The syllabuses of the courses given in the undergraduate program at Eskisehir Osmangazi University Software Engineering Department are listed below.

1	151011215	TURKISH I	2+0	2 ECTS	Compulsory
1	152511001	CALCULUS I	4+0	5 ECTS	Compulsory
1	152511003	PHYSICS I	3+0	3 ECTS	Compulsory
1	152511004	PHYSICS I LAB	0+2	2 ECTS	Compulsory
1	152511008	INTRODUCTION TO SOFTWARE ENGINEERING	2+0	3 ECTS	Compulsory
1	152511005	INTRODUCTION TO PROGRAMMING	3+0	4 ECTS	Compulsory
1	152511006	INTRODUCTION TO PROGRAMMING LAB	0+2	2 ECTS	Compulsory
1	152511007	ADVANCED READING AND WRITING I	3+0	2 ECTS	Compulsory
1	152511009	SOFTWARE PROJECT MANAGEMENT	3+0	4 ECTS	Compulsory
2	151012214	TURKISH II	2+0	2 ECTS	Compulsory
2	152512001	CALCULUS II	4+0	5 ECTS	Compulsory
2	152512004	PHYSICS II	3+0	3 ECTS	Compulsory
2	152512005	PHYSICS II LAB	0+2	2 ECTS	Compulsory
2	152512003	LINEAR ALGEBRA	3+0	3 ECTS	Compulsory
2	152512006	COMPUTER PROGRAMMING	3+0	4 ECTS	Compulsory
2	152512007	COMPUTER PROGRAMMING LAB	0+2	2 ECTS	Compulsory
2	152512008	ADVANCED READING AND WRITING II	3+0	2 ECTS	Compulsory
2	152512009	DISCRETE COMPUTATIONAL STRUCTURES	3+0	4 ECTS	Compulsory
3	152513001	SOFTWARE REQUIREMENTS ENGINEERING	3+0	4 ECTS	Compulsory
3	152513002	OBJECT ORIENTED PROGRAMMING I	3+0	6 ECTS	Compulsory
3	152513003	OBJECT ORIENTED PROGRAMMING I LAB	0+2	2 ECTS	Compulsory
3	152513004	DATA STRUCTURES	2+0	3 ECTS	Compulsory
3	152513005	DATA STRUCTURES LAB	0+2	2 ECTS	Compulsory
3	151011208	ATATURK'S PRINCIPLES AND HISTORY OF TURKISH REVOLUTION I	2+0	2 ECTS	Compulsory
3	152513006	OCCUPATIONAL HEALTH AND SAFETY I	2+0	2 ECTS	Compulsory
3	152513007	NUMERICAL METHODS	3+0	4 ECTS	Compulsory
3	152513008	DIFFERENTIAL EQUATIONS	3+0	4 ECTS	Compulsory
4	152514001	SOFTWARE TESTING	3+0	5 ECTS	Compulsory
4	152514002	OBJECT ORIENTED PROGRAMMING II	3+0	6 ECTS	Compulsory
4	152514003	OBJECT ORIENTED PROGRAMMING II LAB	0+2	2 ECTS	Compulsory
4	152514004	ALGORITHMS AND COMPLEXITY	3+0	5 ECTS	Compulsory
4	152514005	THEORY OF COMPUTATION	3+0	5 ECTS	Compulsory
4	151012209	ATATURK'S PRINCIPLES AND HISTORY OF TURKISH REVOLUTION II	2+0	2 ECTS	Compulsory
4	152514006	OCCUPATIONAL HEALTH AND SAFETY II	2+0	2 ECTS	Compulsory
4	152514007	PROBABILITY	3+0	4 ECTS	Compulsory

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
TURKISH I	151011215

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	2	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
			2	

Course Language	Course Level	Course Type
Turkish	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Provide information about the historical development and status of Turkish; foster correct and effective use of the language in speaking and writing; emphasize its importance as a national language; and introduce principles of composition writing.
Short Course Content	Characteristics of language, world languages, the place of Turkish among them, the history and development of Turkish, language policies, pronunciation, grammar, spelling and punctuation, and composition writing.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the historical development of Turkish, its place among world languages, and the relationship between language and culture.	11	1	A
2	Apply basic Turkish grammar, spelling, and punctuation rules correctly.	9	1	A
3	Write clear and coherent texts in Turkish following the principles of effective expression and structured composition.	9	1	A
4				
5				
6				
7				
8				
9				
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Ergin, M. (1997). Üniversiteler İçin Türk Dili. İstanbul: Bayrak Yayınları
Supporting References	Kaplan, M. (1993). Kültür ve Dil. İstanbul: Dergâh Yayınları (8. baskı)
Necessary Course Material	-

Course Schedule	
1	Definition and characteristics of language.
2	World languages and the place of Turkish among them in terms of structure and origin.
3	The importance of language for culture and national identity; language policies.
4	Spoken language and its varieties (dialects and regional speech).
5	Written language and its characteristics.
6	Classification of speech sounds.
7	Sound changes and phonological processes.
8	Mid-Term Exam
9	Pronunciation.
10	Grammar and punctuation.
11	Spelling rules.
12	Spelling rules.
13	Written composition exercises.
14	Structured composition writing exercises.
15	Course review.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	5	5
Final Exam	1	1	1
Studying for Final Exam	1	5	5
		Total workload	**54**
		Total workload / 30	**1.8**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	50
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	5
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	2

LECTURER(S)				
Prepared by	Lecturer Meriç AKPINAR			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
CALCULUS I	152511001

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	4	0	5

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
5				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Teaching students the basic concepts, theorems of calculus and provide them the ability to solve mathematical problems.
Short Course Content	Functions. Limits and continuity. Differentiation. Applications of derivatives. Integration. Techniques of integration. Application of integration.

Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1 Solving limit problems.	1a	1, 10	A, B
2 Defining differentiation.	1a	1, 10	A, B
3 Applying derivatives to certain problems.	1a	1, 10	A, B
4 Defining integration.	1a	1, 10	A, B
5 Solving definite integrals.	1a	1, 10	A, B
6 Applying integration to certain problems.	1a	1, 10	A, B
7			
8			
9			
10			

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	J. R. Hass, C. E. Heil, M. D. Weir, P. Bogacki, Thomas' Calculus, Pearson, 15th ed., 2023.
Supporting References	J. Stewart, D. K. Clegg, S. Watson, Calculus, Cengage Learning; 9th ed, 2020.
Necessary Course Material	-

Course Schedule	
1	Functions and their graphs. Shifting and scaling.
2	Trigonometric functions. Exponential functions. Inverse functions. Natural logarithm.
3	Limits. Types of limits.
4	Types of limits. Continuity of a function.
5	Differentiation. Tangents and derivative at a point. Differentiation rules.
6	Derivatives of certain functions. Chain rule. Implicit differentiation.
7	Extreme values of a function. Mean value theorem.
8	Mid-Term Exam
9	Antiderivatives. Integration. Definite integrals and their properties.
10	Areas under the graphs. Average value of a continuous function.
11	Indefinite integrals and substitution method. Areas between curves.
12	Volumes using cross-sections. Volumes by disks for rotation. Solids of revolution.
13	Volumes using cylindrical shells. Arc length. Areas of surfaces of revolution.
14	Techniques of integration. Integration by parts. Trigonometric integrals.
15	Integration of rational functions by partial fractions. Heaviside method.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	4	56
Classroom Studying Time (review, reinforcing, prestudy,….)	14	4	56
Homework			
Quiz Exam	2	1	2
Studying for Quiz Exam	2	5	10
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	16	16
Final Exam	1	2	2
Studying for Final Exam	1	16	16
		Total workload	**160**
		Total workload / 30	**5.33**
		Course ECTS Credit	**5**

Evaluation	
Activity Type	**%**
Mid-term	35
Quiz	20
Final Exam	45
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTUTER(S)				
Prepared by	Prof. Dr. Dursun ESER			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
PHYSICS I	152511003

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	3	0	3

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
3				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Teaching Newtonian mechanics as one of the fundamental fields in physics; hence to provide students the proper problem solving strategies in related topics and helping to establish a foundation for their succeeding coursework.
Short Course Content	Units, vectors, motion along a straight line, motion in two-three dimensions, applications of Newton's laws, work, kinetic energy, potential energy, conservation of energy and momentum, rotational motion of rigid bodies, gravitation, and simple harmonic motion.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Use units of physical quantities correctly and perform unit conversions.	1b	1	A
2	Perform mathematical operations involving vectors.	1a, 1b	1, 10	A
3	Analyze translational motion in one, two, and three dimensions using kinematic relations.	1a, 1b	1	A
4	Solve dynamics problems using Newton's laws of motion.	1a, 1b	1, 2, 5	A
5	Apply the concepts of work, kinetic energy, potential energy, and conservation of energy to physics problems.	1a, 1b	1, 2, 5, 7	A
6	Solve problems involving impulse, linear momentum, collisions, and conservation of momentum.	1a, 1b	1, 2, 5, 8	A
7	Solve problems involving rotational motion and rolling without slipping.	1a, 1b	1, 2	A
8	Solve physics problems involving the law of gravitation.	1a, 1b	1, 2	A
9	Develop mathematical models of physical problems and interpret the physical meaning of the results.	1a, 1b	1, 2, 5, 6, 7, 8, 10	A
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Young H.D., Freedman R.A. University Physics with Modern Physics, 14th Edition (2015), Pearson.
Supporting References	Halliday, D., Resnick, R., and Walker, J. (2008). Fundamentals of Physics (8th Edition). John Wiley & Sons, Inc. Serway, R.A., Beichner, R.J., Physics For Scientists and Engineers with Modern Physics (2007), Harcourt College Publishers.
Necessary Course Material	-

Course Schedule	
1	Units, unit conversions, and physical quantities.
2	Vectors.
3	Motion along a straight line.
4	Motion in two and three dimensions.
5	Newton's laws.
6	Application of Newton's laws.
7	Work and kinetic energy.
8	Mid-Term Exam
9	Potential energy and conservation of energy.
10	Rotational motion of rigid bodies.
11	Rotational dynamics.
12	Rotational dynamics.
13	Equilibrium and elasticity.
14	Gravitation.
15	Simple harmonic motion
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	15	15
Final Exam	1	2	2
Studying for Final Exam	1	15	15
		Total workload	**90**
		Total workload / 30	**3**
		Course ECTS Credit	**3**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	3
		b. Knowledge of natural sciences.	5
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Prof. Dr. Tamer AKAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
PHYSICS I LAB	152511004

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	0	2	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
2				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Teaching through conducting experiments on the topics related to the content of Physics I; hence, enhancing the students' perception and understanding on the important concepts and fundamental laws of the Newtonian Mechanics.
Short Course Content	Numerical analysis and error calculation, measurement, specification of the components of forces, Newton's laws of motion, projectile motion, conservation of energy, conservation of momentum, moment of inertia, springs, and viscosity experiments.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Systematically record experimental observations and analyze basic physical relationships.	1b, 5	3, 7	E
2	Use basic laboratory measuring instruments correctly and safely to perform measurements.	1b, 4	3, 6	A, E
3	Calculate numerical errors and uncertainties in experimental data.	1a, 5	3, 10	E
4	Present experimental data using appropriate graphs and interpret the results.	1a, 5, 9	3, 15	E
5	Compare experimental results with mathematical and physical models and interpret deviations.	1a, 1b, 5	3, 8, 10	A, E
6	Prepare laboratory reports describing the experimental setup, methods, findings, and conclusions.	5, 9	11, 15	E
7	Collaborate in laboratory work by sharing responsibilities within a group.	8	3, 12	E
8	Conduct basic experiments involving mechanics, motion, energy, momentum, springs, and viscosity.	1b, 4, 5	3, 6	A, E
9	Explain relationships between physical laws and engineering applications using experimental observations.	1b	2, 7	A, E
10	Assemble and use simple mechanics experimental setups for their intended purposes.	4, 5	3, 6, 10	E

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Physics I Experiments. Eskişehir: Eskişehir Osmangazi Üniversitesi Yayınları, Sertaç Eroğlu, Murat Kellegöz, Gökhan Kılıç, Halil Yasin Adıyaman.
Supporting References	Fundamentals of Physics, Halliday Resnick, John Wiley and Sons Inc. 1988. Giancoli, D.C. (2004). Physics: Principles with Applications (6th Edition). Pearson Education Inc.
Necessary Course Material	-

Course Schedule	
1	Introduction to laboratory, and formation of lab groups.
2	Numerical analysis and error calculation.
3	Measurement experiment.
4	Motion with constant acceleration experiment.
5	Conservation of linear momentum experiment.
6	Projectile motion experiment.
7	Projectile motion experiment.
8	Mid-Term Exam
9	Free fall experiment.
10	Simple pendulum and conservation of energy experiment.
11	Motion on a frictional inclined plane experiment.
12	Springs experiment.
13	Moment of inertia experiment.
14	Viscosity experiment.
15	Make up experiment, general review and preparation for final exam.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)			
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	10	2	20
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam	1	2	2
Studying for Final Exam	1	10	10
		Total workload	**60**
		Total workload / 30	**2**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Report	50
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	3
		b. Knowledge of natural sciences.	5
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	3
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	4
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	4
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	3
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Prof. Dr. Tamer AKAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
INTRODUCTION TO SOFTWARE ENGINEERING	152511008

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	2	0	3

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	3			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	The main objective of this course is to provide fundamental knowledge about software engineering principles, software development life cycle, software processes, requirements engineering, and software design. It also introduces ethical considerations and professional responsibilities in software engineering.
Short Course Content	Introduction to software engineering, software development processes, software design principles, requirements engineering, testing methodologies, project management, software maintenance, and professional ethics.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the fundamental principles of software engineering and the stages of the software development life cycle.	1c, 2, 10	1, 5, 10, 11	A
2	Compare software development process models and agile methodologies.	1c, 10	1, 5, 10, 11	A
3	Explain basic software design principles and modeling techniques.	1c, 1e, 2, 3	1, 5, 10, 11	A
4	Explain the main stages of requirements engineering and the purpose of software documentation.	2, 3, 9	1, 5, 10, 11	A
5	Distinguish between basic software testing methods and quality assurance approaches.	4, 5	1, 5, 10, 11	A
6	Explain the importance of teamwork and effective communication in software projects.	8, 9, 10	1, 5, 10, 11	A
7	Explain ethical principles, professional responsibilities, and the societal impacts of software engineering applications.	6, 7	1, 5, 10, 11	A
8	Summarize current developments and emerging technological trends in software engineering.	4, 11	1, 5, 10, 11	A
9				
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Ian Sommerville, "Software Engineering," 10th Edition, Pearson, 2015, ISBN: 978-0133943030.
Supporting References	Roger S. Pressman, "Software Engineering: A Practitioner's Approach," 8th Edition, McGraw-Hill, 2014.
Necessary Course Material	Computer, projector.

