

Software Engineering Department Course Descriptions

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
General English 1	LNG101	1	2	0		2	2
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The course aims to develop students' general English language skills (reading, writing, listening, speaking) at a basic level; to teach grammar rules, vocabulary, and academic English usage.						
Course Content	The course covers foundational English grammar, vocabulary, and skills with a focus on engineering-related topics to build practical communication abilities.						
References:							
<ol style="list-style-type: none"> 1. English for Engineering, Cambridge 2. Worksheets, Lecture Videos, PowerPoint Presentations 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Principles of Ataturk and History of Revolution 1	GNL101	1	2	0	0	2	2
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course is designed to introduce you to modern Turkish history while developing your critical reading and reasoning skills.						
Course Content	This course is the first part of a two-semester course on modern Turkish history. It focuses on understanding the political, social, economic, and cultural foundations of modern Turkey from the early nineteenth century to the end of the First World War. All aspects of Ottoman modernization will be examined, and the Tanzimat and Constitutional periods will be evaluated to understand the transition from empire to republic. While exploring the formation of modern Turkey, the course will also focus on the continuities and ruptures between the Ottoman Empire and republican Turkey.						
References:							
<ol style="list-style-type: none"> 1. Textbook: Erik J. Zürcher, <i>Turkey: A Modern History</i>. London: I. B. Tauris, 2017. 2. Course Materials: Textbook and PowerPoint presentations. 3. Supplementary Resources: PowerPoint presentations and related images uploaded to the internet each week. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Turkish Language 1	GNL105	1	2	0		2	2
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	<ol style="list-style-type: none"> 1. To enable students to understand what language is and to examine the relationship between language and culture. 2. To explore the stages of simplification in the Turkish language. 3. To enable students to express themselves effectively in oral communication. 4. To teach the importance of body language in communication and effective communication methods. 5. To teach Turkish spelling rules and demonstrate the importance of correct use of punctuation marks. 						
Course Content	<ol style="list-style-type: none"> 1. Techniques for writing emails and petitions 2. Definition of language, its features, and types 3. History of the Turkish language, scripts used by Turks 4. The relationship between language and culture 5. History of the Turkish language, stages of simplification in Turkish 6. Current issues in Turkish 7. Spelling rules, punctuation, and expression disorders 8. Effective and eloquent speaking 9. Basic diction rules 10. Effective communication strategies and body language 						
References:	<ol style="list-style-type: none"> 1. Türkçe Sözlük / Prepared by Şükrü Haluk Akalın... [et al.] (2005), 10th edition, Ankara: Turkish Language Association. 2. Türkçe İmla Kılavuzu (2000). Ankara: Turkish Language Association. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Physics 1	FEF111	1	3	0		3	4
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	<p>To introduce the fundamental concepts of motion essential for engineering science and provide the necessary foundation for engineering students.</p> <p>To provide students with a deeper understanding of the fundamental laws and concepts related to natural phenomena.</p> <p>To develop students' problem-solving skills.</p> <p>To strengthen students' creative and systematic thinking abilities.</p>						
Course Content	Physical quantities, units, and dimensions. Motion in one, two, and three dimensions. Newton's laws of motion. Work-energy theorem, linear momentum and collisions, conservation of energy and momentum. Rotational motion, angular momentum, conservation of angular momentum, torque. Static equilibrium.						
References:	<ol style="list-style-type: none"> 1. Course Book: Hugh D.Young / Robert A. Freedman,University Physics with Modern Physics, 13th Edition. ISBN 13: 978-0-321-69686-1 2. Course Materials: Hugh D.Young / Robert A. Freedman,University Physics with Modern Physics, 13th Edition. ISBN 13: 978-0-321-69686-1 3. Supplementary Readings: ohn W. Jewett, Jr / Raymond A. Serway, Physics for Scientists and Engineers, 8 th Edition. ISBN: 978-1-4390-4846-7 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Physics Lab.	FEF121	1	0	2		1	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	To apply theoretical knowledge by conducting current experiments in mechanics and optics. To learn measurement knowledge by using measurement tools. To enhance group work skills.						
Course Content	Mechanics experiments: Measurement and SI unit system. Simple pendulum, spring pendulum, Hooke's Law. Motion in one and two dimensions. Conservation of angular momentum. Interference of light experiment.						
References:	<ol style="list-style-type: none"> 1. Textbook: <i>Physics Laboratory Experiments</i> (Istanbul Commerce University) / <i>Fiziksel Ölçmeler ve Değerlendirilmesi</i> (İ. Eşme) 2. Course Materials: Physics Lab equipment 3. Supplementary Resources: <i>Physics Laboratory Experiments</i> (Istanbul Commerce University) – Provided to students via online open access.. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Mathematical Analysis 1	MAT121	1	4	0		4	7
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The main purpose of this course is to teach the fundamental concepts of Differential and Integral Calculus, namely Functions, Limits, Derivatives, and Integrals.						
Course Content	Functions, Limits, Continuity, Vertical and Horizontal Asymptotes, Derivatives, Applications of Derivatives, Antiderivatives, Fundamental Theorem of Calculus, Integrals.						
References:	<p>Book: Single Variable Calculus, Brooks Cole, J. Stewart. Course Materials: Computer and projector. Supplementary Resources: Weekly Lecture Notes</p>						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Introduction to Software Engineering	SFT101	1	2	0	0	2	4
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course provides a general introduction to software engineering. It introduces concepts such as software processes and agile methods, as well as the fundamental software development activities ranging from initial specification to system maintenance.						
Course Content	Introduction to Software Engineering, Boolean Algebra, Algorithms and Flowcharts, Software Engineering, Operating Systems, Microprocessors, Number Systems, Data Compression, System and Software, Hardware.						
References:	<ol style="list-style-type: none"> Lecture slides and PDFs Presentations 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Programming	BIL111	1	3	1		3.5	6
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to provide students with knowledge of fundamental programming concepts in the C programming language, including algorithms, data types, control structures, and functions.						
Course Content	<ul style="list-style-type: none"> Introduction to programming and algorithm concepts Flowcharts and the basic structure of C programs Variables, data types, and operators Decision structures (if-else, switch) Loops (for, while, do-while) Arrays and character arrays (strings) Functions and parameter passing Pointers and introduction to memory management Structures (struct) and unions Introduction to file operations <p>Basic introduction to object-oriented programming concepts</p>						
References:	<ol style="list-style-type: none"> Projects Weekend assignments Lab assignments Textbook Lecture presentations 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
General English 2	LNG102	2	2	0	0	2	2
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The purpose of this course is to increase students' confidence in basic English reading and comprehension skills, to develop their vocabulary, and to enable them to recognize and use effectively the basic English structures used in everyday life and academic contexts.						
Course Content	<ul style="list-style-type: none"> • Jobs and work-related verbs • Talking about problems • Past Simple tense • School subjects and the human body • Talking about accidents • Buying medicine and health-related expressions • Going to structure • Food and drinks • Countable and uncountable nouns • Ordering at a restaurant • Comparisons (comparatives and superlatives) • Daily routines • Present Perfect tense • Talking about experiences <p>Asking for explanations and making comments</p>						
References:	<ol style="list-style-type: none"> 1. <i>English File (Elementary)</i> Coursebook, Workbook 2. Worksheets, Lecture Videos, PowerPoint Presentations 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Turkish Language 2	GNL106	2	2	0		2	2
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	To focus on research types, the writing process of research texts, research methods and techniques, digital data scanning methods, utilization of libraries, and topics related to the Turkish novel, short story, and poetry						
Course Content	Selected texts from Turkish literature, academic writing and research methods						
References:	<ol style="list-style-type: none"> 1. <i>Türkçe Sözlük</i> / Prepared by Şükrü Haluk Akalın... [et al.] (2005), 10th edition, Ankara: Turkish Language Association. 2. <i>Türkçe İmla Kılavuzu</i> (2000). Ankara: Turkish Language Association. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Principles of Ataturk and History of Revolution 2	GNL102	2	2	0	0	2	2
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	<p>This course aims to examine the Turkish Revolution—the transition process from the Ottoman Empire to the Republic of Turkey—in its historical, political, economic, social, and cultural dimensions, and to evaluate Atatürk's thought, principles, and reforms that formed the current democratic, secular, and social rule of law order in Turkey.</p> <ul style="list-style-type: none"> • Understands the fundamental principles of Turkish foreign policy. • Evaluates the emergence and development directions of foreign policy issues. • Interprets the effects of Turkey's economic, political, and legal developments in relation to one another. • Critiques attitudes aimed at preserving peace. • Adopts the principles of the democratic system. • Conducts modern and science-based evaluations of recent periods. 						
Course Content	<p>This course evaluates Turkish society at the level of contemporary nation-states in parallel with the establishment of the Republic of Turkey, and examines the unique Turkish Revolution experience within the value system that defines the nation-state (chronologically: the classical and post-classical periods of Turkish imperial past, late 18th and 19th centuries, the 1908 Revolution, 1914–1918, 1918–1922, and 1923–1960 periods). It addresses the establishment of the modern Turkish nation-state within the framework of state and societal restructuring, analyzing the stages of change in the Turkish political and social system as a whole. This includes a comprehensive analysis of the phenomenon of political, social, economic, and cultural transformation (at various levels and dimensions), the analysis of internal and external political events, the thoughts and actions of all actors involved, and the evaluation of these historical processes (the establishment processes of the modern Republic of Turkey) as a reference and comparative framework for solving the problems of contemporary Turkish society and politics today. The course emphasizes the importance of reflecting o</p>						
References:	<ol style="list-style-type: none"> 1. <i>Atatürk İlkeleri ve İnkılap Tarihi 2</i>, Prof. Dr. Cezmi Eraslan (2015), Anadolu Üniversitesi Açıköğretim Fakültesi Yayınları. Yayın No: 1988. 2. Instructor's notes 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Physics 2	FEF112	2	3	0	0	3	4
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to introduce the fundamental concepts and applications of electricity and magnetism, including Coulomb's Law, electric field and potential, Gauss's Law, capacitors and their technical applications, DC circuit analysis concepts and applications, basic concepts of magnetism and technical applications, charge-current-magnetic field interactions, induction currents and technical application areas, basic concepts and applications of AC circuit elements, series-parallel RLC circuit analyses and phasor diagrams, Maxwell's equations, and introduction to oscillation equations. It also covers the basic concepts of electricity and magnetism, measurement instruments (ammeter, voltmeter, ohmmeter), multimeter measurements, electrical circuit elements, simple electrical circuits and circuit analyses, electricity and magnetism experiments, electromagnetic induction and its applications, and Maxwell's equations.						
