

**School of Engineering and Natural Sciences / Computer Engineering (English)**

**2023 - 2024 Academic Year**

**ADVANCED SOFTWARE TECHNIQUES**

**Syllabus**

Course Description					
Name	Code	Semester	T+A Hour	Credit	ECTS
ADVANCED SOFTWARE TECHNIQUES	COE4213193	Spring Semester	3+0	3	6
<b>Prerequisites Courses</b>	NESNE TABANLI PROGRAMLAMA				
<b>Recommended Elective Courses</b>					
<b>Language of Instruction</b>	English				
<b>Course Level</b>	First Cycle (Bachelor's Degree)				
<b>Course Type</b>	Elective				
<b>Course Coordinator</b>	Prof.Dr. Selim AKYOKUŞ				
<b>Name of Lecturer(s)</b>	Prof.Dr. Selim AKYOKUŞ				
<b>Assistant(s)</b>					
<b>Aim</b>	The objective of this course is to learn, understand and analyze modern software development, methods, principles, and practice. Topics include requirements analysis and specification, design, implementation, verification, and maintenance of large software systems; advanced software development techniques and large project management approaches; project planning, scheduling, resource management, risks, configuration control, and documentation. During this course, students implement a large software system in teams using Scrum methodology.				
<b>Course Content</b>	This course contains; Introduction to Software Methodologies and Engineering, Modeling with UML, Software construction, Software Process & Plan-Driven Methods, Agile Methods, Requirements Engineering, Elicitation and Specification, Software Architecture, Design Patterns and Architectural Patterns, Midterm, Model / View / Controller and Web Application Architectures, REST and Web Services, Software Modularity and System Decomposition, Software Usability, Personas, Wireframes, Software Verification and Validation, Software Testing Strategies, Continuous Integration, Software Security and Maintenance.				
<b>Course Learning Outcomes</b>			<b>Teaching Methods</b>	<b>Assessment Methods</b>	
Learn how to plan, design, and build a complex software system that involves various aspects of modern software engineering.					
Gain proficiency in using modern tools and tackling challenging technical domains.					
Experience working on a large software project as part of a team that collaborates with other teams and customer representatives.					
<b>Teaching Methods</b>					
<b>Assessment Methods</b>					
<b>Lecture Schedule</b>					
<b>Sequence</b>	<b>Topics</b>	<b>Preliminary Preparation</b>			
1	Introduction to Software Methodologies and Engineering				
2	Modeling with UML				
3	Software construction, Software Process & Plan-Driven Methods				
4	Agile Methods				
5	Requirements Engineering, Elicitation and Specification				
6	Software Architecture, Design Patterns and Architectural Patterns				
7	Midterm				
8	Model / View / Controller and Web Application Architectures				
9	REST and Web Services				
10	Software Modularity and System Decomposition				
11	Software Usability, Personas, Wireframes				
12	Software Verification and Validation				
13	Software Testing Strategies, Continuous Integration				
14	Software Security and Maintenance				
<b>Evaluation Methods</b>		<b>Weight(%)</b>			
Midterm Exam		30			
General Exam		70			

Resources
- Clean Code, Robert C. Martin, PHI - Design Patterns, Erich Gamma, Addison Wesley - Patterns of Enterprise Application Architecture, Martin Fowler, 1st edition, Addison-Wesley - Enterprise Integration Patterns, Gregor Hohpe, Pearson Addison-Wesley Professional - Code Complete, Steve McConnell, Microsoft Press US Lecture notes that will be delivered during the classes.