Course Schedule	
1	Introduction to software engineering and its importance.
2	Software development life cycle (SDLC) and models.
3	Requirements engineering and software documentation.
4	Software design principles and software architecture.
5	Software testing techniques and quality assurance.
6	Software maintenance and evolution.
7	Agile methodologies and software project management.
8	Mid-Term Exam
9	Object-oriented analysis and design.
10	Software development tools and version control.
11	Human-computer interaction and usability engineering.
12	Risk management in software projects.
13	Professional ethics and responsibilities in software engineering.
14	Future trends in software engineering.
15	General review and preparation for final exam.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	20	20
Final Exam	1	1	1
Studying for Final Exam	1	30	30
		Total workload	**94**
		Total workload / 30	**3.13**
		Course ECTS Credit	**3**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	4
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	2
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	2
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	2
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	2
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	4
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	2
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	3
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	2

LECTURER(S)				
Prepared by	Assoc. Prof. Dr. Canan EREN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
INTRODUCTION TO PROGRAMMING	152511005

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	4			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	This course aims to provide students with competencies regarding program development and algorithm design.
Short Course Content	Fundamentals of algorithms and programming, creating flow diagrams, loops, conditions, functions and variable types.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	To be able to use programming tools effectively.	1d, 4	1,5,10,11	A
2	Creating a flow diagram of a problem.	1e, 2, 3	1,5,10,11	A
3	Knowing and using the syntax rules of the C programming language.	1c, 1d, 4	1,5,10,11	A
4	To be able to convert the algorithm or flow diagram of a problem into C code.	1d, 1e, 2, 3	1,5,10,11	A
5	Creating simple projects and writing programs in C-programming language.	1c, 1d, 1e, 2, 3, 4	1,5,10,11	A
6	Using decision structures.	1d, 2	1,5,10,11	A
7	Using loops.	1d, 2	1,5,10,11	A
8	Working with functions (subroutines).	1d, 1e, 2, 3	1,5,10,11	A
9	Performing array operations.	1d, 2	1,5,10,11	A
10	Ability to perform operations with multidimensional arrays.	1d, 2	1,5,10,11	A

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	H.M. Deitel, P.J. Deitel, I. Pohl, C: How To Program, 2013.
Supporting References	Lecture notes, slides and resources on the internet.
Necessary Course Material	Accessible computers for each student, Visual Studio 2013 or later, projection

Course Schedule	
1	Introduction.
2	C Programming and the basics of the C environment.
3	Structural programming and creating algorithm.
4	Control structures (If, If-Else, Switch-Case).
5	Control structures (Break-continue, assignment operators, increment and decrement operators.)
6	Program control and loops (For, While, Do-While) and logical operators.
7	Program control and loops (For, While, Do-While) and logical operators.
8	Mid-Term Exam
9	Functions.
10	Recursive functions.
11	Arrays
12	Arrays.
13	Usage of arrays and functions.
14	Multi dimensional arrays.
15	Multi dimensional arrays.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	2	28
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	20	20
Final Exam	1	2	2
Studying for Final Exam	1	30	30
		Total workload	**124**
		Total workload / 30	**4.13**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	3
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	4
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İbrahim SAVRAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
INTRODUCTION TO PROGRAMMING LAB	152511006

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	0	2	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	2			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To teach high-level programming environments, Introduction to C language, structure programming, variables, expressions and initializations, functions and calling of functions, prototypes and header files, arrays and pointers.
Short Course Content	This course is practice of writing programs with using C programming language. Students learn how to design a program structure and simple applications which contain loops, decisions and functions.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Develop basic programs in C.	1d, 1e, 2, 3, 4	1,5,6,11,14,15	B, J
2	Use basic programming concepts and terminology accurately when communicating with software developers.	1c, 9	1,5,6,11,14,15	B, J
3	Read simple C programs and explain how they work.	1c, 1d, 2, 4	1,5,6,11,14,15	B, J
4	Write, test and debug C programs for simple problems.	1d, 1e, 2, 4, 5	1,5,6,11,14,15	B, J
5	Explain the basic stages of software development and the role of programming in this process.	1c, 1e, 3	1,5,6,11,14,15	B, J
6	Compare the common features and differences of modern programming languages.	1c, 4	1,5,6,11,14,15	B, J
7	Identify how programming can be used to solve a given problem.	1d, 1e, 2	1,5,6,11,14,15	B, J
8				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	H.M. Deitel, P.J. Deitel, I. Pohl, C: How To Program, 2013.
Supporting References	Lecture notes, slides and resources on the internet.
Necessary Course Material	Accessible computers for each student, Visual Studio 2013 or later, projection

Course Schedule	
1	Introduction of the laboratory.
2	Definition and application of C programming and the basics of the C environment.
3	Application of structural programming and creating algorithm.
4	Application of control structures (If, If-Else, Switch-Case).
5	Application of control structures (Break - continue, assignment operators etc.).
6	Application of program control and loops.
7	Application of program control and loops.
8	Mid-Term Exam
9	Application of functions.
10	Application of recursive functions.
11	Application of arrays.
12	Application of arrays.
13	Application of using arrays and functions.
14	Application of multi dimensional arrays.
15	Application of multi dimensional arrays.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	1	5	5
Project (Preparation and presentation time included)	1	15	15
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam			
Studying for Final Exam			
		Total workload	**62**
		Total workload / 30	**2.07**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Quiz	70
Project Observation	30
Final Exam	
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	3
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	2
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İbrahim SAVRAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
ADVANCED READING AND WRITING I	152511007

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	3	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
			2	

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Introduction to the writing process Teaching paragraph and essay writing Practicing 5-paragraph essay writing Practicing various essay organizations
Short Course Content	Writing process, brainstorming, planning, drafting, revising, editing, paragraph writing, 5-paragraph essay, introduction, body and conclusion paragraphs, process essay, classification essay, cause-effect essay, comparison-contrast essay.

Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1 Ability to write paragraphs	9	1, 2, 6	A, B
2 Ability to write five-paragraph or longer essays	9	1, 2, 6	A, B
3 Development of written communication skills	9	1, 2, 6	A, B
4 Introduction to professional composition writing	9	1, 2, 6	A, B
5			
6			
7			
8			
9			
10			

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Alice Oshima & Ann Hogue, "Longman Academic Writing Series 3," Pearson, 2017.
Supporting References	Stephen Bailey, "Academic Writing: A Handbook for International Students," Routledge, 2018.
Necessary Course Material	-

Course Schedule	
1	Introduction to the course, purpose and expectations
2	The writing process
3	Subject, purpose and audience
4	Brain storming
5	Developing paragraphs
6	Unity and coherence in paragraphs
7	Five-paragraph essay
8	Mid-Term Exam
9	Introduction and conclusion paragraphs
10	Process essay
11	Process essay practice
12	Classification essay
13	Cause/Effect essay
14	Cause/Effect essay practice
15	Comparison/contrast essay
16,17	Final Exam

Calculation of Course Workload

Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam	1	1	1
Studying for Quiz Exam	1	2	2
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	2	2
Final Exam	1	2	2
Studying for Final Exam	1	4	4
		Total workload	**69**
		Total workload / 30	**2.3**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	35
Quiz	20
Final Exam	45
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	5
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Lecturer Seda ÜNER			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
SOFTWARE PROJECT MANAGEMENT	152511009

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
1	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	4			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	The main objective of this course is to introduce students to fundamental principles of software project management, including planning, scheduling, risk management, cost estimation, and quality assurance. The course aims to provide an understanding of best practices in managing software projects effectively.
Short Course Content	Introduction to software project management, software project planning, scheduling techniques, risk management, cost estimation, software quality assurance, agile project management, and team dynamics.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the fundamental concepts and processes of software project management and their relationship to the software development life cycle.	1c, 10	1, 5, 10, 11	A
2	Compare traditional and agile software project planning approaches.	10	1, 5, 10, 11	A
3	Prepare a sample software project plan using basic scheduling and resource allocation techniques.	1e, 2, 10	1, 5, 10, 11	A
4	Identify key risks in software projects and explain appropriate risk management strategies.	2, 10	1, 5, 10, 11	A
5	Explain basic cost estimation and budgeting techniques used in software projects.	10	1, 5, 10, 11	A
6	Explain software quality assurance, process improvement, and performance measurement approaches.	1c, 4, 10	1, 5, 10, 11	A
7	Explain principles of team management, leadership, and effective communication in software projects.	8, 9, 10	1, 5, 10, 11	A
8	Explain professional ethics and responsibilities in software project management.	7, 10	1, 5, 10, 11	A
9	Explain the roles of change management and version control in projects and summarize emerging trends in software project management.	4, 10, 11	1, 5, 10, 11	A
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Bob Hughes, Mike Cotterell, "Software Project Management," 6th Edition, McGraw-Hill, 2016, ISBN: 978-0077166795.
Supporting References	Ian Sommerville, "Software Engineering," 10th Edition, Pearson, 2015.
Necessary Course Material	Computer, projector.

Course Schedule	
1	Introduction to software project management.
2	Software development life cycle and project planning.
3	Project scheduling and resource allocation.
4	Risk management in software projects.
5	Cost estimation techniques and budgeting.
6	Software quality assurance and process improvement.
7	Agile methodologies in project management.
8	Mid-Term Exam
9	Project team management and leadership.
10	Communication and collaboration in software projects.
11	Change management and version control.
12	Software metrics and performance evaluation.
13	Ethical considerations in project management.
14	Future trends in software project management.
15	General review and preparation for final exam.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	2	28
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	20	20
Final Exam	1	2	2
Studying for Final Exam	1	30	30
		Total workload	**124**
		Total workload / 30	**4.13**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	2
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	3
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	2
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	2
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	2
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	5
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	2

LECTURER(S)				
Prepared by	Lecturer Dr. Zeynep ŞAHİN TİMAR			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
TURKISH II	151012214

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	2	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
			2	

Course Language	Course Level	Course Type
Turkish	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To develop an understanding of Turkish word classes and sentence structures; enable students to analyze written and oral forms of expression and produce well-organized, accurate, clear, and coherent texts in Turkish.
Short Course Content	Word structure and word groups; nouns, adjectives, pronouns, adverbs, prepositions, conjunctions, interjections, and verbs; sentences and sentence elements; types of written composition; prepared and impromptu speech; text analysis; structured composition writing.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Classify Turkish parts of speech and word groups using relevant examples.	9	1, 5	A
2	Identify sentence elements and analyze sentence structures in Turkish.	9	1, 5	A
3	Distinguish types of written composition and analyze sample texts in terms of language and expression.	9	1, 2, 5	A
4	Compare the characteristics of prepared and impromptu speech and select appropriate speaking strategies.	9	1, 2, 5, 6	A
5	Write well-organized, clear, and coherent compositions in Turkish by applying grammar and writing conventions.	9	1, 5, 6	A
6				
7				
8				
9				
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Ergin, M. (1997). Üniversiteler İçin Türk Dili. İstanbul: Bayrak Yayınları
Supporting References	Kaplan, M. (1993). Kültür ve Dil. İstanbul: Dergâh Yayınları (8. baskı)
Necessary Course Material	-

Course Schedule	
1	Word classes, word groups
2	Noun
3	Adjective
4	Pronoun
5	Adverb
6	Preposition, Conjunction, Exclamation
7	Verb
8	Mid-Term Exam
9	Sentence, Elements of Sentence
10	Types of Written Composition
11	Types of Written Composition
12	Prepared Speech Practice, Unprepared Speech Practice
13	Text Analysis Studies
14	Planned Composition Writing
15	Course Review
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	5	5
Final Exam	1	1	1
Studying for Final Exam	1	5	5
		Total workload	**54**
		Total workload / 30	**1.8**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	50
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	5
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Lecturer Meriç AKPINAR			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
CALCULUS II	152512001

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	4	0	5

Course Category (Credit)				
Basic Sciences	**Engineering Sciences**	**Design**	**General Education**	**Social**
5				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To teach fundamental concepts and theorems of multivariable calculus and vector analysis and to develop students' ability to solve related mathematical problems.
Short Course Content	Sequences and series, convergence tests, coordinate systems, vectors, limits and continuity of multivariable functions, partial and directional derivatives, multiple integrals, vector fields, line and surface integrals, divergence, curl and integral theorems.

