Content

Course Code	Course Name	Semester	Theory	Practice	Lab	Credit	ECTS
INF243-B	Object Oriented Programming	4	2	0	2	3	5

Prerequisites	INF114
Admission Requirements	INF114

Language of Instruction	
Course Type	Compulsory
Course Level	Bachelor Degree
Objective	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: - Analyze problems from an object-oriented perspective. - Decompose complex software systems into manageable components using abstraction and modularity principles. - Develop reusable, flexible, and sustainable code using Class and Object structures. - Gain proficiency in modeling system architecture via UML diagrams during the design phase, alongside mastering fundamental pillars such as encapsulation, inheritance, and polymorphism.
Content	- Foundations of the Object-Oriented Paradigm: Object-oriented approach in software development, concepts of classes and objects. - Abstraction and Encapsulation: Data hiding principles, access modifiers, and modular structure design. - Class Relationships and Modeling: Analysis of inter-object relationships (is-a, has-a) and system modeling with UML class diagrams. - Inheritance and Code Reusability: Establishing hierarchical structures, method overriding, and extensible software architecture. - Polymorphism and Flexible Design: Dynamic binding, interfaces, abstract classes, and developing loosely coupled systems. - Error Handling and Data Structures: Management of exceptions and dynamic data management. - I/O Operations and Persistence: Interaction with file systems and object serialization techniques.
References	- 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

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 RelationshipsA: ssociation, Aggregation, Composition, and Multiplicity
7	Inheritance: Use of extends, super keyword, Method Overriding
8	Midterm Week: No classes - Midterm Exam

Week	Weekly Contents
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.