Conception Logiciel ***Software Design***			
Code cours *Course code*: **COL**		**Crédits ECTS** *ECTS Credits:* **2,5**	
Département *Department*	: IA	**Cours** *Lectures*	: 17H30
Coordonnateurs *Lecturers*	: M. RICHARD	**T.D.** *Tutorials*	: 17H30
		T.P. *Laboratory sessions*	:
Période *Year of study*	: 3ième année *3rd year*	**Projet** *Project*	:
		Non encadré *Homework*	:
Semestre *Semester*	: 5ième semester – A *5th semester - A*	**Horaire global** *Total hours*	: 35H
Evaluation *Assessment method(s)*	: 1 examen écrit, *1 written exam*		
Langue d'instruction *Language of instruction*	: Français *French*		
Type de cours *Type of course*	: Obligatoire *Compulsory*		
Niveau *Level of course*	: Avancé *graduate*		

Compétences attendues :

À l'issue de ce module, l'étudiant maitrisera :

- L'utilisation d'un langage de modélisation UML et du langage de programmation objet Java
- les concepts avancés de la programmation orientée objet (Polymorphisme, Généricité, Patrons de conception)
- les différentes étapes (conception, implémentation, tests unitaires et d'intégration) d'un cycle de conception logiciel (cycle en V, en spirale, …)
- la conception et l'implémentation des couches DAL, Business et Service d'une architecture logicielle

Pré-requis : Une pratique d'un langage de programmation procédural.

Contenu :

Ce module est structuré en deux parties :

1. Conception et Implémentation d'un logiciel selon le paradigme objet :

Cette partie est consacrée à l'apprentissage de la conception orientée objet d'un logiciel à partir d'un cahier des charges. Nous verrons alors les différents concepts associés à chacune des étapes d'un cycle de développement : Spécification conception, implémentation, tests unitaires et tests d'intégration.

Afin de satisfaire les critères de qualité logiciel une partie de ce chapitre sera consacrée à l'étude et la mise en œuvre des principaux partons de conception (Pattern Design)

2. Architectures logicielles :

Ce chapitre présente les différentes architectures logicielles et notamment l'architecture en couche, principale architecture utilisée dans le développement logicielle. Plus précisément les couches DAL (Data Access Layer) et Métier (Business) et les différentes techniques de liaison seront étudiées.

La dernière partie de ce chapitre s'intéresse aux architectures micro-services, venant s'appuyer sur l'architecture précédente.

Toutes ces notions seront largement mises en œuvre au cours des différentes séances de travaux dirigés.
Le langage de programmation objet utilisé dans le cadre de ce module est le Java.

Bibliographie :

Expected competencies:

At the end of this module, the student will master :
- The use of the UML modeling language and the Java object programming language
- Advanced concepts of object-oriented programming (Polymorphism, Genericity, Design patterns)
- the different steps (design, implementation, unit and integration tests) of a software design cycle (V-cycle, spiral, ...)
- the design and implementation of the DAL, Business and Service layers of a software architecture

Prerequisites: A practice of a procedural programming language.

Content:

This module is structured in two parts:

1. Design and implementation of a software according to the object paradigm:

This part is dedicated to the learning of the object-oriented design of a software from a specification. We will then see the different concepts associated with each step of the development cycle: specification, design, implementation, unit testing and integration testing.

In order to satisfy the criteria of software quality, a part of this chapter will be devoted to the study and the implementation of the principal partons of design (Pattern Design)

2. Software architectures :

This chapter presents the different software architectures and in particular the layer architecture, the main architecture used in software development. More precisely, the DAL (Data Access Layer) and Business layers and the different linking techniques will be studied.

The last part of this chapter focuses on micro-services architectures, which are based on the previous architecture.

Recommended reading: