours-TD	TD	TP	Projet	Stage*
	KINX8AA4	Projet					2					
20	KINX8AA5	Agilité (AGIL)				12			12	20		
21	KINX8AA6	Agilité - mc (AGIL)					2					
22	KINB8ABU	STAGE / TER	II	6	O						150	3
	KINB8ACU	MODÈLES ET ARCHITECTURE DES APPLICATIONS RÉPARTIES	II	6	O							
23	KINX8AC1	Modèles et architecture des applications réparties (MAAR)				22			14	20		
	KINX8AC2	Modèles et architecture des applications réparties (MAAR)					4					
	KINB8ADU	INGÉNIERIE DES SYS. INTERACTIFS ET DES APPLI. WEB DYNAMIQUES	II	6	O							
25	KINX8AD1	Ingénierie des systèmes interactifs et des applications web dynamiques (ISIAWD)				20			10	26		
	KINX8AD2	Ingénierie des systèmes interactifs et des applications web dynamiques - mc (ISIAWD)					2					
27	KINB8AVU	ANGLAIS	II	3	O				24			

* **AN** :enseignenents annuels, **I** : premier semestre, **II** : second semestre
Stage: en nombre de mois

LISTE DES UE

UE	MODÉLISATION, CONCEPTION, DÉVELOPPEMENT COLLABORATIF	6 ECTS	1er semestre
Sous UE	Modélisation, Conception, Développement Collaboratif		
KINX7AA1	Cours : 22h , TD : 18h , TP : 14h	Enseignement en français	Travail personnel 93 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MIGEON Frédéric
Email : frederic.migeon@univ-tlse3.fr

OBJECTIFS D'APPRENTISSAGE

Ce module vise la modélisation et la conception des logiciels complexes, avec une sensibilisation aux spécificités du développement collaboratif.Les aspects traités dans ce cours sont la modélisation, son utilisation dans le cadre d'une démarche de développement orientée objet, la spécification de contraintes afin de rendre les modèles cohérents, les patrons de conception orientés objet et leur utilisation dans le cadre d'un problème de conception, et la mise en œuvre d'un logiciel conformément aux exigences exprimées et aux bonnes pratiques relatives au développement collaboratif.

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

Partie I. Modélisation et Conception
- Gestion de la complexité des logiciels, techniques de gestion de la complexité (décomposition vs. abstraction)
- Modélisation avec UML : modélisation des exigences, modélisation structurelle
- introduction à la spécification de contraintes avec OCL
- modélisation du comportement,
- Processus / Démarche de développement : Modélisation du contexte métier, Modélisation des besoins, Conception et réalisation des besoins

Partie II. Patrons de conception
- Conception à base de patrons, qu'est-ce qu'un patron de conception, catégories de patrons
- description et classification des patrons de conception, utilisation des patrons dans la conception
- principaux patrons du ØGang of FourØ : Stratégie, Médiateur, Observateur, Décorateur, Fabrique abstraite, Fabrication...
- composition de patrons

Partie III. Développement collaboratif
- Introduction au développement collaboratif (droit d'auteur, licences libres...)
- Introduction à la gestion des exigences et des versions d'un logiciel
- Gestion de version décentralisée et modèles de développement associés

PRÉ-REQUIS

Conception et programmation orientée objet (Java), Notions d'UML (diagrammes de classes et de séquence, Notions de processus de développement

COMPÉTENCES VISÉES

- Je conçois et je mets en œuvre la modélisation d'un système à partir d'une spécification des exigences, en appliquant un processus de développement préétabli et en utilisant le langage UML
- Je conçois et je mets en œuvre ma solution au moyen du ou des patrons de conception adéquats, sélectionnés et adaptés sur la base de leur description, selon une méthodologie adaptée au développement orientée objet.
- Je contribue au développement collaboratif d'un logiciel conformément au workflow retenu par l'équipe de développement tout en maintenant son référentiel d'exigences.

