

**Code: ING/INF05**

**Credits: 9**

**Matter: Object Oriented Programming**

**Main language of instruction: Italian**

**Other language of instruction: English**

## Teaching Staff

### Head instructor

**Prof. Antonino LONGO MINNOLO - [antonino.longo@unicusano.it](mailto:antonino.longo@unicusano.it)**

### Introduction

#### *1. Objective of the course :*

The Object-oriented Programming Course aims to provide the student with a good knowledge of the principles governing object-oriented programming. The course proposes the basic concepts of object-oriented programming in general, and declines them in the study of the Java language, with particular reference to the specific aspects of Java, such as the organization of the code in package, the documentation of the code, the functionalities for programming competitor and the standard library. Furthermore, the educational objective of the Course is to provide the student with a detailed knowledge of the most frequently used data structures, framing these in the context of the Java library. The Etivities associated with the Course develop the skills necessary to implement algorithms and data structures in complete programs through the use of the Java language.

### Objectives

#### *2. Course Structure:*

- Illustrate the basic concepts in object-oriented programming
- Explain the peculiarities of the Java language
- Illustrate program development techniques according to the object-oriented programming paradigm through the use of the Java language
- Illustrate the typical data structures of object oriented programming
- Orienting the approach, development, construction of algorithms and data structuring and management



### **Competencies:**

#### A. Knowledge and understanding

At the end of the course, the student will have knowledge of the problems inherent in object-oriented programming and design and the development and maintenance of software written in Java. Furthermore, the student will acquire the knowledge in detail of the characteristics of the Java language such as operators, functions, control structures, classes and objects, inheritance, polymorphism, interfaces, methods and generic classes, exception handling, collections, I / O system. Furthermore, through the Etivities, students will acquire the ability to use the main Java programming environments.

#### B. Applying knowledge and understanding

The student will be able to design and develop complex software using the advanced features of Java; it will also be able to intervene on existing software in order to correct or increase its functionality. The Etivities provide for the application of theoretical knowledge to the implementation of simple C language programs, using the appropriate tools and development environments.

#### C. Making judgements

The student will be able to identify the most appropriate third-party classes and libraries for the services to be developed, analyzing their compliance with the resulting requirements, thus evaluating their possible adoption in the development of complex software.

#### D. Communication skills

The student will be able to describe and hold conversations on topics related to software development based on object-oriented programming, to highlight problems relating to design and implementation, and to offer solutions, using appropriate terminology.

#### E. Learning skills.

At the end of the course, the student will have knowledge of the fundamental notions necessary for the design and development of applications and services according to the object-oriented programming paradigm. This will allow him to independently address any problems relating to the development of software with object languages, being able to deepen complex issues such as polymorphism, dynamic memory management, management competition, and the scalability of the software, providing useful bases to learn what will be proposed in the specialized courses of computer engineering, with particular reference to the topics related to the knowledge of programming languages and paradigms.

### **Syllabus**

#### *3. Programme of the course:*



### **Subject 1 - Introduction to object-oriented programming and Java**

Introduction to Java. Fundamental components of a Java program: the basics of object-oriented programming, methods in Java, variables, constructor methods, packages. Exercise on development environment and compilation of simple Java programs.

### **Subject 2 - Data types**

Identifiers, data types and arrays: primitive and non-primitive data types, arrays in Java. Array tutorial.

### **Subject 3 - Execution flow management**

Operators and execution flow management: basic operators, execution flow management, simple and advanced programming constructs. Exercise on the use of constructs.

### **Subject 4 – Encapsulation**

Encapsulation and visibility: object-oriented programming paradigms, encapsulation, access modifiers, the static modifier. Tutorial on encapsulation and visibility.

### **Subject 5 - Inheritance and polymorphism**

Inheritance and interfaces: inheritance, the final modifier, the Object class, inheritance and encapsulation, interfaces; Polymorphism: polymorphism for methods, overloads and overrides, polymorphism for data. Tutorial on inheritance and polymorphism.

### **Subject 6 - Exceptions and generic types**

Exceptions and assertions: the mechanism for handling exceptions in Java, the propagation of the exception, introduction to assertions; Enumerations and nested types: internal classes, anonymous classes, enumeration types; Generic Types: Create your own generic types, automatic type deduction. Exercise on the use of exceptions and generic types.

### **Subject 7 - Libraries and concurrency**

Library: the String, Object, System, Runtime and Math classes, the java.util package, Date-Time API; Thread management: the Thread class, thread synchronization, inter-thread communication, concurrency management. Exercise on the use of the Java library and on concurrency management.

## **Subject 8 - Collections, I / O and GUI**

Collections: Collection, List, Queue and Deque interfaces, Map and SortedMap, utility algorithms; Input / Output: character and byte stream, keyboard input reading, file management, object serialization; Graphic interfaces: JavaFX, creation of interfaces with Layouts, event management. Exercise on the use of collections, the I / O system, graphic interfaces.

### **Evaluation system and criteria**

The exam consists in carrying out a written test aimed at ascertaining the ability to analyze and rework the concepts acquired and a series of activities (e-tivity) carried out during the course in virtual classrooms.

The expected learning outcomes about the knowledge of the subject and the ability to apply them are assessed by the written test, while the communication skills, the ability to draw conclusions and the ability to self-learn are assessed in itinere through e-tivities.

### **Bibliography and resources**

#### *4. Materials to consult:*

Notes written by the instructor are available in English. The notes cover the course contents and examination programme.

#### *5. Recommended bibliography:*

6.

Suggested readings are:

- Herbert Schildt, "Java la guida completa". McGrawill