COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

Course title	**Computer Programming**
Course code	76203
Scientific sector	INF/01
Degree	Bachelor in Computer Science (L-31)
Semester	1
Year	1
Credits	9
Modular	No

Total lecturing hours	60
Total lab hours	30
Attendance	Not mandatory, but highly recommended.
Prerequisites	There are no specific prerequisites. Basic notions of mathematics and set theory will be used.
Course page	https://ole.unibz.it/

Specific educational objectives	Type of course: "di base" for L-31 Scientific area: "Formazione informatica di base" for L-31 The objective of the course is to teach the fundamental principles of programming. We will focus especially on imperative programing as the basic way to learn: (1) the basics of programming and programming elements; (2) the basics of algorithmic thinking; and (3) The basics of writing code. As programming language, we will use a subset of the Java language, mainly restricted to its imperative part. The student will learn how programs can be constructed, and also structured in more files/objects in order to solve a problem. Students will learn how to solve computational problems with well-designed programs that implement effective solutions. The learning will be based on examples, from very simple ones to more complex. We will use the Java programming language and the integrated development environment (IDE), so the goal is to train the student capability to develop java applications in this environment. The final objective for the student is to acquire the ability to solve basic algorithmic problems in a Java-based application.

Lecturer	Chiara Ghidini
Contact	Faculty of Engineering, Piazza Domenicani 3, chiara.ghidini@unibz.it
Scientific sector of lecturer	INF/01
Teaching language	English
Office hours	By previous appointment by e-mail
Lecturing Assistant (if any)	Ozan Kahramanogullari
Contact LA	Piazza Domenicani 3, Office 2.14, ozan.kahramanogullari@unibz.it@unibz.it
Office hours LA	To be arranged beforehand by email.

List of topics	• Data types and expressions • Basic data structures and generics • Functions and parameter passing • Conditionals and loops • Arrays and collections • Classes and objects • Basic Input/Output • Exception handling • Recursion
Teaching format	Frontal lectures interleaved with exercises, labs with exercises, individual programming projects.

Learning outcomes	Knowledge and understanding: • D1.3 - Know the basic principles of programming. Applying knowledge and understanding: • D2.2 - Ability to solve algorithmic problems using programming methods. • D2.17 - Know how to manage small projects for the development of information systems and how coordinate small working groups. Communication skills • D4.5 - Ability to collaborate in interdisciplinary teams to achieve IT objectives. Learning skills • D5.1 - Learning ability to undertake further studies with a high degree of autonomy.

Assessment	Programming Project and a final exam (written). In the project part of the exam we will assess the learning outcomes related to the application of the acquired knowledge, the ability to make judgments and the communication and learning skills. In fact, the goal of the project is to design a computer application that can effectively solve problems. The project part must be positively evaluated to be allowed to attend the written exam. In the assignments, the students will reply to transfer of knowledge questions and programming exercises. In the written exam, there will be verification questions, transfer of knowledge questions and exercises. The learning outcome related to knowledge and understanding, applying knowledge and understanding and those related to the student ability to learn.
Assessment language	English
Assessment Typology	Monocratic
Evaluation criteria and criteria for awarding marks	Project counts for 40 % of mark (12 points), and the final exam (written) for 60 % of the mark (18 points). In case of a positive mark the project will count for all the three yearly regular exam sessions. .

	Project is evaluated in term of quality of the solution: easy to use, meaningfulness of the implemented functions, quality of the code (according to the principles that will be illustrated during the lectures). Written exam questions will be evaluated in term of correctness and clarity.

Required readings	John Lewis and William Loftus, Java Software Solutions, Pearson, 2018. Kathy Sierra, Bert Bates, Trisha Gee, Head First Java, 3rd Edition. O'Reilly Media, Inc. Cay S. Horstmann, Brief Java: Early Objects. Wiley Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it
Supplementary readings	The Java Tutorials: https://docs.oracle.com/javase/tutorial/
Software used	IntelliJ