RÉFÉRENCES BIBLIOGRAPHIQUES

S. Tockey. How to Engineer Software : A Model-Based Approach – J Warmer, A Kleppe. The Object Constraint Language – E. Freeman. Head First Design Patterns – P. Roques, UML2 - Modéliser une application Web – S. Chacon & B. Straub. Pro Git

MOTS-CLÉS

Conception, Modélisation, Objet, UML/OCL, Design Patterns, Développement Collaboratif

UE	THÉORIE DES LANGAGES	6 ECTS	1er semestre
Sous UE	Théorie des langages (TL)		
KINX7AB1	Cours : 22h , TD : 14h , TP : 18h	Enseignement en français	Travail personnel 96 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

CASSE Hugues
Email : Hugues.Casse@irit.fr

OBJECTIFS D'APPRENTISSAGE

Cette UE a deux objectifs : (a) comprendre le fonctionnement d'un analyseur de source et l'obtention de code intermédiaire (front-end) et (b) exploiter cette représentation pour générer et optimiser des codes exécutables (back-end) tout en mettant en oeuvre des stratégies de vérification afin d'assurer la correction du compilateur.Les compétences visées incluent :

- mettre en oeuvre un analyseur de code source,
- développer des traducteurs vérifiés appliqués à la représentation intermédiaire du programme,
- savoir optimiser les performances d'un programme traduisant des requêtes relationnelles.

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

Compilation de langage impératif

- Exécution et compilation
- Analyse syntaxique
- Génération de code

Vérification

- Preuve assistée
- Modélisation d'AST typés
- Sémantique de langages
- Transformations vérifiées

Compilation d'une requête en langage déclaratif

- Introduction et motivations
- Optimisation de code
- Génération de code

Les TPs consistent en la mise en oeuvre d'un mini-projet permettant de réaliser un compilateur composé d'un analyseur pour un langage de programmation réel exécuté par une machine virtuelle.
La première partie consiste à réaliser un mini-compilateur de l'analyse du source à la génération du code : analyse lexicale, analyse syntaxique, construction des arbres de syntaxe abstrait et génération de code. En seconde partie, des optimisations vérifiées seront réalisées sur la représentation intermédiaire . L'environnement de preuve interactive utilisé pour cela (Coq) est d'abord présenté avec un rappel des notions sous-jacentes, puis un ensemble d'exemples de sémantiques de langages de programmation est formalisé, avant de modéliser et vérifier une transformation agissant sur la représentation intermédiaire.

PRÉ-REQUIS

théorie des langages, preuve interactive, connaissances en bases de données relationnelles

COMPÉTENCES VISÉES

- mettre en oeuvre un analyseur de code source,
- développer des traducteurs vérifiés appliqués à la représentation intermédiaire du programme,
- savoir optimiser les performances d'un programme traduisant des requêtes relationnelles.

RÉFÉRENCES BIBLIOGRAPHIQUES

A. Aho et al. Compilateurs : principes, techniques et outils. Pearson Education.
Y. Bertot. Coq in a Hurry. EJCP, 2016.
M. Bouzeghoub et al. Systèmes de BD : des techniques d'implantation à la conception de schémas. Eyrolles.

MOTS-CLÉS

Compilation, optimisation, génération de code, analyse syntaxique, représentation intermédiaire, sémantiques, preuve assistée, transformation vérifiée.

UE	ALGORITHMIQUE AVANCÉE	6 ECTS	1er semestre
Sous UE	Algorithmique avancée (AA)		
KINX7AC1	Cours : 20h , TD : 24h , TP : 10h	Enseignement en français	Travail personnel 96 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MENGIN Jérôme
Email : mengin@irit.fr

OBJECTIFS D'APPRENTISSAGE

Après des rappels de complexité des algorithmes, illustrés sur des opérations sur des structures de données avancées, ce module présente les principales méthodes et algorithmes pour modéliser et résoudre les problèmes de décision et d'optimisation linéaire et / ou combinatoire. Après validation de ce module, les étudiant-e-s sauront :

