Contenus

Nom du Cours		Semestre du Cours	Cours Théoriques	Travaux Dirigés (TD)	Travaux Pratiques (TP)	Crédit du Cours	ECTS
INF243-B	Programmation Orientée Objet	4	2	0	2	3	5

Cours Pré-Requis	INF114
Conditions d'Admission au Cours	INF114

Langue du Cours	
Type de Cours	Obligatoire
Niveau du Cours	Licence
Objectif du Cours	L'objectif principal de ce cours est de permettre à l'étudiant de comprendre en profondeur le paradigme et les principes de la Programmation Orientée Objet (POO), qui est au cœur des processus modernes de développement logiciel. Tout au long du cours, les étudiants viseront à acquérir les compétences suivantes : - Analyser des problèmes selon une perspective orientée objet. - Décomposer des systèmes logiciels complexes en éléments gérables en utilisant les principes d'abstraction et de modularité. - Développer des codes réutilisables, flexibles et durables en utilisant les structures de Classe (Class) et d'Objet (Object). - Maîtriser les piliers fondamentaux tels que l'encapsulation (data encapsulation), l'héritage (inheritance) et le polymorphisme (polymorphism), tout en acquérant la capacité de modéliser l'architecture d'un système via les diagrammes UML lors de la phase de conception.
Contenus	- Fondamentaux du Paradigme Orienté Objet : Approche orientée objet dans le développement logiciel, concepts de classe et d'objet. - Abstraction et Encapsulation : Principes de masquage des données (data hiding), modificateurs d'accès et conception de structures modulaires. - Relations entre Classes et Modélisation : Analyse des relations entre objets (is-a, has-a) et modélisation de systèmes avec les diagrammes de classes UML. - Héritage et Réutilisation du Code : Mise en place de structures hiérarchiques, redéfinition de méthodes (overriding) et architecture logicielle extensible. - Polymorphisme et Conception Flexible : Liaison dynamique (dynamic binding), interfaces et classes abstraites, développement de systèmes à faible couplage (loosely coupled). - Gestion des Erreurs et Structures de Données : Contrôle des exceptions (exceptions) et gestion dynamique des données. - Opérations d'Entrée/Sortie (I/O) et Persistance : Interaction avec les systèmes de fichiers et techniques de sérialisation d'objets.
Ressources	- Y. Daniel Liang, “Introduction to Java Programming”, Pearson, International Edition, Comprehensive 9th/10th /11th Edition - Y. Daniel Liang, "Introduction to Java Programming and Data Structures", Pearson, 13E - Sarnath Ramnath, Brahma Dathan, “Object-Oriented Analysis and Design“, Springer

Intitulés des Sujets Théoriques

Semaine	Intitulés des Sujets
1	Introduction et Paradigme POO: Programmation Procédurale vs. Orientée Objet, Concepts de Base

Semaine	Intitulés des Sujets
2	Fondamentaux Java et Mémoire: JVM, JRE, Variables, Types de Données (Primitifs vs. Référence), Logique du Stack et du Heap
3	Structure des Classes et Objets: Constructeurs, Surcharge de Méthodes (Method Overloading)
4	Encapsulation des Données: Modificateurs d'Accès (public, private, protected), Accesseurs/Mutateurs (Getters/Setters), mot-clé this, Portée (Scope)
5	Relations entre Classes et UML: Association, Agrégation, Composition et Diagrammes de Classes
6	Relations Avancées: Association, Agrégation, Composition et Multiplicité
7	Héritage (Inheritance): Utilisation de extends, mot-clé super, Redéfinition de Méthodes (Method Overriding)
8	Semaine d'Examen Partiel: Pas de cours - Examen Intra-semestriel (Midterm)
9	Classes Abstraites et Interfaces: Classes Abstraites vs. Interfaces, Problème de l'Héritage Multiple
10	Polymorphisme: Liaison Dynamique (Dynamic Binding), Upcasting et Downcasting / Annonce du Projet de Session
11	Gestion des Exceptions: Blocs Try-Catch, Exceptions Personnalisées, Hiérarchie des Exceptions
12	Fichiers et Flux d'E/S (I/O): Lecture/Écriture de fichiers, Sérialisation (Serialization)
13	Programmation Générique: Implémentation avec des exemples de Structures de Données
14	Révision et Études de Cas: Révision complète des conceptions POO avec des exemples concrets et modernes