Learning Outcomes of the Course		Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Analyze the convergence of sequences and series using appropriate tests.	1a	1, 10	A, B
2	Describe different coordinate systems and perform transformations between them.	1a	1, 10	A, B
3	Apply basic operations to vectors and vector fields.	1a	1, 10	A, B
4	Calculate partial and directional derivatives of multivariable functions.	1a	1, 10	A, B
5	Use partial derivatives to solve multivariable problems.	1a	1, 10	A, B
6	Set up and evaluate double and triple integrals.	1a	1, 10	A, B
7	Apply multiple integrals to area and volume problems.	1a	1, 10	A, B
8	Calculate divergence and curl of vector fields.	1a	1, 10	A, B
9	Evaluate line and surface integrals of vector fields.	1a	1, 10	A, B
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	J. R. Hass, C. E. Heil, M. D. Weir, P. Bogacki, Thomas' Calculus, Pearson, 15th ed., 2023.
Supporting References	J. Stewart, D. K. Clegg, S. Watson, Calculus, Cengage Learning; 9th ed, 2020.
Necessary Course Material	-

Course Schedule	
1	Sequences and infinite series. Convergence. Comparison tests. Ratio and root tests.
2	Alternating series. Absolute convergence. Power series. Taylor and Maclaurin series.
3	Parametric equations. Polar coordinates. Graphing in polar coordinates. Areas and lengths in polar coordinates.
4	Three dimensional coordinate systems. Vectors. Basic operations on vectors.
5	Vectors and analytical geometry.
6	Limits and continuity of multivariable functions.
7	Partial derivatives. Chain rule.
8	Mid-Term Exam
9	Directional derivatives. Gradients.
10	Extreme values and saddle points.
11	Double integrals. Finding limits of integration. Areas by double integration. Double integrals in polar form.
12	Triple integrals. Finding limits of integration. Triple integrals in cylindrical and spherical coordinates.
13	Integration in vector fields. Line integrals. Fundamental theorem for line integrals.
14	Divergence and curl. Parametric surfaces. Surface integrals.
15	Stokes theorem. Divergence theorem.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	4	56
Classroom Studying Time (review, reinforcing, prestudy,….)	14	4	56
Homework			
Quiz Exam	2	1	2
Studying for Quiz Exam	2	5	10
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	16	16
Final Exam	1	2	2
Studying for Final Exam	1	16	16
		Total workload	**160**
		Total workload / 30	**5.33**
		Course ECTS Credit	**5**

Evaluation	
Activity Type	**%**
Mid-term	35
Quiz	20
Final Exam	45
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

INSTRUCTOR(S)				
Prepared by	Prof. Dr. Dursun ESER			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
PHYSICS II	152512004

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	3	0	3

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
3				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To teach students the fundamental laws and related problem solving approaches of electricity and magnetism; hence, to provide them the conceptual materials in order to comprehend the basic scientific foundations in associated novel academic research and emerging technological advancements.
Short Course Content	Electric charges, Coulomb's Law, electric field, Gauss's Law, electric potential, capacitance and dielectric materials, resistance and electric current, magnetic field, magnetic field sources, electromagnetic induction, Maxwell's equations.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the fundamental concepts and physical laws of electricity and magnetism.	1b	1, 2, 5	A
2	Model and solve physics and engineering problems involving electricity and magnetism using mathematical relations.	1a, 1b	1, 2, 5, 10	A
3	Use the concepts of electric field, electric potential, and capacitance to evaluate the effects of dielectric materials on capacitors.	1a, 1b	1, 2, 5, 6	A
4	Analyze simple direct-current circuits using relationships among current, resistance, and electromotive force.	1a, 1b	1, 2, 5, 6	A
5	Analyze direct-current circuits using Kirchhoff's laws based on conservation of charge and energy.	1a, 1b	1, 2, 4	A
6	Explain the relationships among moving electric charges, magnetic fields, and magnetic forces.	1b	1, 5	A
7	Explain electromagnetic induction and the relationship between changing electric and magnetic fields.	1b	1, 5	A
8	Explain the basic mechanism of electricity generation from renewable energy sources using electromagnetic induction.	1b	2, 4, 7	A
9	Give examples of applications of electricity and magnetism in electronics and modern technologies based on physical principles.	1b	1, 4, 7	A
10	Interpret mathematical solutions to electricity and magnetism problems in their physical context.	1a, 1b	1, 2, 11, 13	A

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Young H.D., Freedman R.A. University Physics with Modern Physics, 14th Edition (2015), Pearson.
Supporting References	Halliday, D., Resnick, R., and Walker, J. (2008). Fundamentals of Physics (8th Edition). John Wiley & Sons, Inc. Serway, R.A., Beichner, R.J., Physics For Scientists and Engineers with Modern Physics (2007), Harcourt College Publishers.
Necessary Course Material	-

Course Schedule	
1	Atomic structure and electrical charges.
2	Coulomb's Law.
3	Electric field.
4	Gauss's Law.
5	Electric potential.
6	Capacitances.
7	Dielectric materials.
8	Mid-Term Exam
9	Electric current, resistance, and electromotor force.
10	Direct-current circuits.
11	Kirchhoff's laws.
12	Magnetic field and magnetic force.
13	Sources of the magnetic field.
14	Electromagnetic induction.
15	Maxwell's equations and electromagnetic waves.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	15	15
Final Exam	1	2	2
Studying for Final Exam	1	15	15
		Total workload	**90**
		Total workload / 30	**3**
		Course ECTS Credit	**3**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	3
		b. Knowledge of natural sciences.	5
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Prof. Dr. Tamer AKAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
PHYSICS II LAB	152512005

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	0	2	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
2				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Teaching through conducting experiments on the topics related to the content of Physics II; hence, enhancing the students' perception and understanding on the important concepts and fundamental laws of electricity and magnetism.
Short Course Content	Basic electrical measurements; electrolysis; Ohm's law; the Wheatstone bridge; equipotential surfaces and electric field lines; magnetic force and the Biot–Savart law; transformers; experiments on wire resonance, resonance tubes, and standing waves.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Systematically record observations and analyze basic physical relationships in electricity and magnetism experiments.	1b, 5	3, 7	A, E
2	Perform electrical measurements using voltmeters, ammeters, and other appropriate laboratory instruments.	1b, 4	3, 6	A, E
3	Calculate numerical errors and uncertainties in experimental data.	1a, 5	3, 10	E
4	Present and interpret data obtained from electrical and magnetic experiments using appropriate graphs.	1a, 5, 9	3, 15	E
5	Compare measurements with mathematical and physical models and interpret deviations.	1a, 1b, 5	3, 8, 10	A, E
6	Prepare laboratory reports describing experimental setups, methods, data, and conclusions.	5, 9	11, 15	E
7	Collaborate in laboratory work by sharing responsibilities within a group.	8	3, 12	E
8	Conduct basic experiments involving Ohm's law, Wheatstone bridges, electric fields, and magnetic forces.	1b, 4, 5	3, 6	A, E
9	Explain how electromagnetic principles relate to physical applications such as transformers and resonance experiments.	1b	2, 7	A, E
10	Assemble basic electricity and magnetism experimental setups, check connections, and perform measurements.	4, 5	3, 6, 10	A, E

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Physics II Experiments Laboratory Book, Eskişehir: Eskişehir Osmangazi Üniversitesi Yayınları, Sertaç Eroğlu, Murat Kellegöz, Gökhan Kılıç, Halil Yasin Adıyaman
Supporting References	Halliday, D., Resnick, R., and Walker, J. (2008). Fundamentals of Physics (8th Edition). John Wiley & Sons, Inc. Serway, R.A., Beichner, R.J., Physics For Scientists and Engineers with Modern Physics (2007), Harcourt College Publishers
Necessary Course Material	-

Course Schedule	
1	Introduction to laboratory, and formation of lab groups.
2	Electrolysis experiment.
3	Basic measurements and Ohm's Law Experiment.
4	Ohm's Law experiment
5	Wheatstone Bridge experiment
6	Equipotential and electric field lines experiment
7	Equipotential and electric field lines experiment
8	Mid-Term Exam
9	Magnetic force experiment
10	Biot-Savart Law Experiment
11	Transformers experiment
12	Resonance in Wire experiment
13	Resonance tube and standing waves experiment
14	Resonance tube and standing waves experiment
15	Make up experiment, general review and preparation for final exam.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)			
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	10	2	20
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam	1	2	2
Studying for Final Exam	1	10	10
		Total workload	**60**
		Total workload / 30	**2**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Report	50
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	3
		b. Knowledge of natural sciences.	5
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	3
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	4
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	4
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	3
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Prof. Dr. Tamer AKAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
LINEAR ALGEBRA	152512003

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	3	0	3

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
3				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	The objective of the course is to provide students with a comprehensive understanding of the fundamental concepts and techniques of linear algebra. Students will learn to solve systems of linear equations, understand vector spaces, linear transformations, and matrices, and apply these concepts to various problems in science, engineering, and other fields. The course aims to develop students' analytical and problem-solving skills, enabling them to model and solve real-world problems using linear algebraic methods.
Short Course Content	Systems of linear equations, matrices and matrix operations, Gaussian and Gauss-Jordan elimination, determinants and their applications, vectors, vector spaces and subspaces, basis and dimension, inner product spaces, orthogonality, linear transformations, eigenvalues and eigenvectors.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Perform matrix addition, multiplication and inversion and relate these operations to systems of linear equations.	1a	1,2,5,10,11	A
2	Solve systems of linear equations using Gaussian and Gauss-Jordan elimination and analyze the results.	1a, 2	1,2,5,10,11	A
3	Calculate determinants and use them in linear systems and geometric transformations.	1a, 1e	1,2,5,10,11	A
4	Define vector spaces and subspaces and apply the concepts of span, basis and dimension.	1a	1,2,5,10,11	A
5	Apply inner products, orthogonality, orthonormal bases and projections to relevant problems.	1a, 1e	1,2,5,10,11	A
6	Analyze linear transformations, calculate eigenvalues and eigenvectors, and interpret results in applied problems.	1a, 1e, 2	1,2,5,10,11	A

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Bernard Kolman, David Hill. Elementary Linear Algebra with Applications, Pearson Education Press, 9th edition, 2012.
Supporting References	David C. Lay, Steven R. Lay, Judi J. McDonald. Linear Algebra and Its Applications. 2022, Sixth Edition, Pearson.
Necessary Course Material	

Course Schedule	
1	Linearity and Linear Equations
2	Matrices: Properties, Operations, Inverse Matrices
3	Solving Linear Equations
4	Determinants
5	Elementary Matrices, Applications of Determinants
6	Vectors
7	Vector Spaces: Definitions, Sample Vector Spaces
8	Midterm
9	Vector Spaces: Subspaces
10	Vector Spaces: Span, Basis, Dimension
11	Inner Product Spaces: Definitions, Axioms, Sample Problems
12	Inner Product Spaces: Orthogonality, Orthonormality, Orthonormal Basis
13	Homogeneous Systems, Coordinates and Isomorphisms
14	Linear Transformation, Change of Bases
15	Eigenvalues and Eigenvectors
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	20	20
Final Exam	1	1	1
Studying for Final Exam	1	20	20
		Total workload	**98**
		Total workload / 30	**3.27**
		Course ECTS Credit	**3**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	2
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	2
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

INSTRUCTOR(S)				
Prepared by	Asst. Prof. Dr. Aysun TOK ONARCAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
COMPUTER PROGRAMMING	152512006

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	4			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To develop students' ability to write C++ programs using arrays, pointers, dynamic memory management, structures, file operations, and object-oriented programming.
Short Course Content	Structured programming in C++; one-dimensional and multidimensional arrays, pointers and pointer arithmetic, character strings, dynamic memory management, structures and unions, file operations, and an introduction to object-oriented programming.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain advanced software development concepts using C++.	1c, 1d, 1e	1, 5, 10, 11	A
2	Use C++ programming concepts and technical terminology accurately when communicating with software developers.	1c, 9	1, 5, 10, 11	A
3	Analyze the behavior of advanced programs written in C++.	1d, 2, 4	1, 5, 10, 11	A
4	Develop C++ programs using one-dimensional and multidimensional arrays.	1d, 1e, 2	1, 5, 10, 11	A
5	Apply pointers and pointer arithmetic in C++ programs.	1d, 1e, 2, 4	1, 5, 10, 11	A
6	Use character strings and related library functions.	1d, 2	1, 5, 10, 11	A
7	Perform dynamic memory allocation and deallocation in C++.	1d, 1e, 2, 4	1, 5, 10, 11	A
8	Define composite data types using structures and unions.	1d, 1e, 3	1, 5, 10, 11	A
9	Perform file reading and writing operations in C++ programs.	1d, 1e, 4	1, 5, 10, 11	A
10	Design basic C++ applications using object-oriented programming principles.	1c, 1d, 1e, 2, 3, 4	1, 5, 10, 11	A

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	H. M. Deitel, P. J. Deitel, I. Pohl, C++: How To Program, 2016.
Supporting References	Lecture notes and slides.
Necessary Course Material	Accessible computers for each student, Visual Studio 2013 or later, projection

Course Schedule	
1	Review of introductory programming concepts.
2	Arrays and Matrices.
3	Pointers.
4	Pointer expressions and pointer arithmetic.
5	Pointer-array.
6	Characters and strings.
7	Character handling library and functions.
8	Mid-Term Exam
9	Dynamic memory allocation.
10	Structures and type definitions.
11	Structures and type definitions.
12	Unions.
13	File Processing.
14	Object-Oriented Programming.
15	Object-Oriented Programming.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	2	28
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	20	20
Final Exam	1	2	2
Studying for Final Exam	1	30	30
		Total workload	**124**
		Total workload / 30	**4.13**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	3
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	4
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İbrahim SAVRAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
COMPUTER PROGRAMMING LAB	152512007

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	0	2	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	2			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To develop practical C++ programming skills involving arrays, pointers, dynamic memory management, structures, file operations, and object-oriented programming.
Short Course Content	Advanced C++ programming practice: one-dimensional and multidimensional arrays, pointers and pointer arithmetic, character strings, dynamic memory management, structures and unions, file operations, object-oriented programming, code testing and debugging.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Develop advanced C++ programs in a programming environment.	1c, 1d, 1e, 4	1, 5, 6, 11, 14, 15	B, J
2	Use C++ programming concepts and terminology accurately to explain program functionality.	1c, 9	1, 5, 6, 11, 14, 15	B, J
3	Read advanced C++ code and analyze program behavior.	1d, 2, 4	1, 5, 6, 11, 14, 15	B, J
4	Write, test and debug C++ programs to solve given problems.	1d, 1e, 2, 4, 5	1, 5, 6, 11, 14, 15	B, J
5	Develop C++ applications using one-dimensional and multidimensional arrays.	1d, 1e, 2, 4	1, 5, 6, 11, 14, 15	B, J
6	Apply pointers and pointer arithmetic in C++ programs.	1d, 1e, 2, 4	1, 5, 6, 11, 14, 15	B, J
7	Use character strings and related library functions in programs.	1d, 1e, 4	1, 5, 6, 11, 14, 15	B, J
8	Perform dynamic memory allocation and deallocation in C++.	1d, 1e, 2, 4	1, 5, 6, 11, 14, 15	B, J
9	Design composite data types using structures and unions.	1d, 1e, 3	1, 5, 6, 11, 14, 15	B, J
10	Develop C++ applications that read and write files.	1d, 1e, 4	1, 5, 6, 11, 14, 15	B, J
11	Design and develop basic C++ applications using object-oriented programming principles.	1c, 1d, 1e, 2, 3, 4	1, 5, 6, 11, 14, 15	B, J

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	H. M. Deitel, P. J. Deitel, I. Pohl, C++: How To Program, 2016.
Supporting References	Lecture notes and slides.
Necessary Course Material	Accessible computers for each student, Visual Studio 2013 or later, projection