Course Content	<p>Possess up-to-date knowledge, software, theoretical, and practical information in the field of physics; have the level of knowledge to use physics-related resources.</p> <p>Have theoretical knowledge in physics theory topics; acquire analytical thinking skills.</p> <p>Be able to apply theoretical knowledge acquired in physics in practice.</p> <p>Be able to evaluate experimental data as required.</p> <p>Be able to examine concepts and ideas in the field using scientific methods, interpret data, evaluate, and analyze.</p> <p>Gain skills in writing scientific reports, conducting scientific research, making presentations, and scientific inquiry.</p>						
References:	<ol style="list-style-type: none"> 1. <i>Physics for Scientists and Engineers</i> (Serway & Beichner) – Volume 2 (or equivalent title: Physics 2 for Science and Engineering) 2. <i>Fen ve Mühendislik İçin Fizik 2</i> (Serway & Beichner), Translated by Prof. Dr. Kemal Çolakoğlu 3. Instructor's notes 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Physics Lab. 2	FEF122	2	0	2		1	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	To enable students to better understand the topics of Physics II through practical application, and to develop skills in observation, measurement, data recording, defining variables, data interpretation, drawing conclusions, and conducting experiments.						
Course Content	DC circuit analysis, Ohm's Law, Kirchhoff's Laws. AC circuit analyses, RLC circuits. Transformers and electromagnetic induction. Electrical measurement instruments: ammeter, voltmeter, ohmmeter. Oscilloscope.						
References:	<ol style="list-style-type: none"> 1. Textbook: <i>Physics Laboratory Experiments</i> (Istanbul Commerce University) – open source (provided via online open access) 2. Course Materials: Laboratory equipment 3. Supplementary Resources: <i>Physics II</i>, Serway 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Mathematical Analysis 2	MAT122	2	4	0		4	7
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The main purpose of this course is to teach the fundamental concepts of multivariable differential and integral calculus for multivariable functions, including multivariable functions, limits, partial derivatives, multiple integrals, and surface integrals.						
Course Content	Three-dimensional coordinate system, Vector-valued functions and space curves, Multivariable functions, Limits and continuity, Partial derivatives, Maximum and minimum problems, Lagrange multipliers, Double integrals, Double integrals in polar coordinates, Surface area, Triple integrals, Vector fields, Surface integrals, Green's Theorem, Stokes' Theorem, Divergence Theorem.						
References:	<ol style="list-style-type: none"> 1. Textbook: <i>Multivariable Calculus</i>, Brooks Cole, J. Stewart. 2. Course Materials: Computer and projector. 3. Supplementary Resources: Weekly Lecture Notes. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Data Structure	BIL122	2	2	1		2.5	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course introduces fundamental data structures and their applications in problem solving and algorithm development. Students will examine various data structures such as arrays, linked lists, stacks, queues, trees, and graphs. The course focuses on understanding the strengths and weaknesses of different data structures, their memory representations, and related algorithms for searching, sorting, and traversal. Through lectures and laboratory work, students will develop the skills to analyze and implement data structures efficiently.						
Course Content	Data Structures – Introduction Concepts, Arrays, Algorithms, Variables and Data Types, Array Traversal, Searching and Sorting, Linked Lists, Stacks, Queues, Trees, Binary Search Trees, Heap Trees and Huffman Coding Trees, Graphs.						
References:	<ol style="list-style-type: none"> 1. Data Structures and Algorithms Made Easy–by Narasimha Karumanchi 2. Data Structures and Algorithms: Annotated Reference with Examples by Barnett Granville, Del Tongo Luca. 3. Lecture slides and textbook provided in class 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Logic Circuits	EEE203	2	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	To learn the analysis and design of combinational and sequential logic circuits.						
Course Content	Number systems, Arithmetic in binary number system, Boolean Algebra, Gate-level minimization using Karnaugh maps and tabular methods, Logic gate elements, Analysis and design of combinational and sequential logic circuits.						
References:							
<ol style="list-style-type: none"> 1. Logic circuit simulators 2. <i>Digital Design</i> 3. M. Morris Mano and Michael D. Ciletti 4. Lecture notes 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Engineering Mathematics 1	ENG126	3	3	0		3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The purpose of this course is to introduce several important concepts of Linear Algebra, including 1) matrices, 2) determinants, 3) eigenvalues, 4) eigenvectors, 5) vectors, 6) vector spaces, 7) linear transformations, and 8) inner product spaces, and to equip students to use them effectively in other courses and in their profession.						
Course Content	Systems of Linear Equations, Matrices, Linear Transformations, Determinants, Eigenvalues, Eigenvectors, Vector Spaces, Inner Product Spaces.						
References:							
<ol style="list-style-type: none"> 1. Lecture PDFs 2. Instructor's notes 							

Dersin Adı	Kodu	Yarıyıl	Teori Saat/hafta	Uygulama Saat/hafta	Laboratuvar Saat/hafta	Yerel Kredi	AKTS
Numerical Analysis	ENG211	3	3	0		3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The general purpose of this course is to introduce techniques for finding roots, interpolation, numerical differentiation and integration, and solving differential equations using numerical methods.						
Course Content	Introduction to MATLAB. Definition of error. Solution of linear systems: direct methods and iterative methods. Solution of nonlinear equation systems. Interpolation techniques. Numerical differentiation. Numerical integration. Solution of differential equations. Partial differential equations and their classification.						
References:							
<ol style="list-style-type: none"> 1. T. Kotil, O. Yazıcıoğlu, O. Borat, 2018, <i>Sayısal Çözümleme</i>, 2nd Edition, Seçkin Yayıncılık AŞ, Ankara. 2. İrfan Karagöz, <i>Sayısal Analiz ve Mühendislik Uygulamaları</i>, 2nd Edition, Nobel Yayın Dağıtım Ltd. Şti., 2008. Instructor's notes. 3. Lecture Notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Discrete Mathematics	BIL317	3	3	0		3	4
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The purpose of this course is to provide the fundamental mathematics required for Software Engineering courses.						
Course Content	Explains propositions, logic, sets, and sequences. Functions and relations are discussed. Induction and proof methods are presented. The analysis of algorithms is covered in detail. Number theory is introduced to solve linear congruences, the Euclidean algorithm is explained, and the Chinese Remainder Theorem is presented. Starting from permutations and combinations, conditional probability and Bayes' Theorem are discussed. The calculation of binomial coefficients is explained. Sequential circuit theory and Finite State Machines (FSMs) are covered, and languages and grammars are taught in detail.						
References:							
<ol style="list-style-type: none"> 1. Kenneth H. Rosen, <i>Discrete Mathematics and Its Applications</i>, McGraw-Hill 2. Instructor's lecture notes 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Requirements Analysis	SFT241	3	3	1		3	4
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to introduce languages and models for representing requirements. It covers techniques for analyzing and validating needs, goals, and use cases; determining and analyzing requirements for various system types; requirement documentation standards; requirements in the context of agile processes; and requirement management, including handling requirement changes.						
Course Content	The course introduces students to the requirements definition phase of the software development lifecycle. Methodologies for analyzing system specifications, constraints, and assumptions related to creating appropriate definitions of the desired system behavior and qualities will be discussed.						
References:							
<ol style="list-style-type: none"> 1. Weekly Project Evaluations and Feedback 2. Instructor's lecture notes. 3. Axel van Lamsweerde, <i>Requirements Engineering: From Systems Goals to UML Models to Software Specifications</i>, ISBN: 978-0-470-01270-3, 2009, Wiley. 4. Dean Leffingwell & Don Widrig, <i>Managing Software Requirements: A Use Case Approach</i>, ISBN: 978-0-321-12247-X, 2003, Addison-Wesley. 5. Kent J. McDonald, <i>Beyond Requirements: Analysis with Agile Mindset</i>, ISBN: 978-0-321-83455-3, 2015, Addison-Wesley. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Algorithms	BIL231	3	2	1		2.5	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	To understand and be able to apply the algorithms used in computer science as well as how to analyze them.						
Course Content	Introduction to Algorithms, Complexity Analysis in Algorithms, Time Complexity, Space Complexity, Recursive Algorithms, Sorting Algorithms, Quick Sort, Merge Sort, Algorithm Design Strategies, Divide and Conquer Algorithms, Master's Theorem, Analysis of Recursive Algorithms, Greedy Algorithms, Dynamic Programming Strategy, Top-Down and Bottom-Up Approaches.						
References:							
<ol style="list-style-type: none"> a. Introduction to Algorithms, fourth edition Thomas H. Cormen b. Algorithms Robert Sedgewick ve Kevin Wayne 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Object Oriented Programming	BIL233	3	3	1		3.5	7
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	To introduce the concepts of object-oriented programming and enable the development of object-oriented solutions for software problems.						
Course Content	Classes and Objects, Fundamentals of Programming in Java, Data Types, Methods, Data Abstraction and Encapsulation, Strings, Arrays, Classes and Inheritance, Subclasses, Abstract Classes and Methods, Reusability in Software, Object-Oriented Thinking, Polymorphic Variables, Garbage Collection, Types of Polymorphism, Overriding, Overloading, Abstract Methods, Interfaces, JavaFX/Swing Applets (exception handling/sending graphics), JavaFX/Swing Applets (collection classes, generic classes), JavaFX/Swing Applets (event-driven programming), Working with Collections, Reflection, Object-Oriented Design, Threads.						
References:	<ol style="list-style-type: none"> 1. How to Program Java Deitel. 2. Java : how to program / Deitel, H. M. ; P.J. Deitel, Oracle Academy Notes 3. Instructor's notes. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Üniversite Seçmeli	REKXXX	4	3	0		3	2
Prerequisite(s)							
Course Type	Elective						
Course Objectives							
Course Content							
References:	<ol style="list-style-type: none"> 1. Related coursebooks 2. Instructor's notes. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Engineering Mathematics 2	ENG227	4	3	0		3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The main objectives of this course are: 1. To learn the types of differential equations and how to solve them. 2. To be able to apply differential equations to problems in engineering.						
Course Content	Linear equations, separable equations, exact differential equations, integrating factor, substitution method, linear equations with constant coefficients, method of variation of parameters.						
References:							
1. <i>A First Course in Differential Equations</i> , Pearson Education, Dennis G. Zill. 2. Instructor's Notes							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Probabilistic Methods in Engineering	ENG231	4	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to teach students the fundamental principles of probability theory, statistical methods, and stochastic processes. It provides students with practical skills covering topics such as modeling random variables, probability distributions, statistical inference, and confidence intervals. Additionally, the course emphasizes how probability and statistical methods are used in the analysis of engineering problems and decision-making processes. It equips students with the ability to handle uncertainties encountered in engineering applications and to perform risk analysis.						
Course Content	Counting Techniques: multiplication rule, permutations, combinations. Probability Concepts: sigma-algebra, probability axioms, conditional probability, Bayes' theorem. Random Variables: distribution function, probability function, Chebyshev's inequality. Discrete and Continuous Distributions: uniform distribution, Bernoulli distribution, Poisson distribution, geometric distribution, hypergeometric distribution, normal distribution, exponential distribution, gamma distribution, beta distribution. Moment Generating Functions. Decision Theory. Estimation Concepts. Hypothesis Testing. Nonparametric Tests. Correlation and Regression						
References:							
1. F. Akdeniz, <i>Olasılık ve İstatistik</i> , 11th Edition, Nobel Kitabevi, Adana, 2004. 2. W.W. Hines and D.C. Montgomery, <i>Probability and Statistics in Engineering and Management Science</i> , 2nd Edition, John Wiley & Sons, NY, USA, 1980. 3. Instructor's Notes							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Design and Analysis	SFT228	4	3	1	0	3,5	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course covers Software Architecture and Software Design, Object-Oriented Design, fundamental software development principles, and Software Design Patterns. Students will be informed about software development models, software development and design tools, as well as the application of software development principles and design patterns. Through a term project, the course aims to enable students to gain practical experience—in addition to theoretical knowledge—in making valid engineering decisions when facing realistic problems, as well as in design, implementation, and testing.						
Course Content	Software architecture, software architecture patterns, implementation of architectural patterns, fundamental software design principles, software design patterns.						
References:							
<ol style="list-style-type: none"> 1. <i>Fundamentals of Software Architecture: An Engineering Approach</i>, Marc Richards et al., 2020, O'Reilly. 2. Instructor's notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Programming Languages	BIL234	4	2	1		2.5	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to provide students with knowledge of general programming paradigms through programming languages, to enable understanding of how source code is read and interpreted by compilers or interpreters, to guide the selection of appropriate programming languages, and to further improve programming skills.						
Course Content	Introduction to Programming Languages, Evolution of Major Programming Languages, Definition of Syntax and Semantics, Lexical and Syntax Analysis, Name Binding and Scope, Data Types, Expressions and Assignment Statements, Control Structures at Expression Level, Subprograms, Implementation of Subprograms, Abstract Data Types and Encapsulation, Compatibility.						
References:							
<ol style="list-style-type: none"> 1. Internet resources and materials provided in class 2. <i>Concepts of Programming Languages</i>, R. Sebesta, Pearson 3. Lecture slides and textbook provided in class 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Database systems	SFT242	4	3	1		3.5	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to enable students to learn relational database systems and develop applications.						
Course Content	<p>Can transform real-world problems into a mini-world model for a database</p> <p>Can create and design entity-relationship diagrams</p> <p>Can effectively use relational algebra and SQL statements (DDL, DML, DCL)</p> <p>Understands how data is stored in a database</p> <p>Knows how concepts such as data, information, and knowledge are obtained through databases</p> <p>Gains project development skills</p>						
References:							
<ol style="list-style-type: none"> 1. Çağıltay, N. E., Gül Tokdemir, <i>Veritabanı Sistemleri Dersi</i>, Ada Matbaacılık, 2010 2. Instructor's notes, internet resource 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Signals and Systems	EEE206	4	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The purpose of this course is to provide an understanding of the fundamental concepts related to Signals and Systems encountered in engineering. The main focus is on methods for the characterization and analysis of continuous-time and discrete-time signals and systems. Students will learn some transformation techniques (Laplace transform, Z-transform, and Fourier transform) that are useful for understanding digital communication systems, satellite and mobile communications, digital signal processing, and digital image processing.						
Course Content	Signals and their properties; signal processing, sampling, signal power and energy, convolution; systems and their properties, time-domain and frequency-domain analysis of continuous-time periodic signals, Fourier series and Fourier transforms, time-domain and frequency-domain analysis of discrete-time signals, Fourier series and Fourier transforms, Laplace transform and inverse Laplace transform, system frequency response, Z-transform.						
References:							
<ol style="list-style-type: none"> 1. <i>Signals and Systems</i>, Hwei P. Hsu, 2011 2. <i>Signals and Systems</i>, A. V. Oppenheim, A. S. Willsky, 1996 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Summer Internship 1	ENG301	4	0	0	0	0	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The main purpose of this course is to enable students to reinforce the theoretical knowledge acquired throughout their software engineering education through practical applications in a professional work environment; and to provide them with experience in workplace organization, project management processes, teamwork, and professional ethics.						
Course Content	Within the scope of this internship: examination of the institution's hardware and software infrastructure, observation of the software development life cycle (SDLC), participation in analysis, design, coding, or testing processes in an active project, and presentation of the performed work in the format of a technical report.						
References:							
<ol style="list-style-type: none"> 1. University Internship Directive and Implementation Principles. 2. Summer Internship Guide (Department Internship Guide). 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Microprocessor Systems	BIL331	5	3	0	0	3	4
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to enable students to understand microprocessor system architectures, comprehend the structure of a selected processor as an example, and gain the ability to use assembly language.						
Course Content	Central Processing Unit (CPU): CPU registers, status flags, arithmetic and logic unit. Microinstructions, microprograms, and control unit. CPU pins and signals, Microprocessor Instruction Set, Programming of Microcomputers, Interfacing of Microcomputers, Memory addressing methods, Microcomputer system design. Applications.						
References:							
<ol style="list-style-type: none"> 1. Brey, B. B., 1984, <i>Microprocessor/Hardware Interfacing and Applications</i>, Merrill, 414 p. 2. <i>Microcomputers and Microprocessors: The 8080, 8085, and Z80 Programming, Interfacing, and Troubleshooting</i>, Prentice-Hall, 670 p. 3. Yazıcı, R., 1998, <i>Mikrobilgisayar Donanım ve Yazılımı</i>, KTÜ Yayınları, Trabzon, 345 p. 4. 8086 Microprocessor notes; Yazıcı, R., 1998, <i>Mikrobilgisayar Donanım ve Yazılımı</i>, KTÜ Yayınları, Trabzon. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Cloud Computing and Distributed Systems	SFT335	5	2	2		3	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to teach students the fundamental principles and architectural models of distributed systems, while providing theoretical and practical competencies in virtualization technologies, cloud service models (IaaS, PaaS, SaaS), and modern cloud platforms.						
Course Content	Distributed system architectures, inter-process communication and synchronization, virtualization technologies (VMs and containers), cloud computing service models (SaaS, PaaS, IaaS), microservices architecture, Docker and Kubernetes technologies, cloud security, and big data processing infrastructures						
References:							
<ol style="list-style-type: none"> 1. Tanenbaum, A. S., & Van Steen, M. (2017). <i>Distributed Systems: Principles and Paradigms</i>. Pearson. 2. Marinescu, D. C. (2017). <i>Cloud Computing: Theory and Practice</i>. Morgan Kaufmann. 3. Erl, T., & Puttini, R. (2013). <i>Cloud Computing: Concepts, Technology & Architecture</i>. Prentice Hall. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Database Management	SFT351	5	3	0	0	3	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The purpose of this course is to provide students with sufficient competency regarding the principles and topics related to database management systems.						
Course Content	Design algorithms and applications; entity-relationship model; hierarchical, network, and relational databases; query languages; relational algebra and calculus; data dependencies; normal forms; consistency, integrity, and security issues; term project.						
References:							
<ol style="list-style-type: none"> 1. Raghu Ramakrishnan, Johannes Gehrke, <i>Database Management Systems</i>, 4th Edition, McGraw-Hill, 2015. 2. Lecture Notes 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Modelling and Simulation	MEE321	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	<p>To form the mathematical model of a dynamic system containing a control system</p> <p>To simplify mathematical models to linear, time-invariant systems through linearization and/or block diagram reduction</p> <p>To use time-domain performance criteria to design single-input single-output (SISO) control systems that satisfy the specified time response, accuracy, and stability requirements</p> <p>To create and use frequency response tools to design single-input single-output (SISO) control systems that achieve the specified time response and stability requirements</p> <p>To use MATLAB to design, simulate, and analyze the response of controlled dynamic systems</p>						
Course Content	Introduction to Control Systems, Review of Laplace Transform, Transfer Functions, Block Diagrams, State-Space Representation, Time Response of First-Order Systems, Time Response of Second-Order Systems, Stability, Basic Controllers (P, PD, PI, PID), MATLAB Review, Root Locus Techniques, Frequency Domain: Bode Plots, State-Space Methods, Observability and Controllability						
References:							
<ol style="list-style-type: none"> 1. Textbook: Norman S. Nise, <i>Control Systems Engineering</i>, 6th edition, John Wiley and Sons Inc., 2011. 2. Course Materials: Textbooks, Presentations, MATLAB Codes 3. Supplementary Resources: Gene Franklin, David Powell, Abbas Emami-Naeini, <i>Feedback Control of Dynamic Systems</i>, Prentice Hall; Benjamin Kuo, <i>Automatic Control Systems</i>, Prentice Hall 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Web Programming for Software Engineers	SFT 333	5	2	2	0	3	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The Web Programming for Software Engineers course aims to equip students with the fundamental concepts, tools, and best practices for the design, development, and deployment of web applications in accordance with software engineering discipline. Through applied projects and code reviews, the course targets the development of industry-ready, sustainable, and scalable web application development skills, teamwork competencies, and professional approaches aligned with software engineering principles.						
Course Content	The course content is structured to cover the core building blocks of web programming: semantic markup with HTML, forms and multimedia usage; responsive layout with CSS, Flexbox/Grid, and style management; fundamentals of JavaScript language and DOM manipulation; server-side development with ASP.NET Core, MVC/Razor Pages, and data access with Entity Framework Core. These practical topics aligned with software engineering are addressed throughout the course through applied examples and projects.						
References:							
<ol style="list-style-type: none"> 1. Instructor's Notes 2. Lecture presentations and PDFs 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Operating Systems	BIL342	6	3	0		3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to teach in detail the operating systems, which are among the most fundamental components of a computer. Developed software continuously interacts with the operating system to produce its outputs. The course targets enabling students to reach a level where they can design and code the subsystems of an operating system.						
Course Content	This course provides a comprehensive introduction to operating systems, one of the most fundamental components of computer systems. It covers the structure and functions of operating systems, process and thread management, synchronization and communication mechanisms, CPU scheduling algorithms, deadlock problems and their solutions, memory management (paging, segmentation, virtual memory), file systems, access and protection mechanisms, input/output systems, and disk scheduling.						
References:							
<ol style="list-style-type: none"> 1. Textbook: <ol style="list-style-type: none"> a. Abraham Silberschatz, <i>Operating System Concepts</i> (5th, 6th, 7th, or 8th edition) b. Andrew S. Tanenbaum, <i>Modern Operating Systems</i> (latest edition) 2. <i>Operating System Concepts</i>, Abraham Silberschatz (5th, 6th, 7th, or 8th edition) 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Artificial Intelligence	BIL353	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to introduce students to the fundamental concepts, methods, and algorithms of Artificial Intelligence (AI). It covers intelligent systems, rule-based and decision systems, heuristic problem solving, knowledge representation, graph and tree structures, search algorithms, game theory and minimax, optimization techniques, genetic algorithms, ant colony optimization, fuzzy logic, artificial neural networks, particle swarm optimization, and programming in Prolog for AI applications.						
Course Content	Artificial Intelligence concepts and their applications in engineering Artificial Intelligence algorithms and various applications Graphs and search algorithms Algorithm and game theory Semantic and syntactic algorithms (synthesis and application) Basic operations in Prolog programming language						
References:							
<ol style="list-style-type: none"> 1. Nabiyev, Vasif V., <i>Yapay Zeka: Problemler, Yöntemler, Algoritmalar</i> 2. Braga, Newton C., <i>Robotics, Mechatronics, and Artificial Intelligence: Experimental Projects – lecture notes</i> 3. Instructor's notes 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Validation and Testing	SFT320	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to teach the concepts of verification and validation in order to ensure quality assurance during the software development process, and to equip students with the competence to develop error-free software by using testing processes, techniques (white-box and black-box testing), and automation tools.						
Course Content	Software Testing Life Cycle (STLC), Static and Dynamic Testing, Black-Box and White-Box Techniques, Unit Testing, Integration and System Testing, Regression Testing, Introduction to Test Automation (Selenium etc.), Defect Reporting and Tracking Processes.						
References:							
<ol style="list-style-type: none"> 1. Patton, R. (2005). <i>Software Testing</i>. Sams Publishing. 2. Sommerville, I. (2016). <i>Software Engineering</i> (Chapter: Software Testing). Pearson. 3. Naik, K. & Tripathy, P. (2008). <i>Software Testing and Quality Assurance</i>. Wiley. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Machine Learning	BIL344	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to present the fundamental concepts and methods in the field of machine learning. It introduces basic machine learning algorithms, techniques, and applications, along with general problems related to the analysis and processing of datasets.						
Course Content	Fundamental Concepts, Data Collection Methods and Ethics: HTML parsing, web scraping, Data Repository, Sampling Methods, Bayesian Inference, Ensemble Models—Random Forest, Boosting Regression, Combating Overfitting: Ridge Regression, Tikhonov Regularization, Support Vector Regression, Midterm Exam, Dimensionality Reduction: Singular Value Decomposition, Nonlinear Dimensionality Reduction: Kernel PCA, Locally Linear Embedding, ISOMAP, Semi-Supervised Learning, Active Learning, Reinforcement Learning, Transfer Learning.						
References:							
<ol style="list-style-type: none"> 1. <i>Introduction to Machine Learning</i>, Fourth Edition (Adaptive Computation and Machine Learning series), Ethem Alpaydm, The MIT Press, Fourth Edition. 2. Instructor's notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Computer Architecture	SFT352	6	4	0	0	4	7
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to teach the fundamental components, organization, and working principles of computer systems, while providing a theoretical foundation in instruction set architecture (ISA), central processing unit (CPU) design, memory hierarchy, and performance analysis.						
Course Content	Introduction to Computer Organization, Number Systems and Computer Arithmetic, Boolean Algebra and Logic Circuits, Instruction Set Architecture (ISA – RISC/CISC), Fundamentals of Assembly Language, Processor Datapath and Control Unit Design, Pipelining Technique, Memory Hierarchy (Cache, Virtual Memory), and Input/Output Systems.						
References:	<ol style="list-style-type: none"> 1. Patterson, D. A., & Hennessy, J. L. (2020). Computer Organization and Design: The Hardware/Software Interface (RISC-V Edition). Morgan Kaufmann. 2. Stallings, W. (2019). Computer Organization and Architecture: Designing for Performance. Pearson. 3. Harris, D., & Harris, S. (2012). Digital Design and Computer Architecture. Morgan Kaufmann. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Summer Internship 1	ENG302	6	0	0	0	0	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	The purpose of this course is to help students discover their areas of interest based on the education they have received throughout their engineering studies, prepare them for professional life, and provide an opportunity to apply the theoretical knowledge they have learned.						
Course Content	To enable students to reinforce the knowledge and experience they have gained during their education by observing and participating in hierarchical responsibilities, relationships, organizational structure, and work discipline in the workplaces where they perform their duties; by closely monitoring the functions performed by sector employees and increasing their knowledge and experience related to their diploma programs. To enable them to use the theoretical knowledge they have acquired and transfer it to practice. To develop the habit of working harmoniously with the assigned personnel of the institution where they conduct their workplace training and establishing good communication with other relevant parties (customers or other institutions) connected to the business. To enable them to follow and recognize the technological developments occurring in the sector.						
References:	<ol style="list-style-type: none"> 1. University Internship Directive and Implementation Principles. 2. Summer Internship Guide (Department Internship Guide). 3. Sommerville, I. (2015). <i>Software Engineering</i>. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Security	BIL469	7	2	2	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	Introducing students to basic concepts of software safety and security, deepen's students' understanding of software vulnerabilities, and encouraging students to practice analysis and verification of software safety.						
Course Content	Significance of software safety and security; various facets of security in cyber-physical and computer systems; threat modeling for software safety and security; and categorization of software vulnerabilities. Software analysis and verification: mathematical foundations, data structures and algorithms, program comprehension, analysis, and verification tools; automated vs. human-on-the-loop approach to analysis and verification; and practical considerations of efficiency, accuracy, robustness, and scalability of analysis and verification. Case studies with application and systems software; evolving landscape of software security threats and mitigation techniques. Understanding large software, implementing software analysis and verification algorithms.						
References:	<ol style="list-style-type: none"> 1. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson. 2. Pfleeger, C. P., & Pfleeger, S. L. (2015). Security in Computing. Prentice Hall. 3. OWASP (Open Web Application Security Project) Developer Guide & Testing Guide. 4. Suhel Ahmad Khan, Rajeev Kumar, Raees Ahmad Khan, Software Security: Concepts & Practices, First edition published 2023 by CRC Press 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Computer Networks	BIL441	7	3	0		3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	i. Understand the fundamental concepts of data communications, including signal transmission, encoding, and modulation. ii. Gain knowledge about the OSI and TCP/IP network models and their layers. iii. Learn about various network technologies such as LANs, WANs, multiplexing, and switching. iv. Develop an understanding of data link layer protocols, error detection and correction mechanisms, and medium access control methods. v. Understand Internet Protocol (IP) addressing, routing, and packet forwarding. vi. Explore transport layer protocols (TCP, UDP) and their roles in reliable data transmission. vii. Become familiar with application layer services, including DNS, email, and the World Wide Web. viii. Understand network security principles, including cryptography, VPNs, and firewalls.						
Course Content	This course provides an in-depth introduction to the fundamental concepts and principles of data communications. It covers key topics such as network models, physical and data link layers, switching, multiplexing, error detection, and network protocols. Students will explore the layers of the OSI and TCP/IP models, focusing on the physical transmission of data, reliable communication protocols, and the infrastructure that supports global communication networks. The course also addresses emerging technologies, multimedia communication, and network security to prepare students to understand and analyze modern data communication systems.						
References:							
<ol style="list-style-type: none"> 1. Tanenbaum, Andrew S., <i>Computer Networks</i>, 5th Edition. 2. Forouzan, Behrouz A., <i>Veri İletişimi ve Ağ Oluşturma</i>, 4. Baskı. 3. Stallings, William, <i>Veri ve Bilgisayar İletişimi</i>. 4. Tanenbaum, Andrew S., <i>Bilgisayar Ağları</i>, 5. Baskı. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Project Management	SFT461	7	3	1		3.5	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course introduces students to various software project management concepts, with a primary focus on project analysis, scheduling, resource allocation, risk analysis, monitoring and control, and software configuration management. The main objective of the course is to provide students with knowledge of different software project management concepts so that they can understand and manage a wide range of projects, including highly risky and innovative ones, by using diverse project management skills.						
Course Content	Definition of a Software Project (SP), classification of projects, project management life cycle; project analysis: strategic assessment, technical assessment, economic analysis; activity planning and scheduling: Work Breakdown Structure, bar charts, network planning models; Critical Path Method (CPM), Program Evaluation and Review Technique (PERT), Precedence Diagramming Method (PDM), project time reduction; risk management: nature and identification of risk, risk analysis, Z-values; resource allocation: determining resource requirements, resource leveling; midterm exam; monitoring and control: introduction, data collection, visualizing progress, earned value analysis; managing contracts and people; software project pricing; managing software development projects: Waterfall, V-model etc.; managing software development projects: open source development practices; development and management standards.						
References:	<ol style="list-style-type: none"> Information Technology Project Management: Providing Measurable Organizational Value Jack Marchewka Wiley; 5th edition Instructors Notes 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
System Analysis and Design	SFT312	8	2	2		3	3
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	Students will acquire the ability to perform analysis and design of a software process, including defining scope, problem definition, meeting system requirements, data collection, object and process modeling, and generating alternative solutions, in order to apply feasibility analysis during the decision-making phase.						
Course Content	The course content consists of the fundamental concepts of systems analysis and design, components of information systems, methods for developing information systems, project management, systems analysis approaches, scope definition phase, problem analysis phase, requirements analysis phase, preparation of use-case reports, data modeling and analysis, process modeling, feasibility analysis, and preparation of system proposal reports.						
References:	<ol style="list-style-type: none"> Eric J. Braude and Michael E. Bernstein, Software Engineering: Modern Approaches 2ed, John Wiley & Sons, 2011, ISBN 978-0-471-69208-9 Alcantara ve Becker, Systems Analysis and Design in a Changing World (5th Edition, 2016) Dennis, Wixom ve Roth, Systems Analysis and Design (6th Edition, 2018): 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Graduation Project	ENG402	8	0	4		2	7
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to enable students to acquire knowledge and skills in project planning and management, working in interdisciplinary teams, creating a product in accordance with correct engineering design principles, verification and integration of system components, report writing, and presentation.						
Course Content	Students will implement their proposed solutions through project proposals. In the middle of the semester, a written progress report will be prepared along with a related presentation. At the end of the semester, a final report on the working product will be prepared and a final presentation will be delivered.						
References:							
<ol style="list-style-type: none"> 1. Johnston, Eric R. et al., <i>Programming Quantum Computers: Essential Algorithms and Code Samples</i>, O'Reilly, 2019. 2. Lipton, Richard J. and Regan, Kenneth W., <i>Introduction to Quantum Algorithms via Linear Algebra</i>, Second Edition, The MIT Press, 2021. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Career Planning and Entrepreneurship	GNL450	8	2	0	0	2	0
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course examines the complex and interconnected challenges of providing energy and freshwater to society, treating wastewater, and maintaining environmental cleanliness. These issues have led to increased social and political awareness regarding national and global sustainable development policies in many countries. Discussions on the technological and political dimensions of these crises are frequently observed across various media channels. Although each crisis has distinct causes and consequences, the sustainable provision of water and food, wastewater treatment, and environmental protection are inherently linked and challenging tasks. Understanding the national and global interconnections shaping these linkages requires a deep and holistic comprehension of the political, socioeconomic, and cultural factors influencing them. Within the scope of this course, the historical and contemporary roles of experts in policy, technology, and science in addressing the intricate relationships among energy and freshwater supply, wastewater treatment, and pollution prevention will be analyzed.						
Course Content	Provision of energy and potable water to society, wastewater treatment, environmental protection during these processes, and the resulting social and political awareness influencing policymakers and administrators.						
References:							
<ol style="list-style-type: none"> 1. Bozdoğan, D. (Ed.). (2021). <i>Kariyer Planlama</i>. TOGÜ KARMER. 2. Güzelyurt, L., & Uzun, C. (2021). <i>Kariyer Gelişimi ve Öğrenme Yolculuğu</i>. Ekin Basım Yayın. 3. Ortaylı, İ. (2019). <i>Bir Ömür Nasıl Yaşanır</i>. Kronik. 4. Taş, A., & Aytac, T. (Eds.). (2021). <i>Kariyer Planlama ve Geliştirme</i>. Anı Yayıncılık. 5. Tutar, H. (Ed.). (2020). <i>Kariyer Planlaması ve Yönetimi</i>. Seçkin Yayıncılık. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Cyber Security and Ethics	SFT444	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Compulsory						
Course Objectives	This course aims to examine the ethical, legal, and social issues arising from the use of information technologies; to analyze the individual and societal impacts of cybersecurity breaches; and to foster professional awareness and sense of responsibility regarding engineering ethics codes (ACM/IEEE), intellectual property rights, personal data protection (KVKK/GDPR), and cybercrimes.						
Course Content	Introduction to information ethics and ethical theories, Professional ethics codes (ACM/IEEE Software Engineering Code of Ethics), Types of cybercrimes and legal sanctions, Intellectual property rights and software licensing, Privacy and Personal Data Protection (KVKK & GDPR), Cyber warfare and cyberterrorism, Social engineering and the human factor, Information security management and standards (ISO 27001), Concepts and legal boundaries of ethical hacking (Ethical Hacking)						
References:							
<ol style="list-style-type: none"> 1. Quinn, M. J. (2019). <i>Ethics for the Information Age</i> (8th Edition). Pearson. 2. Whitman, M. E., & Mattord, H. J. (2018). <i>Principles of Information Security</i>. Cengage Learning. 3. Spinello, R. A. (2020). <i>Cyberethics: Morality and Law in Cyberspace</i>. Jones & Bartlett Learning. 4. Law No. 6698 on the Protection of Personal Data (KVKK) and related legislation. 							

ELECTIVE COURSES

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Quality and Standarts	SFT---	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course introduces the concepts, metrics, and models in software quality assurance. It covers the components of software quality assurance systems before, during, and after software development. The course provides a framework for software quality assurance and discusses individual components of this framework such as planning, reviews, testing, configuration management, and so on. It also discusses metrics and models for software quality as a product, in-process quality, and maintenance quality. In this course, students will develop an understanding of software quality and approaches to guarantee software quality.						
Course Content	Software quality, software defects, causes of software defects, software quality assurance, Software quality factors, product operation and revision quality factors, Components of the software quality assurance system, components of the software project life cycle, Pre-project components, elements of the quality plan, SQA components in the project life cycle, SQA defect removal effectiveness and cost, Reviews, formal design reviews (DRs), Software testing and strategies, white-box testing, black-box testing, Midterm Exam, Assuring the quality of software maintenance components, Configuration management, software change control, document control, Components of software quality management, project progress control, Software quality metrics, process metrics, product metrics, Software quality costs, Quality management standards.						
References							
<ol style="list-style-type: none"> 1. Nabyev, Vasif V., <i>Yapay Zeka: Problemler, Yöntemler, Algoritmalar</i> 2. Braga, Newton C., <i>Robotics, Mechatronics, and Artificial Intelligence: Experimental Projects – lecture notes</i> 3. Instructor's notes 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
System Programming	BIL469	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course introduces the structure of the Linux Kernel, the layers between computer hardware architecture and the user interface along with their programming, the use of system calls, examination of kernel structures from a security perspective, device driver programming, memory management, and parallel port programming.						
Course Content	<p>Can examine the concepts necessary for programming computer operating systems.</p> <p>Can understand common operating systems and perform comparative analyses by recognizing their differences.</p> <p>Can provide communication between the computer's hardware and software components.</p> <p>Can perform programming on basic system kernel structures.</p> <p>Can recognize the programming steps of device structures.</p> <p>Can attain expert-level knowledge regarding module programming.</p>						
References:	<ol style="list-style-type: none"> 1. Kerrisk, Michael, <i>The Linux Programming Interface: A Linux and UNIX System Programming Handbook</i>, 5th Edition, No Starch Press. 2. Instructor's notes. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Deep Learning	SFT310	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to present the mathematical, statistical methods, and computational challenges required to create stable representations for high-dimensional data such as images, text, and other data types. Selected topics in deep learning are examined by discussing the latest models in both supervised and unsupervised learning.						
Course Content	History of Deep Learning, Supervised, Unsupervised, Semi-Supervised Learning, Lifelong/Continual Learning, Neural Network Architectures and Tools: Encoding Multi-Class Problems, PyTorch, TensorFlow, Keras, Backpropagation for Deep Models: Chain Rule, Matrix-Vector Multiplication, Optimization: Nesterov Accelerated Gradient Descent, Stochastic Gradient Descent, AdaGrad, RMSProp, Adam, Convolutional Neural Networks: LeNet, AlexNet, Convolutional Neural Networks: VGGNet, GoogLeNet, ResNet, Midterm Exam, Bias-Variance Tradeoff, L1-L2 Regularization, Data Augmentation, Dropout, Sequential Learning Problems: Backpropagation Through Time (BPTT), Truncated BPTT, Long Short-Term Memory Cells (LSTMs), Gated Recurrent Units (GRUs), Autoencoders: Denoising Autoencoders, Object Detection, Object Segmentation, R-CNN, Fast R-CNN, Faster R-CNN, YOLO.						
References:							
<ol style="list-style-type: none"> 1. Goodfellow, Ian et al., <i>Deep Learning</i> (Adaptive Computation and Machine Learning series), MIT Press. 2. Instructor's notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Computer Vision	SFT302	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course provides a comprehensive introduction to computer vision. Topics include camera models, multi-view geometry, reconstruction, some low-level image processing techniques, and high-level vision tasks such as image classification and object detection.						
Course Content	Introduction to Computer Vision, Image Filters, Image Resampling, Edge Detection: RANSAC Model, Corner Detection: Local Features and Harris Model, Feature Invariance, Feature Descriptors and Matching, Midterm Exam, Transformations, Camera Models, Stereo Models, Object Recognition, Pose Estimation, Motion Capture.						
References:							
<ol style="list-style-type: none"> 1. Goodfellow, Ian et al., <i>Deep Learning</i> (Adaptive Computation and Machine Learning series), MIT Press. 2. Instructor's notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Safety	SFT313	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to introduce students to the fundamental concepts of software security and safety, deepen their understanding of software vulnerabilities, and encourage them to apply methods for software security analysis and verification.						
Course Content	Importance of software security and safety; various dimensions of security in cyber-physical systems and computer systems; threat modeling for software security and safety; classification of software vulnerabilities. Software analysis and verification: mathematical foundations, data structures and algorithms, program comprehension, analysis and verification tools; comparison of automated approaches versus human-in-the-loop approaches in analysis and verification; practical considerations such as efficiency, accuracy, robustness, and scalability of analysis and verification. Case studies through application software and system software; the continuously evolving nature of software security threats and mitigation techniques. Understanding large-scale software systems, application of software analysis and verification algorithms.						
References:							
<ol style="list-style-type: none"> 1. Instructor's notes 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Information Systems and Design	SFT317	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to introduce students to the role of Information Systems (IS) within organizational structures and to explain how these systems are created, used, and maintained.						
Course Content	Enterprise systems and integration, gaining competitive advantage through information systems (IS), business process reengineering (BPR) and change management, structured approaches to information systems development, project management, e-commerce, mobile computing and digital economy, front-end web development, database design and modeling, decision support systems, information security and privacy, as well as the impacts of information systems on individuals, businesses, and society.						
References:							
<ol style="list-style-type: none"> 1. Pearlson, Keri E., Saunders, Carol S., Galletta, Dennis F., <i>Managing and Using Information Systems: A Strategic Approach</i>, Wiley, 7th Edition. 2. Instructor's notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
User Interface, Experience and Design	SFT315	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to equip students with the competence to create efficient and effective user interfaces for interactive technologies such as web applications and mobile applications.						
Course Content	Overview of user interface design. Evaluation and testing of user interfaces. Review of analysis principles using object-oriented programming, object-oriented design, and UML in the context of user interface design. Design of windows, menus, and commands. Development of web-based and Windows-based user interfaces. Event-driven programming. Introduction to frameworks and APIs for building user interfaces.						
References:							
<ol style="list-style-type: none"> 1. Stone, Debbie, Jarrett, Caroline, Woodroffe, Mark, Minocha, Shailey, <i>User Interface Design and Evaluation</i> (Interactive Technologies), Morgan Kaufmann Publishers. 2. Johnson, Jeff, <i>Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines</i>, Morgan Kaufmann Publishers, 3rd Edition. 3. Instructor's notes. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Quantum Computing	SFT409	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to provide students with knowledge about quantum processing units (QPUs) and enable them to program them.						
Course Content	What is a QPU, QPU instruction set, Single Qubit, Multiple Qubits, QPU fundamentals, Arithmetic and Logic, Amplitude and Amplification, Quantum Fourier Transform, Quantum Phase Estimation, QPU applications, Real data, Quantum search, Quantum supersampling, Shor's factoring algorithm, Quantum machine learning						
References:							
<ol style="list-style-type: none"> 1. Johnston, Eric R. et al., <i>Programming Quantum Computers: Essential Algorithms and Code Samples</i>, O'Reilly, 2019. 2. Lipton, Richard J. and Regan, Kenneth W., <i>Introduction to Quantum Algorithms via Linear Algebra</i>, Second Edition, The MIT Press. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Research Project Preparation and Management	ENG305	5	2	0	0	2	2
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course serves as preparation for the Capstone Project course that students will take in their final year. The main purpose of the course is to introduce students to scientific research methods before graduation, help them develop project ideas, and equip them with the necessary skills to write project proposals.						
Course Content	Throughout the course, students will be provided with all relevant information regarding the requirements of TÜBİTAK 2209-A and 2209-B programs, project evaluation criteria, project topic selection, and how to write a successful project proposal. Through group work, students will mature their project proposal files. Students will indicate the same academic advisor with whom they plan to carry out their Capstone Project in their final year as their “advisor” in this project application. Students who wish to solve a problem from industry will appoint an industry advisor in addition to their academic advisor.						
References:							
<ol style="list-style-type: none"> 1. Ünel, Mustafa; Soğukpınar, İbrahim, <i>Lisansüstü Öğrenciler İçin Bir Yol Haritası</i> 2. <i>Ar-Ge Mucizesi</i>, TÜBİTAK Publications 3. Baş, Türker, <i>Bilimsel Araştırma Makalesi Nasıl Hazırlanır?</i> 4. Day, Robert A., <i>Bilimsel Makale Nasıl Yazılır ve Yayınlanır?</i>, TÜBİTAK, 1996 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Digital Signal Processing	EEE328	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course introduces digital signal processing and its applications, analog-to-digital conversion, Fourier transforms and their application areas, the Z-transform and its applications in signal processing, finite and infinite impulse response digital filter design techniques, and adaptive digital filters.						
