Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

SYLLABUS
COURSE DESCRIPTION

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
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 give specific professional skills. It will provide students with advanced techniques in Java. In particular, students will acquire knowledge in the overall architecture and components of Java SDK and JRE, use of advanced programming techniques (e.g. multi-threads, reading/writing streams, generics, regular expressions, exception handling and testing) and code documentation (e.g. generate API documentation) of the software developed.

LECTURER	Tiago Prince Sales,
SCIENTIFIC SECTOR OF THE LECTURER	INF/01
TEACHING LANGUAGE	English
OFFICE HOURS	Wednesday 14:00-16:00 previous appointment, office POS 3.05, Tiago.PrinceSales@unibz.it office POS 3.05, Faculty of Computer Science, Piazza Domenicani 3

TEACHING ASSISTANT	Riccardo Billero, Alam Syed Mehtab, and Roberto Confalonieri
OFFICE HOURS	Riccardo Billero: Monday 13:00-14:00 riccardo.billero@unibz.it Office POS 1.04, first floor, Faculty of Computer Science, Piazza Domenicani 3 Alam Syed Mehtab: Monday 13:00-14:00 MehtabAlam.Syed@unibz.it Office POS 2.08, second floor, Faculty of Computer Science, Piazza Domenicani 3 Roberto Confalonieri: Tuesday 16:00-18:00 previous appointment by email roberto.confalonieri@unibz.it office POS 2.11, second floor, Faculty of Computer Science, Piazza Domenicani 3
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 basic and advanced programming techniques • Have a basic knowledge of the most important data structures and their use in programming languages **Applying knowledge and understanding** • Be able to develop small and medium size programs using Java • Be able to solve problems through the application of programming methods **Making judgments** • Be able to work autonomously according to the own level of knowledge and understanding **Communication skills** • 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: (i) weekly **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; (ii) a **group project**, which evaluates if the students acquired the expected programming knowledge and skills; and

	(iii) an **oral exam**, which evaluates the students' understanding 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: (i) up to 10 points will be awarded to the solutions of the weekly lab exercises; (ii) up to 60 points will be awarded to the group project; (iii) up to 30 points will be awarded to the oral exam; In order to enroll for the oral exam, the students must have been awarded at least 30 points on (i) and (ii) together.

REQUIRED READINGS	Lecture notes will be handed out during the course.
SUPPLEMENTARY READINGS	Paul Deitel; Harvey Deitel, Java™ How to Program (Early Objects), Tenth Edition, Prentice Hall, 2014, ISBN 978-0-13-381303-6, available through Library access to Safari Books Online, Permanent link: https://ubz-primo.hosted.exlibrisgroup.com/permalink/f/pok0fm/39UBZ_ALMA_DS51184521750001241
SOFTWARE USED	- IDE for Java programming, e.g. Eclipse (https://www.eclipse.org/), IntelliJ IDEA (https://www.jetbrains.com/idea/), BueJ (https://www.bluej.org/), NetBeans (https://netbeans.org/). - JDK