Course Schedule	
1	Laboratory introduction and summary.
2	Array and matrix applications.
3	Pointers applications.
4	Pointer expressions and implementation of pointer arithmetic.
5	Co-implementation of pointer-arrays.
6	Strings applications.
7	Implementation of standard string library and functions.
8	Mid-Term Exam
9	Dynamic memory allocation applications.
10	Application of structs and type definitions.
11	Application of structs and type definitions.
12	Union applications.
13	File I/O applications.
14	Object-oriented programming applications.
15	Object-oriented programming applications.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	1	5	5
Project (Preparation and presentation time included)	1	15	15
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam			
Studying for Final Exam			
		Total workload	**62**
		Total workload / 30	**2.07**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Quiz	70
Project Observation	30
Final Exam	
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	5
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	3
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	2
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İbrahim SAVRAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
ADVANCED READING AND WRITING II	152512008

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	3	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
			2	

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Teaching how to access sources Teaching how to cite and document sources Teaching how to write an academic paper Awareness about plagiarism Writing a paper on current issues that concern the society including health, environment and energy issues.
Short Course Content	Borrowing information from sources, direct quote, paraphrase, summary, intext citations, use of index cards, reliability of the sources, outline, introduction paragraph, body and conclusion paragraphs, style for references, page layout, writing a 5-6 page paper on topics related to health, environment and energy sources.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Produce clear, coherent and appropriately structured academic writing.	9	1, 2, 6	A, B, E
2	Accurately integrate information from sources through summarizing, paraphrasing and direct quotation.	7, 9	1, 2, 6	A, B, E
3	Develop a thesis statement and outline for an academic paper.	9	1, 2, 6	A, B, E
4	Select reliable information sources and format in-text citations and references according to an appropriate academic style.	5, 9, 11	1, 6, 15	A, B, E
5	Synthesize information from multiple sources to draft and revise a 5–6-page academic paper.	5, 9	1, 6, 15	A, B, E
6	Investigate environmental, health and energy issues and evaluate their societal impacts in evidence-based writing.	6, 9	1, 6, 15	E
7	Identify plagiarism and apply academic integrity principles when using sources and preparing academic writing.	7	1, 2, 6	A, B, E
8				
9				
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Stephen Bailey, "Academic Writing: A Handbook for International Students," Routledge, 2018.
Supporting References	Alice Oshima & Ann Hogue, "Longman Academic Writing Series 4," Pearson, 2017.
Necessary Course Material	-

Course Schedule	
1	Introduction to the course
2	Sources of information, Critical analysis of sources
3	Borrowing information from sources
4	Forms of borrowed information
5	Paraphrasing
6	Paraphrasing practice
7	Summaries
8	Mid-Term Exam
9	Blending source information into own writing
10	Research for the topic
11	Developing a thesis statement
12	Planning and Organization
13	Synthesis
14	Revision
15	Printed page format and course review
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam	1	1	1
Studying for Quiz Exam	1	2	2
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	1	8	8
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	2	2
Final Exam			
Studying for Final Exam			
		Total workload	**71**
		Total workload / 30	**2.36**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	30
Quiz	25
Final Exam (report)	45
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	2
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	2
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	3
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	5
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	2

LECTURER(S)				
Prepared by	Lecturer Seda ÜNER			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
DISCRETE COMPUTATIONAL STRUCTURES	152512009

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
2	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
3	1			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To develop knowledge of counting principles, combinatorial methods, elementary number theory, graphs and trees, and to strengthen mathematical proof, discrete modeling and algorithmic problem-solving skills.
Short Course Content	Counting principles, sets, permutations and combinations, mathematical induction, inclusion–exclusion and pigeonhole principles, binomial coefficients, Pascal's triangle, Fibonacci numbers, combinatorial probability, integers, divisibility, prime numbers, graphs, trees, the traveling salesperson problem and graph matching.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Solve counting problems using fundamental counting principles and permutation methods.	1a, 2	1, 2, 5, 10, 11, 13	A
2	Use combinations, binomial coefficients and Pascal's triangle to perform calculations involving finite sets.	1a, 2	1, 2, 5, 10, 11, 13	A
3	Explain properties of finite sets and prove fundamental combinatorial results.	1a	1, 2, 5, 10, 11, 13	A
4	Apply mathematical induction, inclusion–exclusion and the pigeonhole principle to solve discrete problems.	1a, 2	1, 2, 5, 10, 11, 13	A
5	Examine Fibonacci sequences and solve problems in combinatorial probability.	1a, 2	1, 2, 5, 10, 11, 13	A
6	Apply basic number-theoretic methods involving divisibility and prime numbers to problems.	1a, 2	1, 2, 5, 10, 11, 13	A
7	Model problems using graphs and analyze fundamental properties of the traveling salesperson and graph-matching problems.	1a, 1d, 1e, 2, 3	1, 2, 5, 10, 11, 13	A
8	Explain fundamental properties of trees and relate tree structures to problems in computer science.	1a, 1c, 1d, 1e, 2, 3	1, 2, 5, 10, 11, 13	A
9				
10				

***Teaching Methods** 1: Lecture, 2: Discussion, 3: Experiment, 4: Simulation, 5: Question-Answer, 6: Practice, 7: Observation, 8: Case Study, 9: Technical Visit, 10: Problem Solving, 11: Individual Work, 12: Team/Group Work, 13: Brainstorming, 14: Project Design/Management, 15: Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K**:Class Attendance; **L**:Jury Exam

Main Textbook	Lovász, L., Pelikán, J. & Vesztergombi, K. (2003) Discrete Mathematics Elementary and Beyond. Springer.
Supporting References	Example questions
Necessary Course Material	Computer, projection

Course Schedule	
1	Course introduction, basic definitions, scope and resources.
2	Fundamentals of Counting
3	Fundamentals of Counting
4	Combinatorial Methods
5	Binomial Coefficients and Pascal's Triangle
6	Fibonacci Numbers
7	Combinatorial Probability
8	Mid-Term Exam
9	Integers, Divisors, and Prime Numbers
10	Integers, Divisors, and Prime Numbers
11	Graphs
12	Trees
13	The traveling salesperson problem.
14	Graph matching.
15	Review, example problems and questions.
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	12	3	36
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	20	20
Final Exam	1	1	1
Studying for Final Exam	1	20	20
		Total workload	**120**
		Total workload / 30	**4**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	47
Final Exam	53
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low, 1: Very low)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	2
		d. Knowledge of computational methods.	3
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	2
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	2
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Savaş OKYAY			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
SOFTWARE REQUIREMENTS ENGINEERING	152513001

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	4			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Introduce requirements engineering within modern software development and develop skills in stakeholder analysis, elicitation, specification, modelling, validation, prioritization, traceability, and change management. Introduce ISO/IEC/IEEE 29148 and current requirements-management tools through applied examples, and evaluate ethical, security, and sustainability considerations.
Short Course Content	Requirements engineering processes and ISO/IEC/IEEE 29148; stakeholder and scope analysis; elicitation through interviews, workshops, observation, ethnography, and prototyping; functional and non-functional requirements; use cases, user stories, UML and domain models; SRS and product backlog; requirements validation, acceptance testing, and BDD; prioritization with MoSCoW, Kano, and WSJF; traceability and change management; Jira, Confluence, DOORS NG, and ReqIF; agile/DevOps, ethics, and AI-assisted requirements engineering.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Identify stakeholders, system boundaries, and business goals to define the scope of software requirements.	1c, 2, 10	1, 2, 8	A, G
2	Select and apply elicitation techniques such as interviews, workshops, observation, and prototyping to record requirements information.	1e, 2, 5, 8, 9	2, 5, 6, 11	A, D
3	Classify functional and non-functional requirements and prepare a software requirements specification (SRS) consistent with ISO/IEC/IEEE 29148 principles.	1c, 1e, 2, 9	1, 6, 10, 15	A, D, E
4	Represent requirements using use cases, user stories, and UML/domain models, and evaluate consistency across the models.	1e, 2, 3, 4	6, 8, 10	A, J
5	Review requirements for completeness, consistency, clarity, and verifiability, and recommend improvements for identified defects.	2, 5, 8	2, 8, 10	A, E
6	Prioritize requirements using MoSCoW, Kano, or WSJF and apply traceability and change-management procedures to a sample backlog.	2, 4, 8, 10	2, 10, 12, 14	A, G, J
7	Define acceptance criteria and BDD/Gherkin scenarios, demonstrating their relationship to requirements validation and CI/CD workflows.	2, 4, 5	3, 6, 10	A, I, J
8	Evaluate requirements decisions in terms of ethics, law, privacy, security, and sustainability, and discuss the opportunities and limitations of AI-assisted requirements tools.	6, 7, 11	1, 2, 8, 11	A, F, G

***Teaching Methods** **1:**Lecture, **2:**Discussion, **3:**Experiment, **4:**Simulation, **5:**Question-Answer, **6:**Tutorial, **7:**Observation, **8:**Case Study, **9:**Technical Visit, **10:**Problem Solving, **11:**Individual Work, **12:**Team/Group Work, **13:**Brainstorming, **14:**Project Design / Management, **15:**Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Wiegers, K. & Beatty, J. (2023). Software Requirements (4th ed.). Pearson.
Supporting References	Pohl, K. (2010). Requirements Engineering: Fundamentals, Principles, and Techniques. Springer. Hull, E., Jackson, K., Dick, J. (2020). Requirements Engineering (4th ed.). Springer. Leffingwell, D. (2018). SAFe® 4.5 Distilled: Achieving Business Agility with the Scaled Agile Framework. Addison-Wesley. ISO/IEC/IEEE 29148:2018 – Systems and software engineering Selected IEEE/ACM research papers (posted on LMS).
Necessary Course Material	Laptop with internet, modelling tool (e.g., Visual Paradigm, Enterprise Architect, Draw.io), collaborative backlog/issue tracker (Jira/Azure Boards), document templates.

Course Schedule	
1	Introduction to requirements engineering, process models, and ISO/IEC/IEEE 29148
2	Stakeholder and scope analysis, goal modelling
3	Requirements elicitation I: interviews, workshops, and observation
4	Requirements elicitation II: ethnography, prototyping, and questionnaires
5	Requirements classification, quality attributes, and non-functional requirements
6	Requirements modelling I: use cases, user stories, and story mapping
7	Requirements modelling II: UML diagrams and domain models
8	Mid-term examinations
9	Requirements documentation: SRS, product backlog, and specification patterns
10	Verification and validation: reviews, acceptance tests, and BDD
11	Prioritization and negotiation: MoSCoW, Kano, WSJF, and cost–value analysis
12	Traceability, requirements baselines, and change management
13	Tool support: Jira, Confluence, DOORS NG, and ReqIF
14	Requirements engineering in agile/DevOps, ethics, and AI-assisted methods
15	Team project presentations and feedback
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	2	28
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	20	20
Final Exam	1	2	2
Studying for Final Exam	1	30	30
		Total workload	**124**
		Total workload / 30	**4.13**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	5
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	4
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	3
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	2
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	3
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	3
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	4
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	4
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	2

LECTURER(S)				
Prepared by	Assoc. Prof. Dr. Uğur YAYAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
OBJECT ORIENTED PROGRAMMING I	152513002

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	3	0	6

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	4	2		

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To develop the ability to apply object-oriented programming principles in C++; design software with classes and objects; and build reusable components and a project using encapsulation, operator overloading, inheritance, polymorphism, templates, exception handling, and the STL.
Short Course Content	Fundamentals of C++ and object-oriented programming; classes and objects; encapsulation and access specifiers; constructors; operator overloading; inheritance and multiple inheritance; virtual functions and polymorphism; UML class diagrams; templates; exception handling; the Standard Template Library (STL); project presentations.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the fundamental concepts of object-oriented programming and create classes and objects in C++.	1c, 1d	1, 5, 6, 10	A, B, J
2	Apply encapsulation, access specifiers, and constructors appropriately in C++ classes.	1d, 3, 4	1, 5, 6, 10	A, B, J
3	Define suitable operator-overloading functions for C++ classes.	1d, 3, 4	1, 5, 6, 10	A, B, J
4	Develop software structures consisting of base and derived classes using inheritance.	1d, 1e, 3, 4	1, 5, 6, 10	A, B, J
5	Implement polymorphism using virtual functions and late binding.	1d, 1e, 3, 4	1, 5, 6, 10	A, B, J
6	Model the classes and relationships of an object-oriented software design with UML class diagrams.	2, 3	1, 5, 6, 10	A, B, J
7	Develop reusable software components using C++ templates and STL containers.	1d, 1e, 4	1, 5, 6, 10	A, B, J
8	Handle error conditions with exception-handling constructs and verify expected program behavior.	1d, 2, 4	1, 5, 6, 10	A, B, J
9	Develop an object-oriented C++ project and present its class design and results.	1d, 1e, 3, 4, 9	1, 6, 15	J

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Paul Deitel and Harley Deitel, C++ How to Program, 7th Edition, Pearson Education, 2010.
Supporting References	Bruce Eckel, Thinking In C++ Vol.1 and Vol.2 , Second Edition, Prentice-Hall,2000.
Necessary Course Material	a projector and computer

Course Schedule	
1	Introduction: differences between C and C++; overview of object-oriented programming
2	Classes and objects: definitions and members
3	Classes and objects: encapsulation and access specifiers
4	Classes and objects: constructors and class applications
5	Operator overloading
6	Inheritance: base and derived classes
7	Inheritance: access, multiple inheritance, and applications
8	Mid-term examinations
9	Polymorphism: virtual functions
10	Polymorphism: dynamic binding and applications
11	UML class diagrams
12	Templates
13	Exception handling
14	Standard Template Library (STL)
15	Project presentations
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	2	28
Homework			
Quiz Exam	3	1	3
Studying for Quiz Exam	3	5	15
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	2	10	20
Project (Preparation and presentation time included)	2	15	30
Presentation (Preparation time included)	1	10	10
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	5	5
Final Exam	1	2	2
Studying for Final Exam	1	5	5
		Total workload	**162**
		Total workload / 30	**5.4**
		Course ECTS Credit	**6**

Evaluation	
Activity Type	**%**
Mid-term	30
Quiz	10
Project Observation	20
Final Exam	40
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	5
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Aysun TOK ONARCAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
OBJECT ORIENTED PROGRAMMING I LAB	152513003

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	0	2	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	1	1		