Course Content	Introduction to digital signal processing and its applications, analog-to-digital input/output interface principles for real-time digital systems, discrete Fourier transform; fast Fourier transform, Z-transform and its applications in signal processing, correlation and convolution functions, principles of digital filter design, finite impulse response (FIR) digital filter design, window-based FIR filter design, frequency sampling-based FIR filter design, infinite impulse response (IIR) digital filters, IIR digital filter design, and adaptive digital filters.						
References:							
<ol style="list-style-type: none"> 1. Proakis, J. G., & Manolakis, D. K. (2013). <i>Digital Signal Processing: Principles, Algorithms, and Applications</i> (4th Edition). Pearson. 2. Oppenheim, A. V., & Schaffer, R. W. (2010). <i>Discrete-Time Signal Processing</i> (3rd Edition). Pearson. 3. Lyons, R. G. (2010). <i>Understanding Digital Signal Processing</i> (3rd Edition). Prentice Hall. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Automata Theory	BIL455	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	The main purpose of automata theory is to develop methods that enable computer scientists to define and analyze the dynamic behavior of discrete systems in which signals are periodically sampled.						
Course Content	Finite state machines: DFA, NFA, ϵ -NFA; Regular expressions, Pumping lemma; Context-free grammars and languages; Ambiguity in grammars and languages, Pushdown automata; Turing machines; Decidability, undecidable problems; Reducibility.						
References:							
<ol style="list-style-type: none"> 1. Sipser, Michael, <i>Introduction to the Theory of Computation</i>, 3rd Edition, Cengage Learning, 2013. 2. Assignments 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Information Technology Law	HUK319	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to provide engineering students with a basic understanding of legal concepts; to raise legal awareness regarding the liabilities arising during the development and use of information systems, cybercrimes, intellectual and artistic works (copyrights), and personal data protection.						
Course Content	Fundamental concepts of law and the Turkish judicial system, Introduction to Information Technology Law, Cybercrimes defined in the Turkish Penal Code (TCK) (unauthorized access to systems, alteration/destruction of data), Law No. 5651 on the Regulation of Broadcasting via the Internet, Law No. 6698 on the Protection of Personal Data (KVKK) and GDPR, Copyrights and licensing models for software under the Law on Intellectual and Artistic Works, Electronic Signature Law, Information technology contracts.						
References:							
<ol style="list-style-type: none"> 1. Dülger, M. V. (2021). <i>Bilişim Hukuku</i> (Updated Edition). Seçkin Yayıncılık. 2. Sözer, B. (2019). <i>Bilişim Hukuku Mevzuatı</i>. Beta Basım Yayım. 3. Law No. 6698 on the Protection of Personal Data and Law No. 5237 Turkish Penal Code. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Accessible Software	SFT---	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course aims to teach the principles of "Universal Design" that enable software products to be effectively used by everyone, including individuals with disabilities; and to equip students with inclusive software development competencies by introducing international accessibility standards (WCAG), assistive technologies, and legal obligations.						
Course Content	Introduction to accessibility and types of disabilities (visual, hearing, motor, cognitive), Assistive technologies (screen readers, Braille displays), W3C Web Content Accessibility Guidelines (WCAG 2.1/2.2) principles: Perceivable, Operable, Understandable, Robust (POUR), WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) standards, Semantic HTML structure, Color contrast and typography, Keyboard navigation, Accessibility testing tools and audits.						
References:							
<ol style="list-style-type: none"> 1. W3C Web Accessibility Initiative (WAI). (Current). <i>Web Content Accessibility Guidelines (WCAG) 2.1 / 2.2</i>. W3C Recommendation. 2. Pickering, H. (2016). <i>Inclusive Design Patterns</i>. Smashing Magazine. 3. Kalbag, L. (2017). <i>Accessibility for Everyone</i>. A Book Apart. 4. Horton, S., & Quesenbery, W. (2014). <i>A Web for Everyone: Designing Accessible User Experiences</i>. Rosenfeld Media. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Image Processing	BIL346	5	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course introduces the fundamental principles and algorithms of digital image processing systems. The course content includes image sampling and quantization, image enhancement techniques in the spatial and frequency domains, signal processing theories used for digital image processing such as one-dimensional and two-dimensional convolution, two-dimensional Fourier transform, morphological image processing, color models, and basic color image processing topics.						
Course Content	The course content includes digital images as two-dimensional signals; two-dimensional convolution, Fourier transform, and discrete cosine transform; fundamentals of image processing; image enhancement; image restoration; image coding and compression.						
References:							
<ol style="list-style-type: none"> 1. Digital Image Processing 3rd edition Rafael C. Gonzalez, Richard E. Woods 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Operational Research	ENM335	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	The aim of this course is to develop the skills of model building, problem solving, interpretation, and decision making required for industrial engineering education through the use of Operations Research techniques.						
Course Content	The history and development of Operations Research; deterministic and probabilistic models; the art of model building and problem solving; the role of linear programming in mathematical programming; linear decision models; studies on the formulation of linear decision models; solution of linear programming models; graphical, algebraic, simplex methods; revised simplex method; computer software for solving linear programming models and their use; duality and the dual simplex method; sensitivity analysis; transportation models; assignment models; network models.						
References:							
<ol style="list-style-type: none"> 1. Kaçtıoğlu, S., <i>Operations Research I Lecture Notes</i>, Update 2020. 2. Winston, W. L., <i>Operations Research: Applications and Algorithms</i>, Thomson Learning, 2004. 3. <i>Operations Research</i> / Hamdy A. Taha; translated by Ş. Alp Baray, Şakir Esnaf. 4. Özgüven, C., <i>Linear Programming and Its Extensions: Model Building Examples</i>, Detay Publishing, 2008. 5. Lecture notes of the course instructor. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Microcomputer based Systems	BIL334	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	The aim of this course is to introduce microprocessor and microcontroller architectures; to teach the communication principles between memory and input/output (I/O) units and the processor; and to develop competency in embedded system design by using low-level programming (Assembly/C) and hardware interfacing techniques.						
Course Content	Introduction to microcomputer systems and their historical development; microprocessor architecture (CPU, ALU, Registers); Von Neumann and Harvard architectures; addressing modes and instruction sets; programming with Assembly and Embedded C; memory interfacing and address decoding; interrupt mechanisms and management; input/output ports (GPIO); timers and counters; serial communication protocols (UART, SPI, I2C); analog-to-digital (ADC) and digital-to-analog (DAC) converters.						
References:							
<ol style="list-style-type: none"> 1. Mazidi, M. A., Naimi, S., & Naimi, S. (2016). <i>The AVR Microcontroller and Embedded Systems: Using Assembly and C</i>. Pearson. 2. Brey, B. B. (2019). <i>The Intel Microprocessors: Architecture, Programming, and Interfacing</i>. Pearson. 3. Valvano, J. W. (2014). <i>Embedded Systems: Introduction to ARM Cortex-M Microcontrollers</i>. CreateSpace. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Data Security	BIL354	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	The aim of this course is to teach the fundamental concepts of information and data security (Confidentiality, Integrity, Availability – the CIA Triad); and to provide theoretical and practical knowledge of cryptographic techniques (encryption, hashing), authentication protocols, and access control mechanisms used to protect data during storage and transmission.						
Course Content	Fundamentals of information security and threats; classical encryption techniques; symmetric (secret-key) encryption algorithms (DES, AES) and modes of operation; asymmetric (public-key) encryption (RSA, Diffie-Hellman, ECC); hash functions (SHA, MD5) and Message Authentication Codes (MAC); digital signatures and public key infrastructure (PKI); authentication protocols (Kerberos, X.509); database security; data masking and anonymization techniques.						
References:							
<ol style="list-style-type: none"> 1. Stallings, W. (2022). <i>Cryptography and Network Security: Principles and Practice</i> (8th Edition). Pearson. 2. Forouzan, B. A. (2015). <i>Cryptography and Network Security</i>. McGraw-Hill Education. 3. Paar, C., & Pelzl, J. (2010). <i>Understanding Cryptography: A Textbook for Students and Practitioners</i>. Springer. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Optimization	BIL---	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	The aim of this course is to develop the mathematical modeling skills required to find the best (optimal) solution to engineering problems; and to teach the theoretical foundations, convergence analyses, and computer applications of algorithms used for solving linear and nonlinear optimization problems.						
Course Content	Introduction to optimization theory and basic concepts; mathematical modeling; convex sets and convex functions; Linear Programming and the Simplex Algorithm; Unconstrained Optimization Methods: Steepest Descent (Gradient Descent), Newton–Raphson, and Conjugate Gradient methods; Constrained Optimization: Lagrange Multipliers and Karush–Kuhn–Tucker (KKT) conditions; dual problems; introduction to heuristic optimization techniques (Genetic Algorithms, Particle Swarm Optimization).						
References:							
<ol style="list-style-type: none"> 1. Chong, E. K. P., & Zak, S. H. (2013). <i>An Introduction to Optimization</i> (4th Edition). Wiley. 2. Rao, S. S. (2019). <i>Engineering Optimization: Theory and Practice</i> (5th Edition). Wiley. 3. Nocedal, J., & Wright, S. J. (2006). <i>Numerical Optimization</i>. Springer. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Control System Design	MEE322	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To introduce the fundamental concepts of control systems; to analyze control systems and evaluate their performance; and to teach basic control design techniques.						
Course Content	Introduction to Control Systems; brief review of the Laplace and inverse Laplace transforms; Block Diagrams; Transfer Functions and Signal Flow Diagrams; Time-Domain Behavior of First-Order Systems; Time-Domain Behavior of Second-Order Systems; Stability Analysis in Control Systems; Stability and the Routh–Hurwitz Criterion; Steady-State Analysis of Control Systems; Controller Structures and design techniques; Root Locus; Frequency Response Methods.						
References							
<ol style="list-style-type: none"> 1. <i>Automatic Control Systems</i>, B. C. Kuo; translated by Atilla Bir, Literatür Yayınları, 7th Edition, 1999. 2. Nise, N. S. (2004). <i>Control System Engineering</i> (4th Edition). John Wiley & Sons. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Artificial Neural Networks	BIL462	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To enable students to become familiar with the fundamental principles of artificial neural networks, to understand applications with a particular focus on image processing, and to gain practical experience using a data analytics platform such as KNIME.						
Course Content	This course covers the fundamentals and applications of artificial neural networks, with a particular emphasis on image processing. The content begins with the examination of the artificial neuron model inspired by biological neurons and aims to develop an understanding of concepts such as feedforward networks (MLP), backpropagation, regularization, and optimization. Subsequently, fundamental principles such as convolution, pooling, and classical architectures are detailed through Convolutional Neural Network (CNN) structures; students gain hands-on experience on how neural networks operate in practice by developing models on visual datasets. In the second phase of the course, data preprocessing, network design, and training processes are carried out in the KNIME environment; thus, students reinforce the theoretical framework through applied examples. With the project presentations conducted at the end of the term, a holistic learning experience is created by adapting both fundamental neural network concepts and their image processing-oriented application areas to real-world scenarios.						
