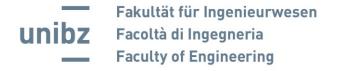


SYLLABUS COURSE DESCRIPTION – ACADEMIC YEAR 2023/2024

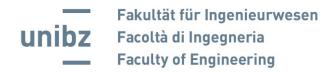
COURSE TITLE	Programming Project
COURSE CODE	76204
SCIENTIFIC SECTOR	INF/01
DEGREE	Bachelor in Computer Science
SEMESTER	2nd
YEAR	1st
CREDITS	9

TOTAL LECTURING HOURS	60
TOTAL LAB HOURS	30
ATTENDANCE	Attendance to course lectures and labs is optional. However, non-attending students have to contact the lecturer at the start of the course to discuss the modality of their independent study.
	The exam modality for both attending and non-attending students is the same, which is described in the fields "Assessment" and "Evaluation criteria and criteria for awarding marks" below.
PREREQUISITES	Students should be familiar with the basic knowledge of object-oriented programming and Java, as taught in the course "Computer Programming"
COURSE PAGE	https://ole.unibz.it/



SPECIFIC EDUCATIONAL OBJECTIVES	Type of course: "caratterizzanti" for L-31 Scientific area: "Discipline informatiche" for L-31 The course is designed to provide generic and object-oriented programming skills, as well as a first experience of software development within a team. Students will program in Java, but a large part of the acquired knowledge 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	Albulen Pano
OFFICE HOURS	Mondays 16:00-18:00 by prior email appointment. Office POS 2.08, Faculty of Engineering, Piazza Domenicani 3. albulen.pano@unibz.it
LIST OF TOPICS COVERED	 Memory models in Java Virtual functions, late binding, overriding, and overloading Exception handling Reflection and runtime type identification Generics and collections I/O, serialization and XML/JSON processing Designing large applications: design patterns Multithreading Code optimization



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

ASSESSMENT	 lab exercises, which are focused on specific topics taught in the course. They are meant to motivate students to study throughout the semester and consolidate the theoretical concepts taught in class; a group project, which evaluates if students acquired the expected programming knowledge and skills; and an oral exam, which evaluates if students assimilated of the theoretical concepts taught in class by reviewing and discussing the group project.
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;

high degree of autonomy.

• Have acquired learning capabilities to pursue further studies with a



• up to 10 points will be awarded to the oral exam;

To enroll in the oral exam, a student must:

- Deliver the group project;
- Have earned 40 points from the weekly assignments and the group project combined.

TEXTBOOK	 Paul Deitel, Harvey Deitel. Java How to Program, Early Objects, Eleventh Edition, Pearson, 2017, ISBN 9780134751962. Permanent link: http://ubz-primo.hosted.exlibrisgroup.com/UNIBZ:All:39UBZ ALMA DS51129685850001241 Lecture notes handed out during the course
SUPPLEMENTARY READINGS	 Jon Loeliger, Matthew McCullough. Version Control with Git, 2nd Edition, 2012, O'Reilly Media, Inc., ISBN 9780596520120. Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/pok0fm/39UBZ_ALMA_DS51185 432710001241 Robert C. Martin. Clean Code, 2008, Prentice Hall, ISBN 9780136083238. Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/1t65344/39UBZ_ALMA_DS5120 8133560001241 Jeff Friesen. Java I/O, NIO and NIO.2, Apress, 2015, ISBN 9781484215654. Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/pok0fm/39UBZ_ALMA_DS51184 522190001241 Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea. Java Concurrency in Practice, 2006, Addison-Wesley Professional, ISBN 0321349601. Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/pok0fm/39UBZ_ALMA_DS51184 497210001241 Shekhar Gulati, Rahul Sharma. Java Unit Testing with JUnit 5: Test Driven Development with JUnit 5, 2017, Apress, ISBN 9781484230152. Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/pok0fm/39UBZ_ALMA_DS51208 247310001241 Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, 1994, Addison-Wesley Professional, ISBN 0201633612. Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/1t65344/39UBZ_ALMA_DS5118 4149850001241
SOFTWARE USED	 IDE for Java programming, e.g. Eclipse (https://www.eclipse.org/), IntelliJ (https://www.jetbrains.com/idea/), Visual Studio



(https://code.visualstudio.com/) or NetBeans (https://netbeans.apache.org/)

- JDK 17
- Git
- Linux or macOS recommended.