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To develop practical C++ skills in designing classes and objects; applying encapsulation, operator overloading, inheritance, and polymorphism; and building object-oriented software with UML, templates, exception handling, and the STL.
Short Course Content	Hands-on C++ programming with classes and objects, encapsulation and constructors, operator overloading, inheritance and multiple inheritance, polymorphism, UML class diagrams, templates, exception handling, STL containers and algorithms, project development and presentations.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Create C++ classes and objects using encapsulation, constructors, and access specifiers.	1c, 1d, 1e, 3, 4	3, 6, 10, 11	I
2	Write and execute suitable operator-overloading functions for C++ classes.	1d, 3, 4	3, 6, 10, 11	I
3	Implement base and derived classes using single and multiple inheritance.	1d, 1e, 3, 4	3, 6, 10, 11	I
4	Implement polymorphic behavior using virtual functions and function overriding.	1d, 1e, 3, 4	3, 6, 10, 11	I
5	Interpret a UML class diagram and implement its classes and relationships in C++.	1d, 1e, 2, 3	3, 6, 10, 11	I
6	Develop reusable C++ function and class templates that operate on different data types.	1d, 1e, 4	3, 6, 10, 11	I
7	Handle expected error conditions safely using C++ exception-handling mechanisms.	1d, 2, 4	3, 6, 10, 11	I
8	Select STL containers and suitable algorithms to solve specified programming problems.	1d, 2, 4	3, 6, 10, 11	I
9	Develop and verify a C++ application using class design, inheritance, polymorphism, and the STL.	1d, 1e, 2, 3, 4	3, 6, 10, 11	I
10	Explain the class design and operation of a developed object-oriented C++ application in a project presentation.	3, 9	6, 11, 15	I

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Paul Deitel and Harley Deitel, C++ How to Program, 7th Edition, Pearson Education, 2010.
Supporting References	Bruce Eckel, Thinking In C++ Vol.1 and Vol.2 , Second Edition, Prentice-Hall,2000.
Necessary Course Material	a projector and computer

Course Schedule	
1	Laboratory introduction: C versus C++ and the development environment
2	Practice with creating classes and objects
3	Practice with encapsulation, constructors, and class members
4	Operator-overloading exercises
5	Inheritance: base and derived classes
6	Inheritance relationships and access specifiers
7	Multiple-inheritance exercises
8	Mid-term examinations
9	Polymorphism: virtual functions and function overriding
10	UML class diagrams and coding exercises
11	Function- and class-template exercises
12	Exception-handling exercises
13	STL containers and algorithms
14	Problem solving and project development using the STL
15	Project presentations
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	3	42
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam			
Studying for Final Exam			
		Total workload	**70**
		Total workload / 30	**2.33**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Experimental Skill	100
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	5
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	5
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Aysun TOK ONARCAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
DATA STRUCTURES	152513004

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	2	0	3

Course Category (Credit)				
Basic Sciences	**Engineering Sciences**	**Design**	**General Education**	**Social**
1	2			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	The aim of this course is to teach fundamental data structures and algorithms and develop students' ability to analyze, select, and implement appropriate data structures.
Short Course Content	Iterative and recursive algorithms; contiguous and linked memory allocation; array-based lists, linked lists, stacks, and queues; trees, binary search trees (BST), AVL trees, and Splay trees; set and map abstract data types (ADTs); comparison-based search and hash tables; selecting and implementing suitable data structures.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Compare iterative and recursive algorithms and develop solutions to fundamental problems.	1d, 2	1, 5, 10	A
2	Analyze the time complexity of algorithms and compare the efficiency of alternative solutions.	1a, 1d, 2	1, 5, 10	A
3	Compare contiguous and linked memory allocation and select suitable data structures for a given problem.	1c, 1d, 2	1, 5, 10	A
4	Implement array-based lists, linked lists, stacks, and queues.	1c, 1d, 1e, 4	1, 5, 10	A
5	Perform basic data-processing operations using trees and binary search trees.	1d, 1e, 3	1, 5, 10	A
6	Perform search and update operations on AVL and Splay trees and explain how these structures work.	1d, 3, 4	1, 5, 10	A
7	Implement set and map abstract data types using appropriate data structures.	1c, 1e, 3	1, 5, 10	A
8	Evaluate comparison-based search methods and hash tables in terms of time and memory cost and select suitable solutions.	1a, 1d, 1e, 2, 3, 4	1, 5, 10	A

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brain Storm, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Data Structures and Algorithm Analysis, Mark Allen Weiss
Supporting References	Introduction to Data Structures, Bhagat Singh, Thomas L. Naps
Necessary Course Material	-

Course Schedule	
1	Iterative algorithms
2	Recursive algorithms
3	Contiguous and linked memory allocation
4	Array-based lists
5	Linked lists
6	Queues
7	Stacks
8	Mid-term exam
9	Trees
10	Set and map ADTs; binary search trees (BST)
11	AVL trees
12	Splay trees
13	Comparison-based search and hash tables
14	Hash tables; further set and map ADTs
15	Review
16,17	Final exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	20	20
Final Exam	1	1	1
Studying for Final Exam	1	20	20
		Total workload	**84**
		Total workload / 30	**2.8**
		Course ECTS Credit	**3**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	3
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	3
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İlker ÖZÇELİK			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
DATA STRUCTURES LAB	152513005

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	0	2	2

Course Category (Credit)				
Basic Sciences	**Engineering Sciences**	**Design**	**General Education**	**Social**
1	1			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	The aim of this laboratory course is to develop practical skills in implementing fundamental data structures and algorithms, selecting suitable structures, and testing program correctness and efficiency.
Short Course Content	Practical exercises in iterative and recursive algorithms; contiguous and linked memory allocation; array-based, singly and doubly linked lists, stacks and queues; trees, binary search trees (BST), AVL and Splay trees; set and map abstract data types; comparison-based searching and hash tables; evaluation of time and space costs.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Implement iterative and recursive algorithms to develop working solutions to given problems.	1d, 1e, 2, 4	1, 5, 6, 10	D
2	Implement contiguous and linked memory allocation and compare the time and space costs of different data structures.	1a, 1d, 2, 5	1, 5, 6, 10	D
3	Implement array-based, singly linked, and doubly linked lists and test basic insertion, deletion, and search operations.	1c, 1d, 1e, 3, 4	1, 5, 6, 10	D
4	Implement stack and queue abstract data types and apply them to appropriate problems.	1d, 1e, 3, 4	1, 5, 6, 10	D
5	Implement and compare search, insertion, and deletion operations in binary search, AVL, and Splay trees.	1d, 1e, 3, 4	1, 5, 6, 10	D
6	Implement set and map abstract data types using suitable tree and hash-table structures.	1c, 1d, 1e, 3, 4	1, 5, 6, 10	D
7	Test search methods and hash tables on sample inputs and compare their time complexity and performance.	1a, 1d, 2, 5	1, 5, 6, 10	D
8	Select an appropriate data structure for a given software problem, implement a solution, and test its correctness.	1e, 2, 3, 4, 5	1, 5, 6, 10	D

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brain Storm, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Introduction to Data Structures, Bhagat Singh, Thomas L. Naps
Supporting References	Data Structures and Algorithm Analysis, Mark Allen Weiss
Necessary Course Material	-

Course Schedule	
1	Laboratory introduction and use of the development environment
2	Iterative and recursive algorithm exercises
3	Contiguous and linked memory allocation exercises
4	Array-based list exercises
5	Linked list exercises
6	Queue exercises
7	Stack exercises
8	Mid-term examination week
9	Tree and binary search tree exercises
10	Set/map abstract data types and binary search tree exercises
11	AVL tree exercises
12	Splay tree exercises
13	Comparison-based search and hash table exercises
14	Hash tables and set/map abstract data type exercises
15	General review and evaluation of implementations
16,17	Final examination week

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework	7	2	14
Quiz Exam	7	1	7
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam			
Studying for Final Exam			
		Total workload	**63**
		Total workload / 30	**2.1**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Homework	100
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	2
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	5
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	2
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İlker ÖZÇELİK			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
ATATURK'S PRINCIPLES AND HISTORY OF TURKISH REVOLUTION I	151011208

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	2	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
			2	

Course Language	Course Level	Course Type
Turkish	Undergraduate	Compulsory

Prerequisite(s) if any	None
Objectives of the Course	Students can understand the Liberation War under the leadership of Atatürk and the foundation of the young Turkish Republic. In addition to that the students will learn the processes of the liberation war and the conditions before the foundation of the Republic.
Short Course Content	Concepts of reform and revolution; Ottoman modernization and developments before the First World War; the First World War and the Armistice of Mudros; Mustafa Kemal Pasha; national resistance organizations; Samsun and the congresses; the National Pact and the opening of the Grand National Assembly; the National Struggle, the İnönü victories, Kütahya-Eskişehir and Sakarya battles, and the Great Offensive.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Define and distinguish the concepts of reform, revolution, uprising, and political change.	9, 11	1	A
2	Explain the Ottoman Empire's modernization efforts and political developments before the First World War.	9, 11	1	A
3	Explain the causes of the First World War, the Ottoman Empire's entry into the war, and the fronts on which it fought.	9, 11	1	A
4	Evaluate the main provisions of the Armistice of Mudros and the subsequent occupation of Ottoman territories.	9, 11	1	A
5	Summarize the principal stages of Mustafa Kemal Pasha's life and his activities before the National Struggle.	9, 11	1	A
6	Explain the roles of Mustafa Kemal's arrival in Samsun and the congresses in organizing the National Struggle.	9, 11	1	A
7	Explain the National Pact (Misak-ı Millî) and the significance of the opening of the Grand National Assembly for national sovereignty.	9, 11	1	A
8	Compare the formation and roles of the National Forces (Kuva-yı Milliye) and the regular army during the National Struggle.	9, 11	1	A
9	Evaluate the effects of the İnönü victories and the Kütahya-Eskişehir battles on the course of the National Struggle.	9, 11	1	A
10	Explain the developments, outcomes, and historical significance of the Battle of Sakarya and the Great Offensive.	9, 11	1	A

***Teaching Methods** **1:**Expression, **2:**Discussion, **3:**Experiment, **4:**Simulation, **5:**Question-Answer, **6:**Tutorial, **7:**Observation, **8:**Case Study, **9:**Technical Visit, **10:**Trouble/Problem Solving, **11:**Induvidual Work, **12:**Team/Group Work, **13:**Brain Storm, **14:**Project Design / Management, **15:**Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Turan Şerafettin, *Türk Devrim Tarihi, C.I-II*, İstanbul, 1991–1995
Supporting References	Ateş, Toktamış, *Türk Devrim Tarihi*, İstanbul: Der Yayınları, 2001. Aybars, Ergün, *Türkiye Cumhuriyeti Tarihi*, İzmir: Ercan Kitabevi, 2000. Eroğlu, Hamza, *Türk İnkılap Tarihi*, Ankara: Savaş Yayınları, 1990. Kongar, Emre, *Devrim Tarihi ve Toplumbilim Açısından Atatürk*, İstanbul: Remzi Kitabevi, 1999. Selek, Sebahattin, *Anadolu İhtilali,* İstanbul: Kastaç Yayınları, 1987. Timur, Taner, *Türk Devrimi ve Sonrası*, Ankara: İmge Kitabevi, 1997.
Necessary Course Material	

Course Schedule	
1	The teaching of the concepts of Revolution, Evolution, Uprising etc.
2	The attempts for the modernisation of the Ottoman Empire and the political thoughts
3	The Wars of Trablusgarp and the Balkans
4	The Beginning of the Great War and the join of the Ottoman Empire
5	The fronts in which the Ottoman Empire fought
6	The end of the war and the partition of the Ottoman
7	The Treaty of Mudros and the invasion of the Ottoman lands
8	Mid-Term Exam
9	Mustafa Kemal Pasha's arrival in Samsun, the start of the National Struggle, and the congresses.
10	National Oath and the opening of Turkish Grand National Assembly
11	National Assembly and the direction of liberation war
12	National Forces and the foundation of the national army
13	First and Second Victories of İnönü; The battles of Kütahya-Eskişehir
14	The Battle of Sakarya
15	Great Attack of 30th August
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1.5	21
Homework			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	0.5	0.5
Studying for Mid-Term Exam	1	5	5
Final Exam	1	0.5	0.5
Studying for Final Exam	1	5	5
		Total workload	**60**
		Total workload / 30	2.00
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	3
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	4

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Serkan DEMİRBAŞ			
Signature(s)				

Date:09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
OCCUPATIONAL HEALTH AND SAFETY I	152513006

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	2	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	2			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To enable students to explain basic occupational health and safety principles, approaches to preventing occupational accidents and diseases, hazard identification and risk assessment, protective measures, and fundamental legal responsibilities.
Short Course Content	Definition and importance of occupational health and safety; safety culture; occupational accidents, causes and consequences; occupational diseases; workplace risk factors; basic workplace precautions; hazard identification and risk assessment; personal protective equipment; fire safety; occupational health and safety legislation.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain fundamental occupational health and safety concepts and the importance of a workplace safety culture.	6, 7	1, 2	A
2	Classify occupational accidents by their causes and types, and interpret their possible consequences.	6	1, 2	A
3	Identify workplace conditions and major risk factors that can lead to occupational diseases.	6	1, 2	A
4	Identify workplace hazards and explain the main stages of occupational risk assessment.	6	1, 2	A
5	Assess risks in sample workplace scenarios and propose appropriate preventive and protective measures.	6, 7	1, 2	A
6	Explain basic workplace safety precautions and the purposes of personal protective equipment.	6	1, 2	A
7	List basic fire-prevention, protection, and emergency response measures.	6	1, 2	A
8	Explain the general scope of occupational health and safety legislation and employer and employee responsibilities.	6, 7	1, 2	A

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	1. Kahya, E., 2014, İş Güvenliği, ESOGÜ Yayın No :246, Eskişehir.
Supporting References	1.Yiğit, A., İş Güvenliği, 2013, Dora basım-Yayın Dağıtım Ltd. Şti, Bursa. 2.Bayır, M. ve Ergül, M., 2006, İş Güvenliği ve Risk Değerlendirme Uygulamaları, Bursa. 3.Dizdar, E.N., 2008, İş Güvenliği, 4.Baskı, Murathan Yayınevi, Trabzon.
Necessary Course Material	

