

---

## IS1220 Object Oriented Software Design

---

**Professor** : Paolo BALLARINI

**Language of instruction** : ANGLAIS – **Number of hours** : 36 – **ECTS** : 3,0 - **Quota** :

**Prerequisites** : IS1210,Algorithms

**Period** : S7 elective 2 between september and january  
S8 elective 8 between february and june

### Course Objectives

The course provides students with an introduction to the principles of design and implementation of component-based software systems based on object-oriented programming (OOP). The first part of the course is concerned with an introduction to the basic concepts of OOP (object, class, encapsulation, inheritance, polymorphism, etc..) followed by a second part dedicated to learning of « design patterns », that is learning of techniques for developing flexible programs (i.e. programs that can straightforwardly be expanded / modified). The course is based on the JAVA programming language and refers to the ECLIPSE Integrated Development Environment (IDE). The course also presents an introduction to high level modeling of systems as a means of supporting the Software Engineering process. To this aim the course contains an introduction to the language "Unified Modeling Language" (UML), which provide programming language independent notations for design.

More specifically the objectives of the course are summarised by the following points :

1. OOP basic concepts : core concepts of object-oriented programming ; classes, objects, messages ; thinking by classes ; inheritance : classes and subclasses ; motivation : why OOP is important and how it differs from non-OOP.
2. JAVA language basics : overview of the JAVA technology ; data types, variables, arrays, operators, control-flow instructions. Classes and objects : declaring classes ; class constructors ; class variables ; class methods ; creating and using objects.
3. Interfaces : what is an interface (motivation) ; how to declare interfaces ; interface as type. Inheritance : superclass and subclass ; inheritance of methods in a subclass ; overriding and hiding methods ; polymorphism ; abstract methods and abstract classes.
4. Exception handling : what is an exception ? (handling errors in JAVA code). Catching and handling (the « try », « catch », « throw » blocks ; Advantages of exceptions.
5. I/O operations in JAVA : I/O from command line ; I/O from files ; byte/character streaming ; scanning and formatting ;
6. Basic principles of concurrent programming. Concurrency in JAVA : multi-threaded programs : defining and running threads ; threads interference ; synchronization ;

7. UML modeling : class diagrams and their application to modelling of OO programs.
8. Introduction to code testing. Notion of test unit in JAVA with JUnit. Introduction to Test Driven Development (Agile Extreme programming)
9. Introduction to Graphical User Interface (GUI) programming in Java. Java AWT, Java Swing. Components, containers, buttons, menus, file chooser, etc. Adding control to a GUI: listeners.

### **On completion of the course, students should be able to**

The student will be able to develop, debug and test JAVA code through an IDE (e.g. Eclipse or else). The student will also have acquired a general knowledge about object-oriented programming, hence he/she will be able to apply the OO development philosophy in general to any software development issue.

The knowledge acquired in this course will also be fundamental for those students who wish to take the IS 1250 module (« Programmation tablette ») which requires a good knowledge of the basics of the JAVA language

### **Bibliography / Teaching Material and Textbooks**

All material supporting the course will be in english. This will include : the slides of each class of the course, and the text of tutorials (and solutions). The students will also be pointed out to supplementary materials, like : online JAVA tutorials. We will refer to several textbooks, like : « Thinking in JAVA », by Bruce Eckel (also available in PDF online), « Effective JAVA » by Joshua Bloch, « UML for Java programmers » by R.C. Martin.

### **Evaluation**

Written exam time 3-hr + JAVA Project noted (be made during the course and PW).