unibz

Fakultät für Ingenieurwesen
Facoltà di Ingegneria
Faculty of Engineering

SYLLABUS
COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

COURSE TITLE	**Programming Project**
COURSE CODE	76204
SCIENTIFIC SECTOR	INF/01
DEGREE	Bachelor in Computer Science
SEMESTER	2nd
YEAR	1st
CREDITS	9

TOTAL LECTURING HOURS	60
TOTAL LAB HOURS	30
ATTENDANCE	Attendance to lectures and labs is optional. However, non-attending students should contact the lecturer at the start of the course to discuss the modality of their independent study. The evaluation process is identical for attending and non-attending students. It is described in the fields "Assessment" and "Evaluation criteria and criteria for awarding marks" below.
PREREQUISITES	Students should be familiar with the basics of imperative/object-oriented programming and Java, as taught in the course "Computer Programming"
COURSE PAGE	Microsoft TEAMS

SPECIFIC EDUCATIONAL OBJECTIVES	Type of course: "caratterizzanti" for L-31 Scientific area: "Discipline informatiche" for L-31 The course is designed for students to develop generic and object-oriented programming skills, and acquire a first experience of software development within a team. Students will program in Java, but a large part of the course's content can be transferred to other programming languages. After completing this course, students should be able to: - design and develop a prototype application in Java, - develop algorithms to solve simple programming problems (and select appropriate data structures), - write readable, concise, modular and documented code, - collaborate with other programmers.

LECTURER	Julien Corman
SCIENTIFIC SECTOR OF THE LECTURER	INF/01
TEACHING LANGUAGE	English
OFFICE HOURS	Mondays 16:00-18:00 by prior email appointment. Office POS 2.06, Faculty of Engineering, Piazza Domenicani 3. corman@inf.unibz.it
TEACHING ASSISTANT	TBA
OFFICE HOURS	
LIST OF TOPICS COVERED	• Objects and Classes: Interfaces, Inheritance and Polymorphism • Abstract data types (set, list, map, queue) • Dynamic Data structures (linked list, hashmap) • Recursion • Serialisation and streams • Mutability, pure functions and lambda expressions • Multithreading • IDE: Git, Source code management, and build automation • Project
TEACHING FORMAT	Frontal lectures, lab exercises, group projects.

LEARNING OUTCOMES	**Knowledge and understanding** • Know in details the fundamental principles of programming. • Have a solid knowledge of the most important data structures and programming techniques. **Applying knowledge and understanding** • Be able to develop small and medium size programs using different programming languages and paradigms. • Be able to solve problems through the application of programming methodologies. **Making judgments** • Be able to collect and interpret useful data and to judge information systems and their applicability. • Be able to work autonomously according to the own level of knowledge and understanding. **Communication skills** • Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately.

	• Be able to structure and write scientific documentation. **Learning skills** • Have acquired learning capabilities to pursue further studies with a high degree of autonomy.

ASSESSMENT	The assessment is based on: • **assignments**, which focus on topics taught during lectures (they are meant to motivate students to study throughout the semester and consolidate the theoretical notions taught in class), • a **group project**, which evaluates whether students acquired the expected programming knowledge and skills, and • an individual **oral exam**, during which students explain and answer questions about their contribution to the group project. The evaluation of the assignments and the group project is individual (in particular, the latter is based on each student's contribution).
ASSESSMENT LANGUAGE	English
EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS	Final marks will be calculated in the following way: • up to 30 points will be awarded to assignments; • up to 60 points will be awarded to the group project (based on individual contributions); • up to 10 points will be awarded to the oral exam. To enroll in the oral exam, a student must: • deliver the group project, and • have been awarded at least 50 points for the assignments and the group project combined.

TEXTBOOKS	• Kathy Sierra, Bert Bates, Trisha Gee. **Head First Java**, 3rd edition, 2022, O'Reilly Media, Inc., ISBN 9781491910771. • Herbert Schildt. **Java: The Complete Reference**, Eleventh Edition, 11th edition, 2018, McGraw-Hill, ISBN 9781260440249. • Lecture’s materials on TEAMS.
SUPPLEMENTARY READINGS	• Joshua Bloch. **Effective Java**, 3rd edition, 2017, Addison-Wesley Professional, ISBN 9780134686097. • Robert C. Martin. **Clean Code**, 2008, Prentice Hall, ISBN 9780136083238. • Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea. **Java Concurrency in Practice**, 2006, Addison-Wesley Professional, ISBN 0321349601.

	• Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. **Design Patterns: Elements of Reusable Object-Oriented Software**, 1994, Addison-Wesley Professional, ISBN 0201633612. • Prem Kumar Ponuthorai, Jon Loeliger. **Version Control with Git**, 3rd edition, 2022, O'Reilly Media, Inc., ISBN 9781492091196. • Shekhar Gulati, Rahul Sharma. **Java Unit Testing with JUnit 5: Test Driven Development with JUnit 5**, 2017, Apress, ISBN 9781484230152.
SOFTWARE USED	• IDE for Java programming, e.g. Eclipse (https://www.eclipse.org/), IntelliJ IDEA (https://www.jetbrains.com/idea/), Visual Studio Code (https://code.visualstudio.com/) or NetBeans (https://netbeans.apache.org/) • JDK 17 • git • Maven • Linux or macOS recommended.