SYLLABUS
COURSE DESCRIPTION YEAR 2024/2025

COURSE TITLE	**Software Systems Engineering**
COURSE CODE	76261
SCIENTIFIC SECTOR	INF/01
DEGREE	Bachelor in Computer Science
SEMESTER	2nd
YEAR	2nd
CREDITS	12
MODULAR	Yes

TOTAL LECTURING HOURS	70
TOTAL LAB HOURS	50
ATTENDANCE	Attendance is not compulsory. Non-attending students have to contact the lecturer at the start of the course to agree on the modalities of the independent study.
PREREQUISITES	
COURSE PAGE	https://ole.unibz.it/

SPECIFIC EDUCATIONAL OBJECTIVES	• Type of course: caratterizzanti • Scientific area: discipline informatiche **MODULE 1**: To understand the role played by software architecture in software development lifecycle; to design software architecture based on patterns and best practices; to obtain an overview of different software architecture styles and the newest trends in software architecting; to evaluate and balance trade-offs of quality attributes on software architecture; and to learn how to apply different software architecture styles to develop high quality software. **MODULE 2**: The goal of the course is to enable students to be able to select, use, customize, and deploy tools and apply techniques for software testing. The students will also be able to set up and integrate software tools to test software all through the collaborative development process.

MODULE 1	**Software Systems Architecture**

MODULE CODE	76261A
MODULE SCIENTIFIC SECTOR	INF/01
CREDITS	6
LECTURER	Eduardo Martins Guerra
SCIENTIFIC SECTOR OF THE LECTURER	INF/01
TEACHING LANGUAGE	**English**
OFFICE HOURS	Thursday, 14:00-16:00, Office POS 1.13, eduardo.martinsguerra@unibz.it, +39 375 6071913. Arrange by email.
TEACHING ASSISTANT	Same as lecturer
OFFICE HOURS	-
LIST OF TOPICS COVERED	• Software and systems architecture principles • Architecture process and activities: specification, validation • Architectural description and modeling • Stakeholders and viewpoints • Quality considerations: security, performance, modifiability • Patterns of systems architectures
TEACHING FORMAT	Frontal lectures, exercises in lab.

MODULE 2	**Tools and Techniques for Software Testing**
MODULE CODE	76261B
MODULE SCIENTIFIC SECTOR	INF/01
CREDITS	6
LECTURER	Barbara Russo
SCIENTIFIC SECTOR OF THE LECTURER	INF/01
TEACHING LANGUAGE	**Italian**
OFFICE HOURS	Tuesday, 14:00-16:00, Office POS 1.15, barbara.russo@unibz.it , first floor, Faculty of Engineering, +390471016170

TEACHING ASSISTANT	Same as lecturer
OFFICE HOURS	-
LIST OF TOPICS COVERED	• Techniques for black box and white box testing • Automated testing • Dynamic Testing • Static testing • Performance and monitoring • Introduction to search-based testing
TEACHING FORMAT	Frontal lectures, exercises in lab.

LEARNING OUTCOMES	**Knowledge and understanding** • To have a thorough knowledge of the main fundamentals techniques and methods of software design, development and maintenance **Applying knowledge and understanding** • Be able to apply the own knowledge to the analysis, design, development and testing of information systems which satisfy given requirements; • Be able to solve typical problems in computer science based on software engineering methodologies, such as the definition of requirements, the analysis of possible methods for a solution, the selection of the most appropriate methods and tools as well as their application **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. • Be able to take the responsibility for development of projects or IT consulting. **Ability to learn** • Have developed learning capabilities to pursue further studies with a high degree of autonomy. • Have acquired learning capabilities that enable to carry out project activities in companies, public institutions or in distributed development communities. • Be able to follow the fast technological evolution and to learn cutting edge IT technologies and innovative aspects of last generation information systems.

ASSESSMENT	A final unified written exam will be performed with verification questions from both modules. Each module will have the following additional activities during the course: **MODULE 1**: Lab assessment is composed by assignments that should be performed and delivered in each week. Optional activities might worth extra points in the final module grade. **MODULE 2**: Project work with assignments done in groups

ASSESSMENT LANGUAGE	English
EVALUATION CRITERIA AND CRITERIA FOR AWARDING MARKS	Written exam: being able to master the terminology of the course; being able to evaluate tools and techniques and their technical details for specific domain of use; being able to solve exercises or summarize theoretical concepts. Project/Lab work assessment: complete assignments by the due date; provide working solutions; develop/customize quality software that fulfills the assignments' tasks. The final grade will be calculated by the average value considering the grade of each module. The grade of each module will be calculated as follows: • **Module 1**: The grade is calculated based on (i) the lab assessment (50% weight) and (ii) the module questions from the final written exam (50% weight). Optional activities in a module can worth extra points in that module grade. • **Module 2**: 80% project work and 20% written exam module questions. Need to pass project work to access to written exam. For non-attending students, two options might be provided for the module teacher: (a) they can delivery lab assessments or project as the attending students; (b) or they can perform an additional evaluation with questions related to the lab or project content. Both modules must be positive to pass the course. A positive evaluation of one module remains valid for all three regular exam sessions of the academic year.

REQUIRED READINGS	• Robert C. Martin. 2017. Clean Architecture: A Craftsman's Guide to Software Structure and Design (1st ed.). Prentice Hall Press, Upper Saddle River, NJ, USA. • Mark Richards. 2015. Software Architecture Patterns. O'Reilly Media, Inc..
SUPPLEMENTARY READINGS	• Len Bass, Paul Clements, and Rick Kazman. 2012. Software Architecture in Practice (3rd ed.). Addison-Wesley Professional.
SOFTWARE USED	**Module 1**: Java JDK, Eclipse (or other Java IDE), other open-source tools. **Module 2**: The course will extensively use software for testing. The following tools is non-exhaustive list: Java 8; Eclipse IDE; git; Maven; Issue tracker; JaCoco; FindBugs; Junit 5; Fitnesse; PPTAM.