Course Schedule	
1	Course scope, delivery, assessment, and introduction to occupational health and safety
2	Workplace safety culture
3	Occupational accidents: causes, types, and performance indicators
4	Occupational accidents: causation theories, statistics, and investigations
5	Occupational diseases
6	Risk factors in the workplace
7	Basic workplace safety precautions
8	Mid-term examinations
9	Basic workplace safety precautions
10	Risk assessment: hazard identification
11	Risk assessment: risk evaluation and control measures
12	Personal protective equipment
13	Fire safety
14	Occupational health and safety legislation
15	Occupational health and safety legislation
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	7	7
Final Exam	1	1	1
Studying for Final Exam	1	10	10
		Total workload	**61**
		Total workload / 30	**2.03**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	5
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	3
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Uğur SELENGİL			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
NUMERICAL METHODS	152513007

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
4				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To enable students to obtain approximate solutions to mathematical problems using numerical methods, analyze errors and convergence, select algorithms appropriate to a given problem, and compute and verify solutions in Python.
Short Course Content	Sources of error, round-off and truncation errors; Taylor series; root finding and convergence; Gaussian elimination, LU decomposition, Gauss–Seidel; least-squares regression and interpolation; numerical differentiation and integration; numerical solutions of ordinary differential equations; eigenvalue and eigenvector problems; unconstrained and constrained optimization, simplex method; Python implementations and forecasting by curve fitting.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Distinguish round-off and truncation errors in numerical modeling and calculate approximate errors and stopping criteria.	1a, 1d	1, 5, 6, 10	A, D
2	Approximate roots of equations using bracketing and open methods and compare their convergence properties.	1a, 1d, 2	1, 5, 6, 10	A, D
3	Solve systems of linear equations using Gaussian elimination, LU decomposition, and Gauss–Seidel methods.	1a, 1d, 4	1, 5, 6, 10	A, D
4	Construct mathematical models from data using interpolation and least-squares regression and interpret the results.	1a, 1d, 1e, 5	1, 5, 6, 10	A, D
5	Apply numerical differentiation and integration methods and evaluate approximation errors.	1a, 1d	1, 5, 6, 10	A, D
6	Solve ordinary differential equations numerically using Taylor, Euler, Heun, and multistep methods.	1a, 1d, 4	1, 5, 6, 10	A, D
7	Compute approximate numerical solutions for eigenvalues and eigenvectors.	1a, 1d	1, 5, 6, 10	A, D
8	Formulate unconstrained and constrained optimization problems and apply appropriate numerical methods, including the simplex method when needed.	1a, 1e, 2	1, 5, 6, 10	A, D
9	Develop and verify numerical solution algorithms in Python and construct a simple forecasting model using curve fitting.	1d, 1e, 2, 4, 5	1, 5, 6, 10	A, D

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	1- Numerical Methods for Engineers. Sixth Edition. Steven C. Chapra and Raymond P. Canale
Supporting References	1- Laurene V. Fausett, Numerical Methods: Algorithms and Applications, Prentice Hall, 2003 2- Kiusalaas, Jaan. Numerical methods in engineering with Python. Cambridge University press, 2010
Necessary Course Material	Projector, Computer

Course Schedule	
1	Introduction to numerical methods and computation in Python
2	Mathematical modeling, engineering problems, and round-off errors
3	Truncation errors and Taylor series; bracketing and open methods for root finding
4	Gaussian elimination for solving systems of linear equations
5	LU decomposition and matrix inversion
6	Special matrices, the Gauss–Seidel method, and one-variable unconstrained optimization
7	Curve fitting and least-squares regression
8	Mid-term examinations
9	Interpolation methods
10	Numerical differentiation and integration
11	Ordinary differential equations: Taylor, Euler, Heun, stiffness, and multistep methods
12	Numerical eigenvalue and eigenvector problems
13	Constrained optimization and the simplex method
14	Worked examples using numerical methods
15	Developing a forecasting model using curve fitting
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	10	2	20
Homework (Preparation and presentation time included)	5	8	40
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	10	10
Final Exam	1	2	2
Studying for Final Exam	1	10	10
		Total workload	**126**
		Total workload / 30	**4.2**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	30
Homework	20
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	3
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	3
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	4
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	2
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Aysun TOK ONARCAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
DIFFERENTIAL EQUATIONS	152513008

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
3	3	0	4

Course Category (Credit)				
Basic Sciences	**Engineering Sciences**	**Design**	**General Education**	**Social**
4				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	The aim of this course is to develop the ability to classify ordinary differential equations, select and apply analytical and numerical solution methods, interpret mathematical models, and obtain symbolic and numerical solutions using MATLAB.
Short Course Content	First-order ordinary differential equations (separable, exact, homogeneous, linear, Bernoulli, and Riccati); orthogonal trajectories; higher-order linear equations and reduction of order; undetermined coefficients, variation of parameters, and Cauchy–Euler equations; power series; differential operators; Laplace transforms; eigenvalue problems; symbolic and numerical solution exercises in MATLAB.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Classify ordinary differential equations by type and order and formulate initial-value problems.	1a	1, 2, 5, 10	A
2	Solve first-order separable, exact, homogeneous, linear, Bernoulli, and Riccati equations using appropriate methods.	1a	1, 2, 5, 10	A
3	Analyze and solve orthogonal-trajectory problems using differential equations.	1a	1, 2, 5, 10	A
4	Solve higher-order linear equations using reduction of order and methods for constant-coefficient equations.	1a	1, 2, 5, 10	A
5	Apply undetermined coefficients and variation of parameters to nonhomogeneous equations, and appropriate methods to Cauchy–Euler equations.	1a	1, 2, 5, 10	A
6	Obtain solutions to differential equations using power series, differential operators, and Laplace transforms.	1a	1, 2, 5, 10	A
7	Formulate eigenvalue problems mathematically and interpret their results.	1a	1, 2, 5, 10	A
8	Compute symbolic and numerical solutions of differential equations in MATLAB and verify the results.	1a, 1d, 4	1, 2, 5, 10	A
9	Model a given physical or engineering problem with a differential equation, select an appropriate solution method, and interpret the results.	1a, 1e, 2	1, 2, 5, 10	A

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Özer, N. ve, Eser, D. "Diferansiyel Denklemler", Eskişehir 2010.
Supporting References	
Necessary Course Material	

Course Schedule	
1	Notation, basic definitions, and classification of differential equations
2	Solutions of ordinary differential equations, initial-value problems, and introduction to MATLAB
3	Exact and separable differential equations, integrating factors; MATLAB exercises
4	Homogeneous, linear, and Bernoulli differential equations; MATLAB exercises
5	Riccati equations and orthogonal trajectories; MATLAB exercises
6	Higher-order differential equations and reduction of order
7	Solutions of linear and constant-coefficient differential equations; MATLAB exercises
8	Mid-term examinations
9	Undetermined coefficients, variation of parameters, and Cauchy–Euler equations; MATLAB exercises
10	Undetermined coefficients, variation of parameters, and Cauchy–Euler equations: problem solving
11	Power series and the differential operator method
12	Laplace transforms and MATLAB exercises
13	Solving differential equations in MATLAB and eigenvalue problems
14	Eigenvalue problems and symbolic solution methods in MATLAB
15	Eigenvalue problems and symbolic solution methods in MATLAB: exercises
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	2	28
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	2	2
Studying for Mid-Term Exam	1	20	20
Final Exam	1	2	2
Studying for Final Exam	1	20	20
		Total workload	**114**
		Total workload / 30	**3.8**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	50
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	3
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	2
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	2
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	2
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Prof. Dr. Dursun ESER			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
SOFTWARE TESTING	152514001

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	3	0	5

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	5			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Explain the role of testing and quality assurance in the software development lifecycle; develop skills in designing requirements-based test strategies and cases, analyzing test results, and automating tests. Introduce test practices with tools such as JUnit, PyTest, Selenium, and JMeter, CI/CD integration, security and performance testing, ethical responsibilities, and emerging AI-assisted approaches.
Short Course Content	Software quality and defect costs; testing levels, V-Model, and risk-based test planning; test documentation under ISO/IEC/IEEE 29119; black-box and white-box techniques; coverage analysis; static reviews; integration, system, acceptance, and regression testing; test automation frameworks and CI/CD; web, mobile, and API testing; performance, load, and security testing; model-based and property-based testing; mutation testing, test metrics, and flaky tests; testing in DevOps and of AI systems; professional ethics.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain fundamental software-testing concepts, testing levels, defect costs, and their role in quality assurance.	1c, 2	1, 2, 5	A
2	Analyze software requirements and risks to prepare an appropriate test plan and testing strategy.	2, 3, 10	1, 2, 8, 10	A, J
3	Design test cases using equivalence partitioning, boundary-value analysis, and white-box techniques, and evaluate their results.	1e, 2, 3, 5	1, 5, 6, 10	A, J
4	Apply statement, branch, and condition coverage criteria to analyze test coverage and propose improvements.	1d, 4, 5	1, 5, 6, 10	A, J
5	Develop automated test suites using JUnit or PyTest and integrate them into a CI/CD workflow.	1d, 1e, 4, 5	1, 6, 10, 11	J
6	Select suitable tools to conduct web, API, performance, and security tests, and interpret the findings.	1e, 4, 5, 6	1, 6, 10, 11	A, J
7	Apply property-based and mutation testing methods to compare the fault-detection effectiveness of test suites.	1d, 4, 5	1, 6, 10	A, J
8	Collaborate to deliver a tested software artifact and present testing findings and quality metrics in a technical report.	1e, 8, 9, 10	12, 14, 15	J, E
9	Critically evaluate AI-assisted test generation and the challenges of testing AI/ML systems using illustrative examples.	4, 5, 11	2, 8, 11	A
10	Evaluate testing decisions in terms of security, privacy, professional ethics, sustainability, and societal impact.	6, 7	1, 2, 8	A

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Ammann, P. & Offutt, J. (2016). Introduction to Software Testing (2nd ed.). Cambridge UP.
Supporting References	Pezzè, M. & Young, M. (2008). Software Testing and Analysis. Wiley. Meszaros, G. (2007). xUnit Test Patterns. Addison-Wesley. Humble, J. & Farley, D. (2011). Continuous Delivery. Addison-Wesley. ISTQB Foundation & Advanced Syllabi (latest versions). Selected recent IEEE/ACM research articles (provided on LMS).
Necessary Course Material	Laptop with admin rights, Git client, IDE (IntelliJ / VS Code / PyCharm), internet access for cloud CI services.

Course Schedule	
1	Software quality, defect costs, the V-Model, and ISO/IEC 25010
2	Requirements analysis, risk-based testing, and ISO/IEC/IEEE 29119 documentation
3	Black-box techniques: equivalence partitioning, boundary-value analysis, and decision tables
4	White-box techniques: statement, branch, path, and condition coverage
5	Static analysis: code reviews, linters, and static analysis tools (Sonar, CodeQL)
6	Integration testing, mocks/stubs, and continuous integration
7	System and acceptance testing, regression testing, and test debt
8	Mid-term examinations
9	Test automation: JUnit, PyTest, Mocha; CI with GitHub Actions and Jenkins
10	Web and mobile testing: Selenium, Cypress, Appium; API testing with Postman/Newman
11	Performance and load testing (JMeter, k6), security and fuzz testing fundamentals
12	Model-based and property-based testing (QuickCheck, Hypothesis)
13	Mutation testing, test metrics, flaky tests, test smells, and dashboards
14	Testing in DevOps and cloud-native environments; ethics and testing AI/ML systems
15	Project presentations, feedback, and exam review
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	10	3	30
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)	1	30	30
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	20	20
Final Exam	1	1	1
Studying for Final Exam	1	30	30
		Total workload	**154**
		Total workload / 30	**5.13**
		Course ECTS Credit	**5**

Evaluation	
Activity Type	**%**
Mid-term	30
Project Observation	30
Final Exam (report)	40
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	4
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	4
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	5
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	2
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	3
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	3
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	3
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	3
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	2

LECTURER(S)				
Prepared by	Assoc. Prof. Dr. Uğur YAYAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
OBJECT ORIENTED PROGRAMMING II	152514002

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	3	0	6

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	4	2		

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Advance object-oriented software development skills using C#/.NET; build desktop interfaces and database connections; and use UML and design patterns to design maintainable, extensible, and reusable software components.
Short Course Content	C#/.NET platform; operators, control flow, loops, and methods; classes, structures, and namespaces; Windows Forms and user-interface controls; SQL Server database connectivity; object-oriented design with UML; creational (Singleton, Factory), structural (Decorator, Facade, Composite, Proxy), and behavioral (Command, Template Method, Iterator, State, Observer) design patterns.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the main components of C#/.NET and the object-oriented approach to application development.	1c, 1d	1, 5, 6	A, J
2	Develop functional C# code using operators, selection and iteration constructs, and methods.	1d, 4	1, 5, 6, 10	A, J
3	Design modular, reusable software components using classes, structures, and namespaces.	1c, 1d, 1e, 3	1, 5, 6, 10	A, J
4	Develop an interactive desktop user interface using Windows Forms controls and event handlers.	1d, 1e, 3, 4	1, 5, 6, 10	A, J
5	Create basic tables in SQL Server and perform database operations through a Windows Forms application.	1d, 1e, 2, 4	1, 5, 6, 10	A, J
6	Model relationships among classes and software components using UML diagrams.	2, 3, 9	1, 6, 10, 11	A, J
7	Apply creational design patterns such as Singleton and Factory to appropriate problems.	1e, 2, 3, 4	1, 5, 6, 10	A, J
8	Apply structural design patterns such as Decorator, Facade, Composite, and Proxy in sample software.	1e, 3, 4	1, 5, 6, 10	A, J
9	Apply behavioral design patterns such as Command, Template Method, Iterator, State, and Observer.	1e, 3, 4	1, 5, 6, 10	A, J
10	Compare design patterns for a software problem, justify a suitable solution, and document its technical design.	2, 3, 9	2, 10, 11, 14, 15	A, J

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	E.Gamma et al. Design Patterns. Addison-Wesley. 1995.
Supporting References	Head First Design Patterns, Freeman and Freeman, O'Reilly, 2004
Necessary Course Material	Computer

Course Schedule	
1	Visual C#/.NET environment and .NET components
2	Operators, control-flow constructs, and loops
3	Methods and functions
4	Classes, structures, and namespaces
5	Windows Forms and controls: ProgressBar, TabControl, TreeView, MenuStrip
6	SQL Server and basic database operations
7	Windows Forms database connectivity, table creation, and sample application
8	Mid-term examinations
9	Singleton and Decorator design patterns; modelling with UML
10	Command and Factory design patterns
11	Facade design pattern
12	Template Method design pattern
13	Iterator and Composite design patterns
14	State and Proxy design patterns
15	Observer design pattern
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	15	3	45
Classroom Studying Time (review, reinforcing, prestudy,….)	15	3	45
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)	1	15	15
Project (Preparation and presentation time included)	1	30	30
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	15	15
Final Exam	1	1	1
Studying for Final Exam	1	20	20
		Total workload	**172**
		Total workload / 30	**5.73**
		Course ECTS Credit	**6**