References:	<ol style="list-style-type: none"> 1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). <i>Deep Learning</i>. MIT Press. 2. <i>Hands-On Deep Learning with KNIME</i> – KNIME Hub and Official Documentation. 3. Haykin, S. S. <i>Neural Networks: A Comprehensive Foundation</i>. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Embedded Systems for Software Engineers	SFT378	6	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To introduce software engineering students to the architecture of embedded systems in which hardware and software operate in an integrated manner; and to develop competency in designing reliable and efficient software that meets real-time constraints in environments with limited resources (memory, processing power, energy).						
Course Content	Introduction to embedded systems and their differences from general-purpose systems; microcontroller architecture (ARM Cortex-M, etc.); embedded software development toolchains (cross-compiler, debugger); Hardware Abstraction Layers (HAL) and device drivers; interrupt-based programming; timers and PWM; serial communication (UART, SPI, I2C); analog interfaces (ADC/DAC); introduction to Real-Time Operating Systems (RTOS); task scheduling and synchronization; power management and optimization in embedded systems.						
References:							
<ol style="list-style-type: none"> White, E. (2011). <i>Making Embedded Systems: Design Patterns for Great Software</i>. O'Reilly Media. Valvano, J. W. (2014). <i>Embedded Systems: Introduction to ARM Cortex-M Microcontrollers</i>. CreateSpace. Simon, D. E. (1999). <i>An Embedded Software Primer</i>. Addison-Wesley. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Biomedical Instrumentation	BIL423	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To learn the operating principles of biomedical sensors and transducers; to understand the use of biomedical devices in disease diagnosis and treatment; and to examine signal processing and recording devices.						
Course Content	Basic concepts; Biomedical Engineering and the human instrumentation system; biosignals and their characteristics; basic electrical concepts and laws; biomedical sensors and transducers; harmful effects of electric current on human health; signal conditioning; basic measurement theories; image formation and medical imaging systems; signal processing and recording devices.						
References:							
<ol style="list-style-type: none"> Prof. Dr. İnan Güler, <i>Biomedical Instrumentation Lecture Notes</i>, Gazi University, 2008. Dr. Mehlika Karamanlioğlu, Dr. Hakan Akça, <i>Introduction to Biomedical Instrumentation</i>. Nobel Publishing, 2021. 							

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Windows Programming	BIL443	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To introduce students to the application development architecture on the Windows operating system (.NET Framework / .NET Core); and to develop competency in creating desktop software with rich graphical user interfaces (GUI), database connectivity, and multithreaded capabilities by teaching the event-driven programming paradigm.						
Course Content	Fundamentals of Windows programming and the .NET platform; use of the Visual Studio IDE; review of Object-Oriented Programming (OOP) using the C# programming language; Windows Forms and WPF (Windows Presentation Foundation) architectures; form design and basic controls (Button, TextBox, ListView, etc.); event and delegate mechanisms; menus and dialog boxes; graphics programming with GDI+; file and directory operations (I/O); database connectivity and CRUD operations using ADO.NET; multithreaded programming and asynchronous operations (Async/Await); application packaging and deployment.						
References:	<ol style="list-style-type: none"> 1. Troelsen, A., & Japikse, P. (2022). <i>Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming</i>. Apress. 2. Petzold, C. (2000). <i>Programming Windows</i> (5th Edition). Microsoft Press. 3. Albahari, J. (2022). <i>C# 10.0 in a Nutshell: The Definitive Reference</i>. O'Reilly Media. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Software Design Patterns	SFT405	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To introduce students to object-oriented design principles and software design patterns, including creational, structural, and behavioral categories. The course aims to enable application of these patterns in real-world software development through projects and analysis of architectural patterns and anti-patterns.						
Course Content	<ul style="list-style-type: none"> • Introduction to design patterns and object-oriented principles • Creational patterns (Singleton, Factory, Prototype, Builder) • Structural patterns (Adapter, Bridge, Composite, Decorator, Facade, Flyweight) • Behavioral patterns (Observer, Strategy, Chain of Responsibility, Iterator, State) • Architectural patterns, anti-patterns, refactoring, and case studies 						
References:	<ul style="list-style-type: none"> • "Design Patterns: Elements of Reusable Object-Oriented Software" by Gamma, Helm, Johnson, Vlissides (Gang of Four). • Lecture notes and online resources (e.g., Refactoring.Guru, Sabanci University syllabus materials). • Supplementary: "Head First Design Patterns" by Freeman & Freeman. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Data Mining	BIL460	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To provide students with the fundamental methods of data mining, a general approach to algorithms that can solve different data mining topics, and to apply these algorithms to real problems.						
Course Content	This course includes the theoretical and applied education of Introduction to Data Mining and Basic Concepts, Data Mining Process, Feature Selection and Data Transformation, Clustering Methods, Classification Methods-Decision Trees, Predictive Analysis Success Measurement Methods, and Association Rule Analysis. Additionally, problem-oriented data mining and End-of-Term Project presentations are included.						
References:	<ol style="list-style-type: none"> 1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Addison Wesley, (2005) 2. Han Jiawei and Kamber Micheline (2006), Data Mining: Concepts and Techniques, Morgan Kaufmann Publisher San Francisco 3. Veri Madencilği Teori Uygulama ve Felsefesi, Papatya Yayınları, Dr. İlker KÖSE (2015) 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Parallel Computers	BIL461	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students parallel computing architectures and hardware classifications (Flynn Taxonomy); to provide competence in designing parallel algorithms for shared memory (Shared Memory) and distributed memory (Distributed Memory) systems, developing high-performance applications using standard libraries such as MPI and OpenMP, and performing performance analysis.						
Course Content	Introduction to parallel computing and motivation (end of Moore's Law), Hardware architectures and Flynn Classification (SISD, SIMD, MISD, MIMD), Memory architectures: Shared, Distributed, and Hybrid models, Shared memory programming and OpenMP, Thread management and synchronization issues (Deadlock, Race Condition), Distributed memory programming and MPI (Message Passing Interface), Point-to-point and collective communication, Performance metrics: Speedup, Efficiency, and Amdahl's Law, GPU Architecture and introduction to CUDA programming, Parallel sorting and matrix multiplication algorithms.						
References:	<ol style="list-style-type: none"> 1. Pacheco, P. S. (2011). An Introduction to Parallel Programming. Morgan Kaufmann. 2. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). Introduction to Parallel Computing (2nd Edition). Pearson. 3. Kirk, D. B., & Hwu, W. M. (2016). Programming Massively Parallel Processors: A Hands-on Approach (3rd Edition). Morgan Kaufmann. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Fuzzy Logic	BIL463	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the mathematical modeling of uncertain and vague concepts that cannot be expressed with classical (crisp) logic; to provide competence in designing artificial intelligence-based control and decision-making systems close to human thought structure using fuzzy set theory, membership functions, and inference mechanisms (Inference Systems).						
Course Content	Differences between classical (Crisp) and Fuzzy sets, Uncertainty concept, Membership functions (Triangular, Trapezoidal, Gauss, etc.) and design, Fuzzy set operations (Union, Intersection, Complement), Linguistic variables and "If-Then" rule bases, Fuzzy Relations and Composition, Fuzzy Inference Systems (FIS): Mamdani and Takagi-Sugeno models, Defuzzification methods (Center of gravity, Maximum membership, etc.), Design of Fuzzy Control systems, Applications of fuzzy logic in machine learning and pattern recognition, Fuzzy system simulations with MATLAB/Python.						
References:	<ol style="list-style-type: none"> Ross, T. J. (2016). Fuzzy Logic with Engineering Applications (4th Edition). Wiley. Klir, G. J., & Yuan, B. (1995). Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall. Zadeh, L. A. (Articles and Reviews). Fuzzy Sets (Seminal Paper, 1965). Information and Control. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Big Data Computing And Technologies	SFT465	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students distributed architectures used for storing and analyzing large-volume, fast, and diverse data (Big Data) that cannot be processed with traditional database systems; to provide competence in developing scalable data processing solutions using ecosystems like Hadoop and Spark, NoSQL databases, and stream processing techniques.						
Course Content	Big Data concept and 5V (Volume, Velocity, Variety, Veracity, Value) principle, Distributed file systems (HDFS - Hadoop Distributed File System), MapReduce programming model, Apache Spark architecture and RDD/DataFrame structures, In-memory computing, NoSQL database types (Key-Value, Document, Column, Graph) and CAP Theorem (Consistency, Availability, Partition Tolerance), Data warehouse tools (Apache Hive), Real-time data stream processing (Spark Streaming, Kafka), Machine learning on big data (Spark MLlib).						
References:	<ol style="list-style-type: none"> White, T. (2015). Hadoop: The Definitive Guide (4th Edition). O'Reilly Media. Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). Learning Spark: Lightning-Fast Big Data Analysis. O'Reilly Media. Marz, N., & Warren, J. (2015). Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Manning Publications. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Compiler Design	BIL467	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the processes of converting high-level programming languages into machine code and the fundamental principles of compiler design; to provide competence in developing their own compilers or interpreters by theoretically and practically understanding lexical, syntactic, and semantic analysis phases.						
Course Content	Introduction to compilers and compilation phases, Lexical Analysis and finite automata, Syntax Analysis and Context-Free Grammar (CFG), Top-down (LL) and bottom-up (LR, LALR) parsing techniques, Parser generators (Lex/Yacc or Flex/Bison), Semantic Analysis and type checking, Symbol table management, Intermediate code generation (Three-address code), Runtime Environments, Code optimization techniques, Target code generation.						
References:	<ol style="list-style-type: none"> 1. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). <i>Compilers: Principles, Techniques, and Tools</i> (2nd Edition). Pearson. 2. Cooper, K. D., & Torczon, L. (2011). <i>Engineering a Compiler</i>. Morgan Kaufmann. 3. Louden, K. C. (1997). <i>Compiler Construction: Principles and Practice</i>. Cengage Learning. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Augmented/Virtual Reality	BIL433	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the theoretical foundations and hardware components of virtual (VR), augmented (AR), and mixed (MR) reality technologies; to provide competence in developing real-time immersive applications on current game engines (Unity/Unreal) by understanding 3D user interaction, rendering techniques, and human perception principles.						
Course Content	Introduction to Virtual and Augmented Reality, History and applications, Human visual system and perception, Hardware components (Headsets, Sensors, Haptic devices), 3D Mathematics and Geometric Transformations, Graphics rendering pipeline, Stereoscopic rendering, Tracking technologies, 3D User Interfaces and Interaction techniques (Selection, Manipulation, Navigation), Marker-based and markerless tracking in Augmented Reality, Motion Sickness and ergonomics, Immersive application development tools.						
