



UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Matematica e Informatica
ACADEMIC YEAR	2023/2024
BACHELOR'S DEGREE (BSC)	COMPUTER SCIENCE
SUBJECT	SOFTWARE ENGINEERING AND SECURITY
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50166-Discipline Informatiche
CODE	20632
SCIENTIFIC SECTOR(S)	INF/01
HEAD PROFESSOR(S)	ROMBO SIMONA ESTER Professore Ordinario Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	6
INDIVIDUAL STUDY (Hrs)	78
COURSE ACTIVITY (Hrs)	72
PROPAEDEUTICAL SUBJECTS	
MUTUALIZATION	
YEAR	3
TERM (SEMESTER)	1° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	ROMBO SIMONA ESTER Monday 9:30 13:30 In presenza: Via Archirafi 34, Piano II, Stanza 220 - Telematico: via Microsoft Teams o altro canale - In entrambi i casi, e' consigliabile prenotarsi tramite email alla docente

DOCENTE: Prof.ssa SIMONA ESTER ROMBO

PREREQUISITES	Programming and object-oriented programming. Elements of database design and querying.
LEARNING OUTCOMES	The student will learn: basic knowledge and understanding of software engineering; knowledge of the Agile methodology and the use of UML in this field. Ability to apply Agile to the design of an application, as complex as in industrial contexts, with possible applications in the field of Artificial Intelligence; knowledge of design and architectural design patterns; ability to identify the utility and applicability of design patterns to a software project. Autonomy of judgment in formulating the design choices. Communication skills in presenting the choices made and the proposed solution, also in front of users with same or larger expertise. Ability to learn new methodologies for software design an implementation.
ASSESSMENT METHODS	The examination consists of two parts: (1) design and development of a project, which reflects the complexity of real world applications, and (2) oral examination. The project is carried out in team, partly during some laboratory hours and partly during the individual study time. The evaluation of the project takes place through revisions and a final presentation, to which the whole team must participate, needed to verify the ability to apply Software Engineering principles, team working, autonomy in judgement. The evaluation of the project is expressed in thirtieths. The oral exam focuses on the topics covered during the course and the evaluation is, again, expressed in thirtieths. The final evaluation is given by the average of the two evaluations obtained in the two parts that make up the examination. In particular, an evaluation of more than 28/30 indicates an excellent knowledge of the course topics and an excellent ability to design following the principles learned. In case of maximum score (30), the teacher will attribute laude to those students able to design effective and efficient solutions for complex problems. An evaluation between 24/30 and 27/30 indicates that the design skills are good, but could be improved, or that the knowledge of the teaching topics could be more thorough. An evaluation of less than 24/30 means that both design skills and knowledge of the topics could be improved.
EDUCATIONAL OBJECTIVES	The course is an introduction to concepts of Software Engineering and software development, based on the Unified Process (UP) Agile methodology, which adopts UML as its main modeling language. The students will be required to develop a non-trivial application by clearly identifying its business logic, data base interactions and other interfaces needed to fulfill the requirements. Moreover they will learn how to build and manage a project plan based on team working, goals and development phases.
TEACHING METHODS	The course has a strong practical focus. For this reason it consists of 24 classroom hours and a 48 lab hours. This way students will have the chance to apply the methodologies at the basis of this course. The considered case studies come directly from the real world, reflecting the complexity of systems implemented within industrial contexts. The design and development methodologies and tools are up-to-date with those currently used in the industries/organizations. Both direct (through invited seminars) and indirect (through provided and/or reviewed material) interactions with professionals from industry are planned. Brain storming are organized, as well as the presentation of solutions in front of a "peer" audience, composed of other students and, possibly, teachers and professionals from the industrial world.
SUGGESTED BIBLIOGRAPHY	Ingegneria del Software, 10/Ed. Con MyLab, Ian Sommerville, Pearson, ISBN: 9788891902245 (2017). Patterns of Enterprise Application Architecture, Martin Fowler, Addison Wesley (2003).

SYLLABUS

Hrs	Frontal teaching
6	FUNDAMENTALS: Presentation of the course. Recalling Object Oriented Programming. Professional software development, best practices and versioning. Software process models, process activities, process improvement. Software development with agile methods. Agile development techniques. Agile Project Management: Scrum. Practical examples of design according to plan-driven (cascade, using Gantt) and Agile (Scrum, with user stories) methodology.
6	REQUIREMENTS, ANALYSIS AND DESIGN: Large-scale agile development strategies. Examples and discussion. Elements of Requirements Engineering. User Stories and Scenarios. Unified Modeling Language (UML). Use cases. System models and main diagrams (use case, sequence, status, class).

SYLLABUS

Hrs	Frontal teaching
6	DESIGN PATTERNS: The design pattern theory. Patterns of the "Gang of Four" (GoF). Relationships between design patterns. Other design patterns. Complete catalog of GoF patterns modeled in UML and implemented in Java. Inversion of Control. Dependency Injection. Pattern based software architectures. Systematic development of pattern-based software.
2	TESTING: Test-driven design. Development Test (units, components, system integration). Release test. User Test.
4	SECURITY AND DISTRIBUTED: Open Web Application Security Project (OWASP) and list of applications at risk. Examples of non-secure software design. Principles for the design of secure software. Secure design patterns. Peer-to-Peer architectures.
Hrs	Workshops
6	DESIGN PATTERNS: Exercises and implementation on design patterns.
6	TESTING: Practical application on examples and implementation. Test-driven design: JUnit and applications.
6	SECURITY BY DESIGN: Exercises and applications.
6	DESIGN OF A CASE STUDY: Distinction between functional and non-functional requirements in the specific case of the considered project. Logical architectural design of the system.
6	DESIGN OF A CASE STUDY: Formulation of user stories (in the form WHO, WHAT, WHY) for the assigned project. Sprint definition and sizing. Sprint backlog.
6	DESIGN OF A CASE STUDY: Sprint Poker planning. Design of Web Applications, including frameworks and technologies.
6	DESIGN OF A CASE STUDY: Application of Agile Methodology: first we design the units using the design patterns, then we write the tests in Junit, then we write the code of the units; at the end of each Sprint the system test is carried out, which will gradually have more complex functions.
6	DESIGN OF A CASE STUDY: Continuation of design and development, completing at least one sprint. Changes in the requirements. Refactoring.