- Reconnaître un problème d'optimisation combinatoire difficile
- Modéliser des problèmes en flots, en programmation linéaire (avec ou sans variable entière), en programmation par contraintes
- Choisir et utiliser efficacement un outil de résolution approprié
- Implémenter les opérations sur les arbres de recherche et les tas, et des algorithmes de résolution exacte ou incomplète pour un problème d'optimisation combinatoire, et analyser leur complexité

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

1. Structures de données avancées (tas, arbres de recherche)
2. Flots : réseaux de transport sans, calcul de flots maximum
3. Programmation linéaire : formalisme, résolution graphique, problème dual
4. Problèmes classiques d'optimisation combinatoire difficile : partitionnement, ordonnancement, routage, ...
5. Algorithmes approchés, coefficient d'approximation
6. Le problème SAT, méthode DPLL
7. Classes de complexité : La classe NP, réductions, NP-complétude et classes d'approximation
8. Formalismes génériques et recherches exhaustives : PPC, backtrack, heuristiques, forward checking ; PLNE, relaxation ; modélisation
9. Méthodes incomplètes : principe des recherches locales / sur populations ; méta-heuristiques (liste tabou, ...)

Pour chaque type de problème d'optimisation ou de décision abordé, on étudie la complexité des algorithmes et la modélisation. Les travaux pratique consisteront en l'implémentation d'un solveur de type "backtrack", en l'implémentation d'algorithmes de recherche locale, et en l'utilisation de solveurs de PPC / PLNE.

PRÉ-REQUIS

Graphes, Structures de Données, Complexité

RÉFÉRENCES BIBLIOGRAPHIQUES

Algorithmique. T. Cormen, C. Leiserson, R. and C. Stein. 3ème éd., 2010
Algorithms. S. Dasgupta, C.H. Papadimitriou, U.V. Vazirani. McGraw-Hill, 2006.

MOTS-CLÉS

Flots, Programmation Linéaire, optimisation combinatoire, SAT, CSP, classes de complexité, Méta-Heuristiques, Tas binomiaux, B-arbres

UE	PARALLÉLISME	6 ECTS	1er semestre
Sous UE	Parallélisme (PARA)		
KINX7AD1	Cours : 18h , TD : 20h , TP : 16h	Enseignement en français	Travail personnel 96 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

DA COSTA Georges
Email : dacosta@irit.fr

OBJECTIFS D'APPRENTISSAGE

L'objectif de ce module d'enseignement est d'introduire les fondements du parallélisme sur des architectures distribuées ou massivements parallèles (GPU) :

- les modèles du parallélisme (synchrone/asynchrone)
- l'abstraction d'algorithmes parallèles (Réseaux de Petri)
- les concepts de coopération et de synchronisation
- les modèles permettant d'atteindre un parallélisme efficace (MPI, Cuda)

A l'issue de cet enseignement, l'étudiant sera capable de concevoir, analyser et évaluer des algorithmes parallèles. L'étude des approche et API MPI et Cuda permettra à l'étudiant de résoudre des problèmes à l'aide d'algorithmes parallèles efficaces. La présentation des "conditions" complètera la formation de l'étudiant quant à la synchronisation d'activités parallèles.

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

La pédagogie se basera sur plusieurs mises en situation (TD et TP) pour intégrer les particularités du parallélisme.

Parallélisme/synchronisation

- moniteurs de Hoare, conditions
- réseaux de Petri

Modèle MPI

- modèle et primitives
- parallélisation et implémentation d'algorithmes

Modèle Cuda

- modèle de programmation et API
- modèle d'exécution, mémoires et optimisation
- applications, bibliothèques et outils

Master Class

PRÉ-REQUIS

Algorithmique avancée, programmation concurrente, processus, threads, variables partagées, architecture des ordinateurs, réseau, abstraction de problèmes

RÉFÉRENCES BIBLIOGRAPHIQUES

Fundamentals of Parallel Multicore Architecture, Chapman and Hall/CRC, Y. Solihin
Principles of Concurrent and Distributed Programming, Addison-Wesley.
Communication and Concurrency, Prentice Hall Int. Series in Computer Science, R. Milner.

