

## Content

Course Code	Course Name	Semester	Theory	Practice	Lab	Credit	ECTS
INF243-A	Object Oriented Programming	4	2	0	2	3	5
Prerequisites	INF114						
Admission Requirements	INF114						
Language of Instruction	French						
Course Type	Compulsory						
Course Level	Bachelor Degree						
	The primary objective of this course is to provide students with a deep understanding of the Object-Oriented Programming (OOP) paradigm and principles, which are at the heart of modern software development processes. Throughout the course, students aim to:						
Objective	<ul style="list-style-type: none"><li>- Analyze problems from an object-oriented perspective.</li><li>- Decompose complex software systems into manageable components using abstraction and modularity principles.</li><li>- Develop reusable, flexible, and sustainable code using Class and Object structures.</li><li>- Gain proficiency in modeling system architecture via UML diagrams during the design phase, alongside mastering fundamental pillars such as encapsulation, inheritance, and polymorphism.</li><li>- Foundations of the Object-Oriented Paradigm: Object-oriented approach in software development, concepts of classes and objects.</li><li>- Abstraction and Encapsulation: Data hiding principles, access modifiers, and modular structure design.</li><li>- Class Relationships and Modeling: Analysis of inter-object relationships (is-a, has-a) and system modeling with UML class diagrams.</li></ul>						
Content	<ul style="list-style-type: none"><li>- Inheritance and Code Reusability: Establishing hierarchical structures, method overriding, and extensible software architecture.</li><li>- Polymorphism and Flexible Design: Dynamic binding, interfaces, abstract classes, and developing loosely coupled systems.</li><li>- Error Handling and Data Structures: Management of exceptions and dynamic data management.</li><li>- I/O Operations and Persistence: Interaction with file systems and object serialization techniques.</li></ul>						
References	<ul style="list-style-type: none"><li>- Y. Daniel Liang, "Introduction to Java Programming", Pearson, International Edition, Comprehensive 9th/10th /11th Edition</li><li>- Y. Daniel Liang, "Introduction to Java Programming and Data Structures", Pearson, 13E</li><li>- Sarnath Ramnath, Brahma Dathan, "Object-Oriented Analysis and Design", Springer</li></ul>						

## Theory Topics

Week	Weekly Contents
1	Introduction & OOP Paradigm: Procedural vs. Object-Oriented Programming, Core Concepts
2	Java Fundamentals & Memory: JVM, JRE, Variables, Data Types (Primitive vs. Reference), Stack and Heap Logic
3	Class & Object Structure: Constructors, Method Overloading
4	Data Encapsulation: Access Modifiers (public, private, protected), Getter/Setter Methods, this keyword, Scope
5	Class Relationships & UML: Association, Aggregation, Composition, and Class Diagrams
6	Advanced Relationships: Association, Aggregation, Composition, and Multiplicity
7	Inheritance: Use of extends, super keyword, Method Overriding
8	Midterm Week: No classes - Midterm Exam
9	Abstract Classes & Interfaces: Abstract Classes vs. Interfaces, The Diamond Problem (Multiple Inheritance)
10	Polymorphism: Dynamic Binding, Upcasting and Downcasting / Announcement of Term Project
11	Exception Handling: Try-Catch blocks, Custom Exceptions, Exception Hierarchy
12	File Operations & I/O: StreamsReading/Writing Files, Serialization
13	Generic Programming: Implementation with Data Structure examples
14	Review & Case Studies: Comprehensive review of OOP designs with modern, real-world examples.