

## Semester: 4

**Methodological Teaching Unit:** UEM

**Subject:** Object-Oriented Programming

**Credits:** 4

**Coefficient:** 2

### Teaching Objectives

The objective of this course is to introduce the basic concepts of object-oriented programming (OOP) through practice using the Java language. Each chapter includes certain concepts that are translated into Java at its end so that the student can translate the theoretical concepts acquired into practice. By the end of the semester, the student is expected to have acquired the following skills:

1. Understanding the essence of object-oriented programming and its translation into the Java language.
2. Developing intuitive reasoning to provide a solution to a simple problem using the object-oriented approach.
3. Writing a functional program in the Java language.
4. Understanding the essence and importance of object-oriented reasoning and OOP.

### Prerequisite Knowledge

Language C

### Subject Content

#### Chapter 1. Basics of OOP

1. Introduction
2. Fundamental Concepts of OOP
  - Brief history of OOP
  - Procedural programming vs. object-oriented programming
  - Code reuse
  - Introduction to modularity
3. Objects and Classes
  - Object concepts
  - Class concepts
  - Attributes
  - Message concept
  - Problem-solving through message exchange
4. Introduction to Java
  - Types and control structures in Java
  - Classes and instantiation
  - Methods
  - References and parameter passing
  - Input/Output
  - Default constructor and other constructors
  - Destructors

## Chapter 2. Encapsulation

1. Visibility levels
2. Encapsulation
  - Data encapsulation (attributes)
  - Code encapsulation (messages)
3. Encapsulation in Java
  - Access control (public, private)
  - Accessors (get and set)
  - Instance access (this)
  - Class variables and methods (static)

## Chapter 3. Inheritance

1. Subclasses and inheritance
2. Simple inheritance, multiple inheritance
3. Class hierarchy
4. Polymorphism
5. Inheritance and polymorphism in Java
  - Simple inheritance (extends)
  - Encapsulation in inheritance
    - Member protection (protected)
    - Class constructors (this(), super())
    - 'Object' class
    - Implicit and explicit typecasting
    - Inheritance limitation (final)
  - Polymorphism
    - Method overloading
    - Method overriding
  - Abstract classes (usage and importance)
  - Interfaces (usage and importance)

## Evaluation Method

Exam (60%), Continuous assessment (40%)

## References

1. "Learn Object-Oriented Programming with the Java Language." Luc Gervais. Eni. 2nd edition.
2. <https://openclassrooms.com/courses/apprenez-a-programmer-en-java>
3. "Java 8 - Learn Object-Oriented Programming and master the language." Thierry GROUSSARD Luc GERVAIS. Edition ENI. 2015.
4. "Object-Oriented Programming in Java." Michel Divay. Edition DUNOD. 2006.