MOTS-CLÉS

Architectures parallèles, Modèles parallèles, Modèles répartis, cohérence de données, expressions et conditions de synchronisation, MPI, CUDA

UE	UI/UX ET APPLICATIONS FRONTALES	6 ECTS	1er semestre
Sous UE	UI/UX et applications frontales (UIUX)		
KINX7AE1	Cours : 20h , TD : 10h , TP : 26h	Enseignement en français	Travail personnel 90 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MARTINIE Celia
Email : Celia.Martinie@irit.fr

OBJECTIFS D'APPRENTISSAGE

La partie du module dédiée à l'UI/UX a pour objectif de permettre aux étudiants de maîtriser les principes de conception, programmation et évaluation d'interfaces utilisateur et techniques d'interaction avec pour objectif la production de systèmes informatiques utilisables.
La partie du module dédiée aux applications frontales vise à étudier les langages côté client (front end) et à savoir mettre en œuvre un framework dans le cadre d'une application web dynamique de type "Single Page Application".

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

Partie "UI/UX" :

1. Conception centrée utilisateur (UCD) selon le standard ISO 9241-210, règles et standards Introduction à l'analyse des utilisateurs (personas, modélisation des tâches)
2. Etude et mise en oeuvre de techniques de conception : prototypage basse, moyenne et haute-fidélité
3. Etude et mise en oeuvre de techniques d'évaluation analytiques (cheminement centré tâches, évaluation heuristique, prédictions à base de modèles) et empiriques (tests d'utilisabilité)
4. Retour sur investissement de l'UCD
5. UX et ses différentes dimensions

Partie "Front" :

1. Rudiments d'un langage côté client (Javascript ou typescript) : syntaxe du langage, utilisation dans le navigateur, les principales structures de données, etc
2. Spécificités d'un langage côté client (Javascript ou typescript) : le modèle objet du langage, la puissance des fonctions, ...
3. Etude d'un framework de type front end (Angular, React ou Vue.js)
4. Manipulation du DOM et notion de DOM virtuel
5. Gestion des événements et des formulaires
6. Routage et application "Single Page Application"

Techno utilisées en TP : outils de prototypage et de modélisation des tâches, Javascript ou typescript et framework de type front end, node.js

PRÉ-REQUIS

Programmation orientée objet et programmation fonctionnelle, modélisation UML, éléments du langage HTML et CSS

COMPÉTENCES VISÉES

— Proposer et mettre en oeuvre une démarche de conception centrée utilisateur et un ensemble de techniques associées pour le développement d'applications informatiques.
— Concevoir et implémenter la partie représentation d'une application web réactive en m'appuyant sur un framework dédié.

RÉFÉRENCES BIBLIOGRAPHIQUES

Standard ECMA-262
JavaScript and JQuery : Interactive Front-End Web Development. Jon Duckett
ISO 9241-210 - Part 210, 2018

MOTS-CLÉS

Programmation front end, framework, DOM, single page application
UI, Conception centrée utilisateur, prototypage, modèles de tâches, UX

UE	GPRIA	9 ECTS	2nd semestre
Sous UE	Travaux d'initiation à la recherche (TIR)		
KINX8AA1	Cours : 6h	Enseignement en français	Travail personnel 139 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MORENO José
Email : jose.moreno@univ-tlse3.fr

UE	GPRIA	9 ECTS	2nd semestre
Sous UE	Projet		
KINX8AA2	Cours : 8h , TD : 24h	Enseignement en français	Travail personnel 139 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MIGEON Frédéric
Email : frederic.migeon@univ-tlse3.fr

UE	GPRIA	9 ECTS	2nd semestre
Sous UE	Agilité (AGIL)		
KINX8AA5	Cours : 12h , TD : 12h , TP : 20h	Enseignement en français	Travail personnel 139 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MIGEON Frédéric
Email : frederic.migeon@univ-tlse3.fr

