

# NSY208 - Architecture, Patterns, et Intégration : systèmes embarqués et mobiles en Java et Android (1)

## Présentation

### Prérequis

Auditeurs se destinant aux métiers de l'intégration des systèmes et désireux de maîtriser en particulier les technologies de la mobilité. Des connaissances préalables en génie logiciel, en conception et en programmation par objets sont requises pour suivre ce cours, avec connaissance d'au moins un langage à objet (Java, C#, ..).

### Objectifs pédagogiques

Savoir concevoir et développer des systèmes intégrant des éléments embarqués et communicants, en utilisant les produits industriels ou grand public (téléphones mobiles, tablettes, ...) et les technologies logicielles actuelles (Java, C#, .NET, Android,...)

### Compétences

Maîtriser l'architecture d'un système autonome ou basé sur un intergiciel (middleware). Savoir choisir les patrons de conception (design patterns) adaptés. Etre capable de réaliser la conception, l'intégration, le déploiement, et la mise en oeuvre de tels systèmes.

## Programme

### Contenu

**Programmation Java avancée** : concurrente, distributive et dynamique.

**La conception orientée domaine** (*Domain Driven Design*)

**Design Patterns**: notion de motif (description, utilisation). Intégration des motifs de conception dans les systèmes embarqués. *Patterns MVC*, *Inversion de Contrôle*, *Interceptor*, ...

**Programmation en langage Python**

Modélisation des systèmes: **le langage SysML** (*System Modeling Language*).

**Persistance et mapping objet-relational (ORM)**: principes de base, programmation de la persistance. L'interface **JPA** (*Java Persistence API*), **Hibernate**, patterns pour la persistance, **DAO**, ...

**Architecture REST**: principes et mise en oeuvre des services REST

**Programmer avec le cloud** : principes, SaaS, PaaS, IaaS. Concepts et programmation avec **Amazon AWS** (EC2, S3, EFS, ...).

**Bases de données NoSQL, Big Data**: motivations, typologie des approches, paradigmes clé-valeur, bases orientées document, orientées colonnes, ... Panorama des logiciels existants.

**La programmation en Javascript et le langage HTML5**

**Les architectures à micro-services. La conteneurisation avec Docker. L'approche DevOps**

**Etude de cas.**

### Modalités de validation

- Examen final

### Description des modalités de validation

Mis à jour le 01-09-2019



**Code : NSY208**

Unité d'enseignement de type cours

6 crédits

Volume horaire de référence (+/- 10%) : **50 heures**

**Responsabilité nationale :**

EPN05 - Informatique / 1

**Contact national :**

EPN05 - Informatique

2 rue Conté

33.1.13A

75003 Paris

01 40 27 26 81

Safia Sider

[safia.sider@lecnam.net](mailto:safia.sider@lecnam.net)

## Bibliographie

---

Titre	Auteur(s)
Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects.	Douglas C Schmidt, Michael Stal, Hans Rohnert, Frank Busc
Pattern Oriented Software Architecture Volume 5: On Patterns and Pattern Languages	Frank Buschmann, Kevlin Henney , Douglas C. Schmidt
Architectures, systèmes, intégration (Ellipses, référence science)	Yann Pollet
Architectures logicielles réparties. Du client-serveur au cloud (Ellipses, référence science)	Yann Pollet