Course Name : Software Engineering

Course Code	Course Type	Regular Semester	Lecture (hours/week)	Seminar (hours/week)	Lab. (hours/week)	Credits	ECTS
CMP 315	B	Fall	2.00	0.00	2.00	3.00	6.00

Lecturer	Hersi Kopani, Msc
Assistant	
Course language	Albanian
Course level	Bachelor
Description	This course provides an in-depth exploration of object-oriented software engineering principles through the use of UML and Java. It covers essential topics such as design patterns, software architecture, and the software development lifecycle. Students will engage in hands-on projects that apply theoretical concepts to real-world scenarios, enhancing their understanding of design methodologies and best practices in developing robust software solutions.
Objectives	The objectives for the course are: 1. Understand object-oriented design principles and their application in software development. 2. Learn to create and interpret UML diagrams for modeling software systems. 3. Identify and implement common design patterns to enhance code maintainability. 4. Explore the phases of the software development lifecycle, from requirements to deployment. 5. Engage in hands-on projects to apply theoretical concepts in real-world scenarios. 6. Develop analytical and problem-solving skills through practical exercises and case studies.
Core Concepts	The core concepts of the course include object-oriented design principles, UML modeling, and design patterns, alongside the software development lifecycle and Agile methodologies. It also covers quality assurance techniques and the basics of version control for effective collaboration in software projects.

Course Outline

Week	Topic
1	The topic of Introduction to Software Engineering, providing an overview of the field and emphasizing the importance of software process models. The activities will include discussions on real-world software projects and an introduction to the Software Development Life Cycle (SDLC). This week’s study material will be based on Software Engineering: A Practitioner’s Approach by Roger S. Pressman, covering Chapter 1 (pp. 1-31).
2	Explore Process Models and Agile Development, focusing on traditional process models such as Waterfall, Incremental, and Spiral, as well as Agile approaches like Scrum and XP. The activities will include comparing Agile and traditional models, and introducing basic Agile tools. This week’s study material will come from Software Engineering: A Practitioner’s Approach by Roger S. Pressman, Chapters 2 and 3 (pp. 32-81).
3	Cover Requirements Engineering, including requirements elicitation, specification, and analysis, as well as use case modeling. Activities will feature a case study on gathering requirements and creating use case diagrams. The study material will include Software Engineering: A Practitioner’s Approach by Roger S. Pressman, Chapter 5 (pp. 121-169) and Object-Oriented Software Engineering using UML, Patterns, and Java by Bernd Bruegge, Chapter 4 (pp. 115-150).
4	Software Project Management, covering project planning, estimation, scheduling, and risk management. Activities will involve creating a project plan and discussing risk management techniques. This week’s study material will be from Software Engineering: A Practitioner’s Approach by Roger S. Pressman, Chapter 7 (pp. 207-245).

5	Introduce Object-Oriented Concepts and UML, covering object-oriented design principles and UML class and sequence diagrams. Students will engage in hands-on UML diagram creation based on real-world scenarios. The study material will come from Object-Oriented Software Engineering using UML, Patterns, and Java by Bernd Bruegge, Chapters 5 and 6 (pp. 151-210).
6	Cover Software Design Principles, focusing on design concepts like cohesion, coupling, and architectural design. Students will work on a group project to design a software architecture using UML. The study material will be from Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapters 8 and 9 (pp. 246-305).
7	Introduce Design Patterns in Software Engineering, focusing on design patterns such as Singleton and Factory, and their application in software design. Activities will include implementing a basic design pattern in Java. The study material will come from Object-Oriented Software Engineering using UML, Patterns, and Java by Bernd Bruegge, Chapter 7 (pp. 211-250).
8	Midterm Exam
9	Cover User Interface Design, focusing on UI design principles and usability engineering. Students will prototype a simple UI using Java. The study material will be from Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapter 11 (pp. 370-409).
10	Focus on Software Construction and Coding Standards, discussing best practices for software construction, coding guidelines, and refactoring. Activities will include code review exercises and refactoring Java code. The study material will come from Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapter 13 (pp. 438-475).
11	Cover Software Testing Strategies, including unit testing, integration testing, system testing, and Test-Driven Development (TDD). Students will write and execute test cases for Java programs. The study material will include Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapters 14 and 15 (pp. 476-545).
12	Focus on Quality Management in Software Engineering, covering software quality assurance (SQA), reviews, audits, and software metrics. Activities will include a group discussion on quality assurance methods and a code review session. The study material will come from Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapter 16 (pp. 546-575).
13	Address Software Maintenance and Configuration Management, discussing software maintenance models and version control. Students will practice using Git for version control and explore common maintenance tasks. The study material will come from Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapter 21 (pp. 662-689).
14	Cover Advanced Topics: Reengineering and DevOps, focusing on software reengineering, refactoring, and introducing DevOps practices like Continuous Integration/Continuous Deployment (CI/CD). Activities will include a discussion on modern practices and implementing a simple CI/CD pipeline. The study material will include Software Engineering: A Practitioner's Approach by Roger S. Pressman, Chapter 22 (pp. 690-715).
15	Conclude with a Capstone Project and Review. Students will present their final projects, demonstrating the application of design, testing, and project management principles. The week will wrap up with a comprehensive review of key concepts and best practices.
16	Final Exam

Prerequisites	The student must attend the course at a minimum rate of 75%.
Literature	• Software Engineering A practitioner’s approach, Mc-Graw Hill 7th Edition 2010 by Roger S Pressman • Object Oriented Software Engineering using UML Patterns and JAVA. 2004, by Bernd Bruegge, Pearson 3d Edition.
References	• Head First Object-Oriented Analysis and Design. 2006, by Brett D. McLaughlin, Gary Pollice, David West, O'Reilly Media.

Course Outcome

1	Students will recognize the importance and need for software engineering to address modern trends in the software industry.
2	Students will have knowledge of all phases of the software life cycle, including the artifacts produced.
3	Students will be sufficiently capable of analyzing, evaluating, and applying a range of case tools.
4	The semester project will enable students to implement software engineering concepts in a disciplined manner, to compete in the local and international market.

Course Evaluation

In-term Studies	Quantity	Percentage
Midterms	1	20
Quizzes	3	9
Projects	1	30
Term Projects	0	0
Laboratory	0	0
Class Participation	0	0
Total in-term evaluation percent		**59**
Final exam percent		**41**
Total		**100**

ECTS Workload (Based on Student Workload)

Activities	Quantity	Duration (hours)	Total (hours)
Course duration (Including the exam week: 16x Total hours of the course)	16	4	64
Study hours outside the classroom (Preparation, Practice, etc.)	14	2	28
Duties	1	18	18
Midterms	1	15	15
Final Exam	1	25	25
Other	0	0	0
Total Work Load			**150**
Total Work Load / 25 (hours)			**6.00**
ECTS			**6.00**