UE	GPRIA	9 ECTS	2nd semestre
Sous UE	Agilité - mc (AGIL)		
KINX8AA6	Master Class : 2h	Enseignement en français	Travail personnel 139 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MIGEON Frédéric
Email : frederic.migeon@univ-tlse3.fr

UE	STAGE / TER	6 ECTS	2nd semestre
KINB8ABU	Stage : 3 mois minimum , Projet : 150h	Enseignement en français	Travail personnel 150 h

[Retour liste des UE]

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

L'étudiant.e peut faire un stage d'au moins 3 mois en entreprise ou en laboratoire, ou travailler à l'université pour une durée équivalente sur l'un des sujets proposés par l'équipe pédagogique du master.

UE	MODÈLES ET ARCHITECTURE DES APPLICATIONS RÉPARTIES	6 ECTS	2nd semestre
Sous UE	Modèles et architecture des applications réparties (MAAR)		
KINX8AC1	Cours : 22h , TD : 14h , TP : 20h	Enseignement en français	Travail personnel 90 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

MARTIN-DOREL Érik
Email : erik.martin-dorel@ens-lyon.org

OBJECTIFS D'APPRENTISSAGE

Ce cours présente les méthodes et outils permettant de concevoir et de gérer les applications informatiques distribuées et hétérogènes.La première partie du cours est consacrée à la représentation de données structurées et l'interopérabilité, avec un focus particulier sur les langages XML (modélisation via différents types de grammaires, langages de transformation et d'interrogation), et sur l'approche orientée services (aspects architecturaux, conception, développement et publication).Dans la deuxième partie, on poursuivra l'étude des Design Patterns du GoF initiée en S7 au tronc commun. Cette partie présentera aussi des patterns de programmation concurrente et répartie. En outre, on étudiera les bases de la programmation concurrente en Java et de la programmation répartie avec Java RMI.

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

Partie I : Interopérabilité des données et approches orientées services

1. Gestion de build et des dépendances dans un projet Java : Apache Maven
2. Représentations de données structurées :
   — Langages XML : Modèles de données XML (DTD, schémas XML) ; Transformations et extractions d'informations (XSLT, XPath) ; Parsing de documents XML (DOM, SAX, StAX, JAXB)
   — Comparaison avec JSON (JSON Schema, JSONP, JSONB)
3. Approches orientée services :
   — Web Services SOAP : WSDL, SOAP, UDDI ; Méthodologies de conception contract-first et code-first
   — Web Services REST

Partie II : Design patterns (compléments), Programmation concurrente et répartie en Java

1. Compléments sur les design patterns du GoF
2. Programmation concurrente en Java et multithreading :
   — Multithreading en Java (classe Thread, interface Runnable, synchronisation de threads, attentes et notifications) ; Design patterns de programmation concurrente (Objet actif)
3. Programmation répartie en Java :
   — Introduction au middleware, principes des middleware à objets, et Design patterns de programmation répartie
   — Java RMI et Conception/programmation à base d'objets répartis

PRÉ-REQUIS

Programmation orientée objet (notamment Java), connaissances élémentaires en bases de données et en programmation répartie (modèle client-serveur)

RÉFÉRENCES BIBLIOGRAPHIQUES

— N. Bradley : The XML companion. Addison-Wesley, 3rd edition, 2001
— E. Freeman, E. Freeman, K. Sierra, B. Bates : Head First Design Patterns. O'Reilly, 2006
— C. Delannoy : Programmer en Java, 9e édition. Eyrolles, 2014

MOTS-CLÉS
Interopérabilité, données structurées, services web, design patterns, concurrence, répartition, middleware, Java.