Evaluation	
Activity Type	**%**
Mid-term	30
Project Observation	20
Final Exam	50
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	5
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	5
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	3
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Uğur GÜREL			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
OBJECT ORIENTED PROGRAMMING II LAB	152514003

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	0	2	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	1	1		

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Reinforce practical object-oriented software development skills in C#/.NET; implement desktop interfaces and database connectivity; and develop maintainable, extensible, and reusable software using UML and design patterns.
Short Course Content	Practical C#/.NET development; operators, selection and iteration, methods; classes, structures, and namespaces; Windows Forms controls and event handling; SQL Server database connectivity; UML class diagrams; implementation of Singleton, Decorator, Command, Factory, Facade, Template Method, Iterator, Composite, State, Proxy, and Observer patterns.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Create a project in the Visual C#/.NET development environment and use fundamental .NET components.	1c, 1d, 4	3, 6, 10, 11	D, I
2	Develop working C# programs using operators, selection and iteration constructs, and methods.	1d, 1e, 2, 4	3, 6, 10, 11	D, I
3	Develop modular software components using classes, structures, namespaces, and encapsulation.	1c, 1d, 1e, 3, 4	3, 6, 10, 11	D, I
4	Build interactive desktop interfaces using Windows Forms controls and event handlers.	1d, 1e, 3, 4	3, 6, 10, 11	D, I
5	Create SQL Server tables and perform basic database operations from a Windows Forms application.	1d, 1e, 2, 4	3, 6, 10, 11	D, I
6	Model relationships among objects and suitable design patterns using UML class diagrams.	2, 3, 4	3, 6, 10, 11	D, I
7	Implement Singleton, Factory, and Decorator design patterns in sample C# applications.	1e, 2, 3, 4	3, 6, 10, 11	D, I
8	Apply Command, Facade, and Template Method design patterns to specified programming problems.	1e, 3, 4	3, 6, 10, 11	D, I
9	Develop sample software using Iterator, Composite, State, Proxy, and Observer patterns.	1e, 3, 4	3, 6, 10, 11	D, I
10	Select an appropriate design pattern for a software problem, test the working solution, and explain design decisions in writing.	1e, 2, 3, 4, 9	6, 10, 11, 15	D, I

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	E.Gamma et al. Design Patterns. Addison-Wesley. 1995.
Supporting References	Head First Design Patterns, Freeman and Freeman, O'Reilly, 2004
Necessary Course Material	Computer

Course Schedule	
1	Exercises with the Visual C#/.NET environment and basic .NET components
2	Coding exercises using operators, control flow, and loops
3	Program development with methods and functions
4	Exercises with classes, structures, namespaces, and encapsulation
5	Windows Forms controls: ProgressBar, TabControl, TreeView, MenuStrip
6	SQL Server: creating databases and tables
7	Windows Forms database connections and sample application
8	Mid-term examination week
9	Singleton and Decorator patterns; UML class diagrams
10	Implementing Command and Factory design patterns
11	Implementing the Facade design pattern
12	Implementing the Template Method design pattern
13	Implementing Iterator and Composite design patterns
14	Implementing State and Proxy design patterns
15	Implementing the Observer design pattern and general review
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework	3	3	9
Quiz Exam	14	1	14
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam			
Studying for Mid-Term Exam			
Final Exam			
Studying for Final Exam			
		Total workload	**65**
		Total workload / 30	**2.17**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Homework	20
Experimental Skill	80
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	5
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	5
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	4
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	5
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	5
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Uğur GÜREL			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
ALGORITHMS AND COMPLEXITY	152514004

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	3	0	5

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	5			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Introduce fundamental algorithm-design approaches and algorithm analysis; develop the ability to evaluate the time and space complexity of solutions using asymptotic notation, recurrences, sorting, randomization, greedy methods, dynamic programming, and graph algorithms.
Short Course Content	Algorithm analysis and asymptotic notation; recursion, divide-and-conquer, and solving recurrences; comparison-based sorting, quicksort, and heapsort; order statistics and randomized algorithms; linear-time sorting; greedy algorithms and Huffman coding; dynamic programming, 0-1 knapsack, and matrix-chain multiplication; graphs, BFS and DFS traversals; comparing time and space complexity.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Express algorithm efficiency using O, Ω, and Θ asymptotic notation and analyze time and space complexity.	1a, 1c, 1d, 2	1, 2, 5, 10, 11	A
2	Design recursive and divide-and-conquer algorithms and solve recurrences using substitution and the Master Theorem.	1a, 1d, 1e, 2, 3	1, 2, 5, 10, 11	A
3	Implement basic comparison-based sorting algorithms and compare their running times and memory requirements.	1d, 1e, 2, 3, 4	1, 2, 5, 10, 11	A
4	Explain randomized algorithms and order-statistics methods and evaluate their expected running times.	1a, 1d, 2, 3	1, 2, 5, 10, 11	A
5	Distinguish comparison-based from linear-time sorting and analyze when counting, radix, and bucket sorting are applicable.	1a, 1d, 2, 3	1, 2, 5, 10, 11	A
6	Design greedy algorithms and evaluate their solution approach and complexity using Huffman coding as an example.	1a, 1d, 1e, 2, 3	1, 2, 5, 10, 11	A
7	Apply top-down and bottom-up dynamic programming to solve knapsack and matrix-chain multiplication problems.	1a, 1d, 1e, 2, 3	1, 2, 5, 10, 11	A
8	Represent graphs, implement BFS and DFS traversals, and calculate their computational complexity.	1c, 1d, 1e, 2, 3	1, 2, 5, 10, 11	A
9	Compare different algorithmic strategies for a computational problem in terms of time, space, and applicability to select an appropriate solution.	1d, 1e, 2, 3, 4	1, 2, 5, 10, 11	A

***Teaching Methods 1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Assessment Methods A:Exam, B:Quiz, C:Oral Exam, D:Homework, E:Report, F:Article Examination, G:Presentation, I:Experimental Skill, J:Project Observation, K:Class Attendance; L:Jury Exam**

Main Textbook	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms, 4th Edition, MIT Press, 2022.
Supporting References	1- Lekh Raj Vermani, Shailini Vermani. An Elementary Approach to Design and Analysis of Algorithms.2019. World Scientific Publishing 2- Soubhik Chakraborty, Prashant Pranav, Naghma Khatoon, Sandip Dutta. A Guide to Desıgn and Analysıs of Algorıthms, 2023, NOVA Press
Necessary Course Material	

Course Schedule	
1	Description of the concept of algorithms, Asymptotic notations
2	Recursion, Divide and Conquer
3	Solving recurrences: Master Theorem and substitution method
4	Sorting, Sorting Algorithms, Selection Sort, Bubble Sort, Insertion Sort, Merge Sort
5	Quicksort
6	Medians and Order Statistics, Randomized Algorithms, Monte Carlo & Las Vegas Strategies
7	Heaps and Heapsort
8	Midterm
9	Linear Time Sorting, Decision Trees, Counting Sort, Radix Sort, Bucket Sort
10	Greedy Algorithms, Sample Problems
11	Huffman coding and prefix trees
12	Dynamic Programming, Top-Down & Bottom Up Approaches, Sample Problems
13	Dynamic Programming, 0-1 Knapsack Problem, Matrix Chain Multiplication
14	Graph Theory
15	Graph traversals: complexity of BFS and DFS
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	14	3	42
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	25	25
Final Exam	1	1	1
Studying for Final Exam	1	30	30
		Total workload	**141**
		Total workload / 30	**4.7**
		Course ECTS Credit	**5**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	4
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	4
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	3
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	5
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	2
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Emrah ATILGAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
THEORY OF COMPUTATION	152514005

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	3	0	5

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	5			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Develop foundational knowledge and proof skills in formal languages, automata, computability, and complexity theory; enable students to represent computational problems with appropriate formal models and analyze decidability and complexity limits. Demonstrate the connection between theoretical foundations and software verification through parsers, SAT/SMT solvers, and model-checking applications.
Short Course Content	Sets, relations, and proof techniques; regular languages, DFA/NFA, regular expressions, pumping lemma, and Myhill–Nerode theorem; context-free grammars and pushdown automata, normal forms, LL/LR parsing; Turing machines, Church–Turing thesis, decidability, halting problem, and reductions; time and space complexity, P, NP, NP-completeness, PSPACE, and EXPTIME; SAT/SMT solving, model checking, and simple parser/verifier applications.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Define regular languages and perform equivalent transformations among DFAs, NFAs, and regular expressions.	1a, 1c, 2	1, 2, 5	A
2	Construct formal proofs of language regularity or non-regularity using closure properties, pumping lemmas, and the Myhill–Nerode theorem.	1a, 2	1, 5, 10	A
3	Design context-free grammars and pushdown automata, and construct a simple parser model.	1a, 1d, 3, 4	1, 6, 10	A, J
4	Model computational problems using Turing machines and distinguish decidable from recognizable languages.	1a, 2	1, 5, 10	A
5	Justify the undecidability of selected problems using reductions and Rice's theorem.	1a, 2	1, 5, 10	A
6	Examine algorithmic problems in terms of P, NP, NP-complete, PSPACE, and EXPTIME classes and evaluate time and space requirements.	1a, 2	1, 5, 10	A
7	Use a SAT/SMT solver or model checker to verify properties of simple finite-state systems and interpret the results.	1d, 1e, 3, 4	1, 6, 10	A, J
8	Develop a parser or formal-verification example and explain the model choice, limitations, and results in a technical presentation.	1e, 3, 4, 9	6, 14, 15	J

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Sipser, M. (2023). Introduction to the Theory of Computation (4th ed.). Cengage.
Supporting References	Hopcroft, J.E., Motwani, R., Ullman, J.D. (2019). Automata Theory, Languages and Computation (4th ed.). Pearson. Kozen, D. (2012). Automata & Computability. Springer. Baier, C. & Katoen, J.P. (2018). Principles of Model Checking. MIT Press. Selected recent ACM/IEEE research papers on SAT/SMT and software verification (provided on LMS).
Necessary Course Material	Scientific calculator or symbolic math software, laptop with Python/Java, access to JFLAP, ANTLR, Z3 or similar tools.

Course Schedule	
1	Sets, relations, proof techniques; DFA/NFA review and conversions
2	Regular expressions, algebraic properties, and lexical analyzer (lexer) design
3	Pumping lemma and non-regularity proofs; Myhill–Nerode
4	Context-free grammars (CFGs), derivations, and parse trees
5	Pushdown automata (PDAs) and CFG–PDA equivalence
6	Normal forms (CNF, GNF), LL/LR parsing, and parser generators
7	Pumping lemma for context-free languages; closure and decision properties
8	Mid-term examinations
9	Turing machines, variants, and the Church–Turing thesis
10	Decidable and recognizable languages; universal Turing machine and halting problem
11	Reductions, Rice's theorem, and other undecidable problems
12	Time complexity and asymptotic analysis; classes P and NP
13	NP-completeness, Cook–Levin theorem, and common reductions
14	PSPACE and EXPTIME; introduction to SAT/SMT solving and model checking
15	Mini-project presentations (e.g., CFG parser or SAT-based verifier)
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	10	3	30
Homework			
Quiz Exam			
Studying for Quiz Exam			
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)	1	30	30
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	20	20
Final Exam	1	1	1
Studying for Final Exam	1	30	30
		Total workload	**154**
		Total workload / 30	**5.13**
		Course ECTS Credit	**5**

Evaluation	
Activity Type	**%**
Mid-term	30
Project Observation	30
Final Exam	40
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	3
		d. Knowledge of computational methods.	4
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	3
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	5
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	4
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	4
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	2
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. İbrahim SAVRAN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
ATATURK'S PRINCIPLES AND HISTORY OF TURKISH REVOLUTION II	151012209

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	2	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
			2	

Course Language	Course Level	Course Type
Turkish	Undergraduate	Compulsory

Prerequisite(s) if any	None
Objectives of the Course	This course aims to help students understand the establishment of the Republic following the victory of the Turkish War of Independence, particularly in the aftermath of the Treaty of Lausanne, as well as Atatürk's principles and reforms. It seeks to cultivate individuals who embrace and uphold secular, democratic, and modern values.
Short Course Content	The Armistice of Mudanya, the Abolition of the Sultanate, the Treaty of Lausanne, the Proclamation of the Republic, the Abolition of the Caliphate, the Constitution of 1924, the transition attempts to multi-party political life, the Sheikh Said Rebellion, other reactions against the Republic, the Alphabet Reform, the University Reform, the History and Language Reforms, reforms in the fields of economy, socio-economic life and law, domestic and foreign policies pursued during Atatürk's era, Atatürk's Principles, and developments in Turkey and the world following Atatürk's death.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the circumstances and historical outcomes of the Armistice of Mudanya and the Treaty of Lausanne.	9, 11	1	A
2	Explain chronologically the abolition of the Sultanate, the proclamation of the Republic, and the abolition of the Caliphate.	9, 11	1	A
3	Evaluate the attempts to introduce multiparty politics, the İzmir assassination attempt, and the Menemen Incident in their historical contexts.	9, 11	1	A
4	Explain early Republican constitutional reforms, the Civil Code, and developments concerning women's rights.	9, 11	1	A
5	Evaluate the aims of the Law on the Unification of Education, the Alphabet Reform, and reforms in language, history, and education.	9, 11	1	A
6	Analyse agricultural, industrial, and tax reforms in relation to the principle of statism.	9, 11	1	A
7	Explain the social impacts of reforms concerning dress, dervish lodges and shrines, surnames, and the weekly holiday.	9, 11	1	A
8	Compare major foreign-policy issues, international partnerships, and treaties during Atatürk's presidency.	9, 11	1	A
9	Define Atatürk's six fundamental principles and complementary principles and relate them to historical developments.	9, 11	1	A
10	Compare major domestic and foreign-policy developments during the İsmet İnönü and Democrat Party periods.	9, 11	1	A

