Syllabus

Course Description

Course Title	Introduction to Programming
Course Code	76271
Course Title Additional	
Scientific-Disciplinary Sector	INFO-01/A
Language	English
Degree Course	Bachelor in Computer Science
Other Degree Courses (Loaned)	
Lecturers	dr. Tiziano Dalmonte, Tiziano.Dalmonte@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/47069 Prof. Chiara Ghidini, Chiara.Ghidini@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/49601
Teaching Assistant	
Semester	First semester
Course Year/s	1
CP	9
Teaching Hours	60
Lab Hours	30
Individual Study Hours	135
Planned Office Hours	
Contents Summary	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.
Course Topics	• Data types and expressions • Basic data structures and generic • Functions and parameter passing • Conditionals and loops • Arrays and collections • Classes and objects • Basic Input/Output • Exception handling • Recursion
Keywords	Programming, Algorithms, Java, Object Oriented
Recommended Prerequisites	The course requires knowledge of basic mathematics and set theory.
Propaedeutic Courses	
Teaching Format	The course includes frontal lectures with exercises, lab sessions, and individual programming projects.
Mandatory Attendance	Not mandatory, but highly recommended
Specific Educational Objectives and Learning Outcomes	Knowledge and Understanding - D1.2: Know in details the fundamental principles of programming - D1.3: Have a solid knowledge of the most important data structures and programming techniques Applying knowledge and understanding - D2.2: Be able to develop small and medium size programs using different programming languages and paradigms.

	- D2.3: Be able to solve problems using programming methodologies. Ability to make judgments - D3.1: Be able to collect and interpret useful data and to judge information systems and their applicability. Communication skills - D4.1: Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately. Learning skills - D5.1: Have developed learning capabilities to pursue further studies with a high degree of autonomy.
Specific Educational Objectives and Learning Outcomes (additional info.)	
Assessment	The assessment consists of a programming project and a final written exam. The project is designed to evaluate learning outcomes related to the application of acquired knowledge, critical thinking, communication, and learning skills. Specifically, students are expected to design a computer application capable of effectively solving a given problem. The written exam assesses knowledge and understanding, the ability to apply that knowledge, and the student's learning skills. It includes verification questions, transfer-of-knowledge questions, and practical exercises.
Evaluation Criteria	The project accounts for 40% of the final grade (12 points), while the written exam represents 60% (18 points). If the project receives a positive evaluation, the result remains valid for all three regular exam sessions within the academic year. The project will be assessed based on the quality of the solution, including ease of use, the relevance and effectiveness of the implemented functions, and the quality of the code, in line with the principles discussed during the lectures. Written exam answers will be evaluated based on their correctness and clarity.
Required Readings	• John Lewis and William Loftus. Java Software Solutions. PEARSON INDIA, 2018. ISBN 978-93-5306-361-0.

	• Kathy Sierra, Bert Bates, and Trisha Gee. Head First Java: A Brain-Friendly Guide. O'Reilly Media, Sebastopol, CA, 3rd edition, June 2022. ISBN 978-1-4919-1077-1. • Cay S. Horstmann. Brief Java: Early Objects. John Wiley & Sons Inc, 9th edition, 2020. ISBN 978-1-119-74019-3.
Supplementary Readings	The Java Tutorials at https://docs.oracle.com/javase/tutorial/
Further Information	IntelliJ IDEA (https://www.jetbrains.com/idea/)
Sustainable Development Goals (SDGs)	Quality education