UE	INGÉNIERIE DES SYS. INTERACTIFS ET DES APPLI. WEB DYNAMIQUES	6 ECTS	2nd semestre
Sous UE	Ingénierie des systèmes interactifs et des applications web dynamiques (ISIAWD)		
KINX8AD1	Cours : 20h , TD : 10h , TP : 26h	Enseignement en français	Travail personnel 92 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

PALANQUE Philippe
Email : Philippe.Palanque@irit.fr

OBJECTIFS D'APPRENTISSAGE

La partie "Ingénierie des Applications Web dynamiques" se positionne côté "back office" et est dédiée à des aspects de conception (archi. multi-couches, rôle d'un "framework") et technologiques (J2EE : servlet, JSP, ORM, etc).Compétences :

- Mettre en œuvre le patron MVC dans une architecture multi-couches
- Faire cohabiter une représentation relationnel d'un SGBD avec une représentation objet

La partie "Ingénierie des Systèmes Interactifs" (ISI) vise la maîtrise de la modélisation et de la programmation des systèmes interactifs (SI). Compétences :

- Architecturer et modéliser les SI pour garantir utilisabilité, modifiabilité et fiabilité
- Exploiter le design pattern MVC
- Valider la fiabilité d'un SI (test et vérification)
- Mettre en œuvre dans un environnement de programmation par événement

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

Partie "Applications web dynamiques" :

1. Eléments d'architecture répartie : client-serveur vs. n-tier, MVC1 et MVC2
2. Le rôle d'un framework : injection de dépendances, inversion de contrôle
3. La plateforme J2EE : cycle de vie d'une servlet et d'une JSP, syntaxe JSP, portée et communication par objets implicites, cohabitation d'un modèle objet et relationnel d'un SGBD : JDBC, JPA

Techno utilisées en TP : Spring Boot ; Hibernate ; Thymeleaf

Partie "Systèmes interactifs" :

1. Principes architecturaux des systèmes interactifs (modèles de Seeheim et ARCH)
2. Design pattern MVC, de sa mise en oeuvre en Java et de son intégration dans ARCH
3. Principes de modélisation des systèmes interactifs à base d'automates à états finis étendus.
4. Implémentation à base de modèle dans un environnement de programmation par événement
5. Description de propriétés de systèmes interactifs et vérification sur modèles. Comment gérer utilisabilité, modifiabilité et fiabilité dans un même cadre méthodologique
6. Mise en œuvre des principes de validation : expression et vérification de propriétés, définition et mise en œuvre de tests sur des systèmes interactif

Techno utilisées en TP : NetBeans, Java Swing

PRÉ-REQUIS

Programmation orientée objet (Java), UML, HTML, modèle relationnel pour les données, Java SWING et Programmation par événement, modélisation par automates

RÉFÉRENCES BIBLIOGRAPHIQUES

Conception d'applications en Java/JEE. Jacques Lonchamp. Dunod.
L. Bass, P. Clements, R. Kazman, Software Architecture in Practice, (3rd edition), Addison-Wesley, 2012.

MOTS-CLÉS

Application répartie, persistance, J2EE,
Modélisation de systèmes interactifs, architecture logicielle, fiabilité, modifiabilité, utilisabilité, vérification

UE	ANGLAIS	3 ECTS	2nd semestre
KINB8AVU	TD : 24h	Enseignement en français	Travail personnel 51 h

[Retour liste des UE]

ENSEIGNANT(E) RESPONSABLE

CHAPLIER Claire
Email : claire.chaplier@univ-tlse3.fr

OBJECTIFS D'APPRENTISSAGE

Niveau C1 du CECRL (Cadre Européen Commun de Référence pour les Langues)Permettre aux étudiants de développer les compétences indispensables à la réussite dans leur future vie professionnelle en contextes culturels variés.Acquérir l'autonomie linguistique nécessaire et perfectionner les outils de langue spécialisée permettant l'intégration professionnelle et la communication d'une expertise scientifique dans le contexte international.

