

COMPUTER ENGINEERING (LM75)

(Lecce - Università degli Studi)

Teaching SERVICE ARCHITECTURE DESIGN

GenCod A006804

Owner professor ROBERTO VERGALLO

Teaching in italian PROGETTAZIONE DI ARCHITETTURE DI SERVIZI

Teaching SERVICE ARCHITECTURE DESIGN

SSD code ING-INF/05

Reference course COMPUTER ENGINEERING

Course type Laurea Magistrale

Credits 6.0

Teaching hours Front activity hours: 54.0

For enrolled in 2024/2025

Taught in 2024/2025

Course year 1

Language ITALIAN

Curriculum PERCORSO COMUNE

Location Lecce

Semester First Semester

Exam type Oral

Assessment Final grade

Course timetable

<https://easyroom.unisalento.it/Orario>

BRIEF COURSE DESCRIPTION

After the course the student should be able to: a. Apply main software engineering principles and control software qualities (both internal and external); b. Design and implement software following industrial standards (UML) and structured software production processes; c. Manage the software engineering i.e. execute tasks as planning, organizing, staffing, controlling, estimating (software cost and size); d. Design the software adopting standard software architectures; e. Select and adopt software design patterns (creational patterns, structural patterns, behavioral patterns); f. Verify the software exploiting standard tools and adopting well-known metrics; g. Develop complex model-view-controller web and mobile software systems, exploiting at the back end the Spring framework, and at the front end the React framework, connecting them through REST/JSON web services; h. Manage the fundamentals of modern cloud computing and cloud service deployment; i. Use the main open source tools for the software testing and refactoring, and for the software configuration management.

REQUIREMENTS

The prerequisites for attending the course are the knowledge of structured programming languages (Java) and the fundamentals of computer science.

COURSE AIMS

The main goal of the course is to deepen students' knowledge on modern design and development techniques for interactive software systems. In particular, methods and tools for automated software testing, dev ops and design patterns for micro service architectures. All concepts will be experimented by students designing, developing and testing a software prototype of a service based web application with a mobile extension (app). The software prototype will be developed on top of modern frameworks (Spring, React).

TEACHING METHODOLOGY

Classroom and online lessons, classroom and online practice, project work.

ASSESSMENT TYPE

The module will be verified with a small software prototype implementation, intended to verify the practice of micro-service architectures and tests, which will be discussed during an oral examination. The software system must be designed using UML, adopting standard design patterns. The software system must be developed starting from MVC frameworks (Spring, React), and must be systematically tested collecting metrics. The software prototype must be developed following an agile process and must be documented. A month before the end of the course, the general requirements of the software prototype will be published by the teacher, a new requirements set for each year. The requirements will be effective till a new set of specifications will appear.

REFERENCE TEXT BOOKS

Ian Sommerville - Engineering Software Products: An Introduction to Modern Software Engineering - Pearson, 2020.