***Teaching Methods** **1:**Expression, **2:**Discussion, **3:**Experiment, **4:**Simulation, **5:**Question-Answer, **6:**Tutorial, **7:**Observation, **8:**Case Study, **9:**Technical Visit, **10:**Trouble/Problem Solving, **11:**Induvidual Work, **12:**Team/Group Work, **13:**Brain Storm, **14:**Project Design / Management, **15:**Report Preparation and/or Presentation

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	Turan Şerafettin, Türk Devrim Tarihi, C.I-II, İstanbul, 1991–1995
Supporting References	Ateş, Toktamış, Türk Devrim Tarihi, İstanbul: Der Yayınları, 2001. Aybars, Ergün, Türkiye Cumhuriyeti Tarihi, İzmir: Ercan Kitabevi, 2000. Eroğlu, Hamza, Türk İnkılap Tarihi, Ankara: Savaş Yayınları, 1990. Kongar, Emre, Devrim Tarihi ve Toplumbilim Açısından Atatürk, İstanbul: Remzi Kitabevi, 1999. Selek, Sebahattin, Anadolu İhtilali, İstanbul: Kastaç Yayınları, 1987. Timur, Taner, Türk Devrimi ve Sonrası, Ankara: İmge Kitabevi, 1997.
Necessary Course Material	

Course Schedule	
1	The Armistice of Mudanya; The Abolition of the Sultanate and the Treaty of Lausanne
2	The Proclamation of the Republic and the Abolition of the Caliphate
3	Efforts to Transition to Multi-Party Political Life; The İzmir Assassination Attempt and the Menemen Incident
4	Reforms in the Field of Law: The Constitutions of the New Turkish State
5	Legal Reforms: Adoption of the Turkish Civil Code and measures concerning women's rights
6	Reforms in Education and Culture: The Law on the Unification of Education (Tevhid-i Tedrisat Law), the Adoption of the Latin Alphabet, and Reforms in Language, History, and Other Fields
7	Reforms Concerning Economic Life: The Abolition of the Tithe (Aşar) Tax, Developments in Agriculture and Industry, and Statism
8	Mid-Term Exam
9	Reforms Concerning Social Life: The Dress Reform, the Closure of Dervish Lodges and Shrines, the Surname Law, and the Regulation of the Weekly Holiday
10	Turkish Foreign Policy During Atatürk's Era: The Etabli Issue, the Mosul Question, and Relations with Foreign States
11	Turkish Foreign Policy During Atatürk's Era: Membership in the League of Nations, the Balkan Pact, the Montreux Convention Regarding the Straits, and the Saadabad Pact
12	Atatürk's Principles: Republicanism, Secularism, Reformism, Nationalism, Populism, and Statism
13	The Complementary Principles of the Atatürkist Thought System
14	Domestic and Foreign Political Developments During the İsmet İnönü Period
15	The Democrat Party Period
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1.5	21
Homework			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	0.5	0.5
Studying for Mid-Term Exam	1	5	5
Final Exam	1	0.5	0.5
Studying for Final Exam	1	5	5
		Total workload	**60**
		Total workload / 30	2.00
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	3
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	4

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Serkan DEMİRBAŞ			
Signature(s)				

Date:09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
OCCUPATIONAL HEALTH AND SAFETY II	152514006

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	2	0	2

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
	2			

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	To teach fundamental concepts of occupational health and safety, environmental health, and information security; develop awareness of occupational accidents, diseases, ergonomics, safety culture, environmental pollution, and waste management; and enable students to explain preventive measures, relevant legal responsibilities, and the basic principles of ISO/IEC 27001.
Short Course Content	OHS concepts and historical development; safety culture, organization, and ergonomics; socioeconomic and legal aspects of occupational accidents; national and international organizations; solid and electronic waste, radioactive contamination, air, water, and soil pollution; occupational diseases; fire, electrical, and laboratory safety, first aid; information security, ISO/IEC 27001, IT law, and cybercrimes.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Explain the development and fundamental concepts of occupational health and safety and workplace safety culture.	6, 7	1, 2	A
2	Identify ergonomic hazards in the workplace and propose basic protective measures.	6	1, 2	A
3	Explain the socioeconomic consequences of occupational accidents, legal responsibilities, and the roles of relevant organizations.	6, 7	1, 2	A
4	Classify solid waste, electronic waste, and radioactive contamination, and assess their health and environmental impacts.	6	1, 2	A
5	Compare major sources of air, water, and soil pollution and basic pollution-control methods.	6	1, 2	A
6	Classify occupational diseases by their causes and explain preventive measures.	6	1, 2	A
7	Explain fire, electrical, and laboratory safety requirements and basic first-aid precautions using example scenarios.	6	1, 2	A
8	Explain information-security principles and the fundamental requirements of the ISO/IEC 27001 information security management system.	7	1, 2	A
9	Evaluate OHS and IT-related legal responsibilities, cybercrimes, and preventive measures in example cases.	6, 7	1, 2	A

***Teaching Methods** **1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Measuring Methods** **A:**Exam, **B:**Quiz, **C:**Oral Exam, **D:**Homework, **E:**Report, **F:**Article Examination, **G:**Presentation, **I:**Experimental Skill, **J:**Project Observation, **K:**Class Attendance; **L:**Jury Exam

Main Textbook	1. Kara, S., Yıldırım, M.E., Tuncel, M., Kıvanç, M., Tamer, Ü., Özdemir, A., Kaytakoğlu, S., Ergun, B., Döğeroğlu, T., Var, F., Uygan., N., Tezcan, Ü., Lüle, M., Çevre Sağlığı, (Ed: N. Varcan), Anadolu Üniversitesi, Açık Öğretim Fakültesi Yayınları, 1995. 2. Dizdar, E., İş Güvenliği, Murathan Yayınevi, Trabzon, 2008.
Supporting References	1. Yiğit, A., İş güvenliği ve İşçi Sağlığı, Aktüel Yayınları, İstanbul, 2005. 2.Crowl, D.A. and Louvar, J.F., Chemical Process Safety Fundamentals with Applications, Prentice Hall, 2002. 3.Karpuzcu, M. , Çevre Kirlenmesi ve Kontrolü, Dördüncü baskı, İstanbul, 1994.
Necessary Course Material	Computer, projector.

Course Schedule	
1	Course scope, assessment principles, and introduction to occupational health and safety
2	Basic definitions and concepts in occupational health and safety
3	Historical development of OHS concepts and rules
4	OHS policies and workplace safety culture
5	OHS organization and methods; ergonomics and occupational safety
6	Socioeconomic impacts of occupational accidents, legal responsibilities, preventive measures, and IT law
7	National and international organizations in occupational health and safety
8	Mid-term examinations
9	Solid waste, electronic waste, and radioactive contamination
10	Air, water, and soil pollution and basic control approaches
11	Occupational diseases: classification, risk factors, and prevention
12	Fire and electrical safety; basic first aid
13	Laboratory safety and information security
14	ISO/IEC 27001 information security management system
15	Cybercrimes and preventive measures
16,17	Final examinations

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	2	28
Classroom Studying Time (review, reinforcing, prestudy,….)	14	1	14
Homework			
Quiz Exam			
Studying for Quiz Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	7	7
Final Exam	1	1	1
Studying for Final Exam	1	10	10
		Total workload	**61**
		Total workload / 30	**2.03**
		Course ECTS Credit	**2**

Evaluation	
Activity Type	**%**
Mid-term	40
Final Exam	60
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO) (5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)			
No	**PROGRAM OUTCOME**		**Level**
1	Engineering Knowledge	a. Knowledge of mathematics.	
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	5
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	4
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Assoc. Prof. Dr. Yıldıray ANAGÜN			
Signature(s)				

Date: 09.11.2026

ESOGU SOFTWARE ENGINEERING DEPARTMENT

COURSE INFORMATION FORM

Course Name	Course Code
PROBABILITY	152514007

Semester	Number of Course Hours per Week		ECTS
	Theory	Practice	
4	3	0	4

Course Category (Credit)				
Basic Sciences	Engineering Sciences	Design	General Education	Social
4				

Course Language	Course Level	Course Type
English	Undergraduate	Compulsory

Prerequisite(s) if any	
Objectives of the Course	Introduce the foundations of probability theory, conditional probability, and random variables; develop the ability to analyze discrete and continuous distributions, calculate expectation and measures of dispersion, and interpret data using basic statistical methods.
Short Course Content	Sets and counting techniques; probability axioms; conditional probability and independence, Bayes' theorem; discrete and continuous random variables; probability distributions and the normal distribution; multivariate random variables; expectation, moments, and variance; measures of central tendency and dispersion; correlation and regression.

	Learning Outcomes of the Course	Contributed PO(s)	Teaching Methods *	Measuring Methods **
1	Use set operations and basic counting techniques to determine sample spaces and events.	1a	1, 5, 11	A, B
2	Apply probability axioms and addition rules to calculate probabilities of events.	1a	1, 5, 11	A, B
3	Solve problems involving conditional probability, independence of events, and Bayes' theorem.	1a	1, 5, 11	A, B
4	Define discrete random variables and probability mass functions and calculate associated probabilities.	1a	1, 5, 11	A, B
5	Use continuous random variables and probability density functions to calculate probabilities for normal and other continuous distributions.	1a	1, 5, 11	A, B
6	Analyze multivariate and independent random variables and determine joint and marginal probabilities.	1a	1, 5, 11	A, B
7	Calculate and interpret expected values, moments, and variances of random variables.	1a	1, 5, 11	A, B
8	Calculate mean, median, mode, standard deviation, and other measures of dispersion for a data set.	1a	1, 5, 11	A, B
9	Analyze relationships in data using correlation and basic regression methods and interpret the results.	1a, 2	1, 5, 11	A, B

***Teaching Methods 1:Lecture, 2:Discussion, 3:Experiment, 4:Simulation, 5:Question-Answer, 6:Tutorial, 7:Observation, 8:Case Study, 9:Technical Visit, 10:Problem Solving, 11:Individual Work, 12:Team/Group Work, 13:Brainstorming, 14:Project Design / Management, 15:Report Preparation and/or Presentation**

****Assessment Methods A:Exam, B:Quiz, C:Oral Exam, D:Homework, E:Report, F:Article Examination, G:Presentation, I:Experimental Skill, J:Project Observation, K:Class Attendance; L:Jury Exam**

Main Textbook	Sheldon Ross, A First Course in Probability, Prentice Hall, 7th edition, 2006.
Supporting References	1. J. L. Devore, Probability and Statistics, Thomson Brooks/Cole, 2004. 2. H. Stark and J. W. Woods, Probability and Random Processes with Applications to Signal Processing, Pearson Education, 2002.
Necessary Course Material	Computer, projection

Course Schedule	
1	Set theory
2	Counting techniques
3	Introduction to probability
4	Conditional probability
5	Independence
6	Random variables
7	Continuous random variables
8	Mid-Term Exam
9	Normal random distributions
10	Other continuous distributions
11	Multi-dimensional random variables
12	Independent random variables
13	Mean, median, mode, and other measures of central tendency
14	Standard deviation and other measures of dispersion
15	Correlation and regression
16,17	Final Exam

Calculation of Course Workload			
Activities	**Number**	**Time (Hour)**	**Total Workload (Hour)**
Course Time (number of course hours per week)	14	3	42
Classroom Studying Time (review, reinforcing, prestudy,….)	12	3	36
Homework			
Quiz Exam	3	1	3
Studying for Quiz Exam	3	3	9
Oral exam			
Studying for Oral Exam			
Report (Preparation and presentation time included)			
Project (Preparation and presentation time included)			
Presentation (Preparation time included)			
Mid-Term Exam	1	1	1
Studying for Mid-Term Exam	1	12	12
Final Exam	1	1	1
Studying for Final Exam	1	16	16
		Total workload	**120**
		Total workload / 30	**4**
		Course ECTS Credit	**4**

Evaluation	
Activity Type	**%**
Mid-term	30
Quiz	30
Final Exam	40
Total	100

RELATIONSHIP BETWEEN COURSE LEARNING OUTCOMES AND PROGRAM OUTCOMES (PO)
(5: Very high, 4: High, 3: Moderate, 2: Low; blank: no direct contribution)

No	PROGRAM OUTCOME		Level
1	Engineering Knowledge	a. Knowledge of mathematics.	5
		b. Knowledge of natural sciences.	
		c. Knowledge of fundamental engineering subjects.	
		d. Knowledge of computational methods.	
		e. Ability to apply the knowledge acquired in items a–d to the solution of complex engineering problems related to Software Engineering.	
2	Problem Analysis	Ability to identify, formulate and analyze complex engineering problems related to Software Engineering by applying knowledge of basic sciences, mathematics and engineering while considering the relevant United Nations (UN) Sustainable Development Goals.	2
3	Engineering Design	Ability to design creative solutions (systems, processes, devices or products) to complex engineering problems related to Software Engineering, taking realistic constraints and conditions into account to meet present and future needs.	
4	Use of Techniques and Tools	Ability to select and use appropriate techniques, resources and modern engineering and information technology tools, including prediction and modeling, for the analysis and solution of complex Software Engineering problems, with awareness of their limitations.	
5	Research and Investigation	Ability to use research methods effectively to investigate complex Software Engineering problems, including literature review, design and execution of experiments, data collection, and analysis and interpretation of results.	
6	Global Impact of Engineering Applications	Knowledge of the impacts of Software Engineering applications on society, health and safety, the economy, sustainability and the environment in the context of the UN Sustainable Development Goals; awareness of the legal consequences of engineering solutions.	
7	Ethical Conduct	Awareness of the ethical principles and responsibilities of the Software Engineering profession; commitment to professional conduct, ethical responsibility in engineering applications, non-discrimination, impartiality and inclusion of social diversity.	
8	Individual and Teamwork	Ability to work independently and to contribute effectively as a member or leader of disciplinary and multidisciplinary teams in face-to-face, remote or hybrid environments.	
9	Oral and Written Communication	Ability to communicate clearly and effectively, both orally and in writing, on technical matters while considering differences in the target audience's educational level, language and professional background.	
10	Project Management	Knowledge of Software Engineering project management processes and business practices such as economic feasibility analysis; awareness of entrepreneurship and innovation.	
11	Lifelong Learning	Ability to learn independently and continuously, adapt to emerging technologies, critically evaluate technological changes, access information, follow scientific and technological developments, and engage in continuous self-development with awareness of the need for lifelong learning.	

LECTURER(S)				
Prepared by	Asst. Prof. Dr. Sinem BOZKURT KESER			
Signature(s)				

Date: 09.11.2026