DESCRIPTION SYNTHÉTIQUE DES ENSEIGNEMENTS

Développer :
- les compétences liées à la compréhension de publications scientifiques ou professionnelles rédigées en anglais ainsi que les compétences nécessaires à la compréhension de communications scientifiques orales.
- les outils d'expression permettant de maîtriser une présentation orale et/ou écrite et d'aborder une discussion critique dans le domaine scientifique
- la maîtrise des éléments d'argumentation critique à l'oral et/ou à l'écrit d'une publication scientifique
une réflexion plus large sur leur place, leur intégration et leur rayonnement en tant que scientifiques dans la société, abordant des questions d'actualité, d'éthique, d'intégrité...

PRÉ-REQUIS

- Niveau B2 du CECR

COMPÉTENCES VISÉES

S'exprimer avec aisance à l'oral, devant un public, en usant de registres adaptés aux différents contextes et aux différents interlocuteurs.Se servir aisément d'une langue vivante autre que le français : compréhension et expression écrites et orales :
- Comprendre un article scientifique ou professionnel rédigé en anglais sur un sujet relatif à leur domaine.- Interagir à l'oral en anglais : réussir ses échanges formels et informels lors des colloques, réunions ou entretiens professionnels.- De&#769 ;fendre sa candidature par e&#769 ;crit (CV) ou a&#768 ; l'oral (entretien de recrutement) en anglais

MOTS-CLÉS

Projet Anglais scientifique Rédaction Publication Communication esprit critique scientifique interculturel

GLOSSAIRE

TERMES GÉNÉRAUX

SYLLABUS

Dans l'enseignement supérieur, un syllabus est la présentation générale d'un cours ou d'une formation. Il inclut : objectifs, programme de formation, description des UE, prérequis, modalités d'évaluation, informations pratiques, etc.

DÉPARTEMENT

Les départements d'enseignement sont des structures d'animation pédagogique internes aux composantes (ou facultés) qui regroupent les enseignantes et enseignants intervenant dans une ou plusieurs mentions.

UE : UNITÉ D'ENSEIGNEMENT

Un semestre est découpé en unités d'enseignement qui peuvent être obligatoires, à choix ou facultatives. Une UE représente un ensemble cohérent d'enseignements auquel sont associés des ECTS.

UE OBLIGATOIRE / UE FACULTATIVE

L'UE obligatoire fait référence à un enseignement qui doit être validé dans le cadre du contrat pédagogique. L'UE facultative vient en supplément des 60 ECTS de l'année. Elle est valorisée dans le supplément au diplôme. L'accumulation de crédits affectés à des UE facultatives ne contribue pas à la validation de semestres ni à la délivrance d'un diplôme.

ECTS : EUROPEAN CREDITS TRANSFER SYSTEM

Les ECTS constituent l'unité de mesure commune des formations universitaires de licence et de master dans l'espace européen. Chaque UE obtenue est ainsi affectée d'un certain nombre d'ECTS (en général 30 par semestre d'enseignement, 60 par an). Le nombre d'ECTS varie en fonction de la charge globale de travail (CM, TD, TP, etc.) y compris le travail personnel. Le système des ECTS vise à faciliter la mobilité et la reconnaissance des diplômes en Europe.

TERMES ASSOCIÉS AUX DIPLOMES

Les diplômes sont déclinés en domaines, mentions et parcours.

DOMAINE

Le domaine correspond à un ensemble de formations relevant d'un champ disciplinaire ou professionnel commun. La plupart des formations de l'UT3 relèvent du domaine « Sciences, Technologies, Santé ».

MENTION

La mention correspond à un champ disciplinaire. Il s'agit du niveau principal de référence pour la définition des diplômes nationaux. La mention comprend, en général, plusieurs parcours.

PARCOURS

Le parcours constitue une spécialisation particulière d'un champ disciplinaire choisie par l'étudiant·e au cours de son cursus.

LICENCE CLASSIQUE

La licence classique est structurée en six semestres et permet de valider 180 crédits ECTS. Les UE peuvent être obligatoires, à choix ou facultatives. Le nombre d'ECTS d'une UE est fixé sur la base de 30 ECTS pour l'ensemble des UE obligatoires et à choix d'un semestre.

LICENCE FLEXIBLE

À la rentrée 2022, l'université Toulouse III - Paul Sabatier met en place une licence flexible. Le principe est d'offrir une progression "à la carte" grâce au choix d'unités d'enseignement (UE). Il s'agit donc d'un parcours de formation personnalisable et flexible dans la durée. La progression de l'étudiant·e dépend de son niveau de départ et de son rythme personnel. L'inscription à une UE ne peut être faite qu'à condition d'avoir validé les UE pré-requises. Le choix de l'itinéraire de la licence flexible se fait en concertation étroite avec une direction des études (DE) et dépend de la formation antérieure, des orientations scientifiques et du projet professionnel de l'étudiant·e. L'obtention du diplôme est soumise à la validation de 180 crédits ECTS.

DIRECTION DES ÉTUDES ET ENSEIGNANT·E RÉFÉRENT·E

La direction des études (DE) est constituée d'enseignantes et d'enseignants référents, d'une directrice ou d'un directeur des études et d'un secrétariat pédagogique. Elle organise le projet de formation de l'étudiant·e en proposant une individualisation de son parcours pouvant conduire à des aménagements. Elle est le lien entre l'étudiant·e, l'équipe pédagogique et l'administration.

TERMES ASSOCIÉS AUX ENSEIGNEMENTS

CM : COURS MAGISTRAL(AUX)

Cours dispensé en général devant un grand nombre d'étudiantes et d'étudiants (par exemple, une promotion entière), dans de grandes salles ou des amphithéâtres. Ce qui caractérise également le cours magistral est qu'il est le fait d'une enseignante ou d'un enseignant qui en définit les structures et les modalités. Même si ses contenus font l'objet de concertations avec l'équipe pédagogique, chaque cours magistral porte donc la marque de la personne qui le crée et le dispense.

TD : TRAVAUX DIRIGÉS

Ce sont des séances de travail en groupes restreints (de 25 à 40 étudiantes et étudiants selon les composantes), animées par des enseignantes et enseignants. Les TD illustrent les cours magistraux et permettent d'approfondir les éléments apportés par ces derniers.

TP : TRAVAUX PRATIQUES

Méthode d'enseignement permettant de mettre en pratique les connaissances théoriques acquises durant les CM et les TD. Généralement, cette mise en pratique se réalise au travers d'expérimentations et les groupes de TP sont constitués de 16 à 20 étudiantes et étudiants. Certains travaux pratiques peuvent être partiellement encadrés ou peuvent ne pas être encadrés du tout. A contrario, certains TP, du fait de leur dangerosité, sont très encadrés (jusqu'à une enseignante ou un enseignant pour quatre étudiantes et étudiants).

PROJET OU BUREAU D'ÉTUDE

Le projet est une mise en pratique en autonomie ou en semi-autonomie des connaissances acquises. Il permet de vérifier l'acquisition de compétences.

TERRAIN

Le terrain est une mise en pratique encadrée des connaissances acquises en dehors de l'université.

STAGE

Le stage est une mise en pratique encadrée des connaissances acquises dans une entreprise ou un laboratoire de recherche. Il fait l'objet d'une législation très précise impliquant, en particulier, la nécessité d'une convention pour chaque stagiaire entre la structure d'accueil et l'université.

SESSIONS D'ÉVALUATION

Il existe deux sessions d'évaluation : la session initiale et la seconde session (anciennement appelée "session de rattrapage", constituant une seconde chance). La session initiale peut être constituée d'examens partiels et terminaux ou de l'ensemble des épreuves de contrôle continu et d'un examen terminal. Les modalités de la seconde session peuvent être légèrement différentes selon les formations.

SILLON

Un sillon est un bloc de trois créneaux de deux heures d'enseignement. Chaque UE est généralement affectée à un sillon. Sauf cas particuliers, les UE positionnées dans un même sillon ont donc des emplois du temps incompatibles.


Université
de Toulouse