References:	<ol style="list-style-type: none"> 1. LaValle, S. M. (2017). <i>Virtual Reality</i>. Cambridge University Press. 2. Schmalstieg, D., & Hollerer, T. (2016). <i>Augmented Reality: Principles and Practice</i>. Addison-Wesley. 3. Burdea, G. C., & Coiffet, P. (2003). <i>Virtual Reality Technology</i> (2nd Edition). Wiley-IEEE Press. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Internet of Things	SFT411	7	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the basic architecture and components of the Internet of Things ecosystem; to provide competence in designing and implementing end-to-end IoT solutions covering sensors, embedded systems, low-power communication protocols, and cloud integration						
Course Content	Introduction to IoT concept and architectural layers (Perception, Network, Application), Sensors and Actuators, Embedded prototyping platforms (Arduino, Raspberry Pi, ESP32), M2M Communication protocols: MQTT, CoAP, HTTP and WebSocket, Wireless connectivity technologies: Wi-Fi, Bluetooth Low Energy (BLE), Zigbee and LoRaWAN, Cloud computing integration and IoT platforms (AWS IoT, Azure), Edge Computing and Fog Computing, IoT security and privacy, Industry 4.0 and smart city applications.						
References:	<ol style="list-style-type: none"> 1. Bahga, A., & Madiseti, V. (2014). Internet of Things: A Hands-On Approach. VPT. 2. Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., & Henry, J. (2017). IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. Cisco Press. 3. McEwen, A., & Cassimally, H. (2013). Designing the Internet of Things. Wiley. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Digital Audio	BIL444	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To learn the qualitative properties of sound and audio. To learn digital sound synthesis methods. To learn the advantages and disadvantages of different synthesis methods relative to each other.						
Course Content	Basic properties of physical sound and audio. Sound and audio measurements. Sampling theorem, aliasing, quantization, dither. Digital sound recording, pulse code modulation, dither generator, low-pass filters. A/D converters, D/A converters, digital filters, timebase correction. Error correction methods. Magnetic recording, digital audio tape, optical disk storage. Digital Sound Synthesis: Basic sound synthesis techniques, Digital sound production, abstract synthesis methods, sound production with physical models. Digital audio signal processing. MIDI: Basic MIDI definitions, MIDI control of sound, synthesis, MIDI and data compression (MPEG4).						
References:	<ol style="list-style-type: none"> 1. Bilbao, S. (2009). Numerical sound synthesis: finite difference schemes and simulation in musical acoustics. John Wiley & Sons. 2. Digital signal processing : a practical guide for engineers and scientists / Steven W. Smith Smith, Steven W. 2003 3. Russ, M. (2004). Sound synthesis and sampling. Taylor & Francis. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
E-Commerce System Design	BIL446	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	Introduction to E-Commerce, E-Commerce Business Models and Concepts, E-Commerce Infrastructure: Internet Web, Mobile Platform, Creating E-Commerce Presence: Web Sites Mobile Sites and Applications, E-Commerce Security and Payment Systems, E-Commerce Marketing Concepts; social, mobile, local, E-Commerce Marketing Communication, Ethical, Social, and Political E-Commerce topics.						
Course Content	Introduction to E-Commerce, E-Commerce Business Models and Concepts, E-Commerce Infrastructure: Internet Web, Mobile Platform, Creating E-Commerce Presence: Web Sites Mobile Sites and Applications, E-Commerce Security and Payment Systems, E-Commerce Marketing Concepts; social, mobile, local, E-Commerce Marketing Communication, Ethical, Social, and Political E-Commerce topics.						
References:	<ol style="list-style-type: none"> 1. Computer and projector. 2. Laudon/Travers E-commerce 8/e 2012, Introduction to E-commerce, Zheng Qin, Springer 2009 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Test Engineering	BIL454	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the fundamentals of software quality assurance, verification and validation processes; to provide competence in developing error-free and reliable software by theoretically and practically understanding black-box and white-box testing techniques, static analysis methods, and test automation tools.						
Course Content	Introduction to software testing and basic principles, The place of testing in Software Development Life Cycle (SDLC) and V-Model, Static testing techniques and code review, Black-box testing techniques: Equivalence class partitioning, Boundary value analysis, Decision tables, White-box testing techniques: Statement, Decision and Path coverage analyses, Unit testing and Test-Driven Development (TDD), Integration, System and Acceptance testing, Regression testing, Introduction to test automation and tools (Selenium, JUnit), Performance and load testing, Defect lifecycle and management.						
References:	<ol style="list-style-type: none"> 1. Jorgensen, P. C. (2013). Software Testing: A Craftsman's Approach (4th Edition). CRC Press. 2. Graham, D., van Veenendaal, E., Evans, I., & Black, R. (2008). Foundations of Software Testing: ISTQB Certification. Cengage Learning. 3. Ammann, P., & Offutt, J. (2016). Introduction to Software Testing (2nd Edition). Cambridge University Press. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Logic Programming	BIL456	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To introduce students to the non-procedural (declarative) programming paradigm; to teach how mathematical logic (First-Order Predicate Logic) is used as a computational tool and to provide competence in developing applications in areas such as symbolic artificial intelligence, natural language processing, and expert systems using the Prolog language.						
Course Content	Introduction to logic programming and history, Propositional logic and Predicate logic (First-Order Predicate Logic), Horn clauses and Resolution principle, Prolog programming language basics: Facts, Rules and Queries, Prolog's working mechanism: Unification and Backtracking, Recursion and List processing, Arithmetic operations, Cut operator and negation, Search strategies with logic programming (DFS, BFS), Natural language parsing with Definite Clause Grammars (DCG).						
References:	<ol style="list-style-type: none"> 1. Bratko, I. (2012). Prolog Programming for Artificial Intelligence (4th Edition). Addison-Wesley. 2. Sterling, L., & Shapiro, E. (1994). The Art of Prolog: Advanced Programming Techniques (2nd Edition). MIT Press. 3. Clocksin, W. F., & Mellish, C. S. (2003). Programming in Prolog (5th Edition). Springer. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Bioinformatics	BIL458	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the algorithmic foundations necessary for analyzing biological data using computational methods; to provide competence in sequence alignment of DNA, RNA and proteins, genetic database usage, and phylogenetic analysis, thereby enabling the development of informatics-based solutions for molecular biology problems.						
Course Content	Introduction to Bioinformatics and the Central Dogma of Molecular Biology, Biological databases (NCBI, GenBank, PDB, UniProt), Pairwise sequence alignment, Dynamic programming algorithms (Needleman-Wunsch and Smith-Waterman), Scoring matrices (PAM and BLOSUM), Heuristic search algorithms and BLAST, Multiple sequence alignment, Phylogenetic trees and evolutionary analysis methods (UPGMA, Neighbor Joining), Protein structure prediction and visualization, Genomic and proteomic analysis techniques.						
References:	<ol style="list-style-type: none"> 1. Jones, N. C., & Pevzner, P. A. (2004). An Introduction to Bioinformatics Algorithms. MIT Press. 2. Lesk, A. M. (2019). Introduction to Bioinformatics (5th Edition). Oxford University Press. 3. Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Natural Language Processing	SFT488	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the theoretical and practical methods used for processing, understanding, and generating natural languages in a computer environment; to introduce text mining, linguistic analysis, and statistical/deep learning-based language models, thereby providing competence in developing NLP applications such as machine translation, sentiment analysis, and question-answering systems.						
Course Content	Introduction to Natural Language Processing and history, Text preprocessing techniques (Tokenization, Stemming, Lemmatization, Stop-words), Regular Expressions, N-Gram language models and probability calculations, Part-of-Speech Tagging (POS Tagging) and Hidden Markov Models (HMM), Syntactic Parsing, Vector Semantics and Word Embedding models (Word2Vec, GloVe, FastText), Sequence models (RNN, LSTM, GRU), Attention mechanism and Transformer architectures (BERT, GPT), Machine Translation, Sentiment Analysis, and Named Entity Recognition (NER).						
References:	<ol style="list-style-type: none"> 1. Jurafsky, D., & Martin, J. H. (2024). Speech and Language Processing (3rd Edition Draft). Pearson. 2. Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press. 3. Eisenstein, J. (2019). Introduction to Natural Language Processing. MIT Press. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Network Security	BIL468	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	This course provides an in-depth understanding of network security principles, encryption techniques, and modern cybersecurity threats. Students will explore fundamental security objectives, encryption methods, cryptographic algorithms, and network security protocols. The course covers topics such as symmetric and asymmetric cryptography, hash functions, key management, cybersecurity threats, malware analysis, and secure communication protocols. Additionally, students will gain knowledge about security considerations in web applications, email, and cloud environments. Through theoretical study and practical exercises, students will develop the skills necessary to analyze and implement secure network solutions.						
Course Content	Network Security, Block Cipher and Data Encryption Standard, Finite Fields, Contemporary Symmetric Encryption, Pseudorandom Number Generation and Stream Ciphers, Introduction to Number Theory, Public Key Encryption and RSA, Other Public Key Encryption Systems, Encryption and Network Security, Hash and MAC Algorithms, Key Management and Distribution, Cybersecurity, Cyberattack, Method, Opportunity, and Motive, Viruses and Other Malicious Code, Transport Layer Security, Web Applications, Privacy and Open-Source Intelligence, Email and Cloud Security.						
References:	<ol style="list-style-type: none"> 1. Cryptography and Network Security”, 5/e, by William Stallings 2. Network Security Essentials: Applications and Standards by William Stallings 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
DevOps and CI/CD Applications	SFT478	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	The objective of this course is to teach the DevOps culture and practices that enhance collaboration between software development and operations teams; to develop competencies in automating Continuous Integration (CI) and Continuous Deployment (CD) processes that accelerate the software lifecycle, as well as in container technologies, orchestration, and infrastructure management through code (Infrastructure as Code – IaC).						
Course Content	The concept, history, and culture of DevOps; Version Control Systems (Git) and workflows; principles and tools of Continuous Integration (CI) (Jenkins, GitLab CI, GitHub Actions); Continuous Delivery and Deployment (CD) strategies (Blue-Green, Canary); virtualization and container technologies (Docker); architecture of container orchestration (Kubernetes); Infrastructure as Code (Terraform, Ansible); configuration management; DevOps for microservice architectures; monitoring and logging stacks (Prometheus, Grafana, ELK); and the concept of DevSecOps.						
References:	<ol style="list-style-type: none"> 1. Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press. 2. Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley. 3. Freeman, E. (2019). DevOps for Dummies. Wiley. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Blockchain and Smart Contracts	SFT412	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach students the fundamental principles of decentralized systems and blockchain technology; to provide a solid understanding of cryptographic foundations, consensus mechanisms, and smart contract architectures, thereby enabling them to develop secure and functional decentralized applications (DApps) on platforms such as Ethereum.						
Course Content	Introduction to blockchain technology and its history (Bitcoin); centralized and distributed ledger technologies (DLT); cryptographic foundations (hash functions, digital signatures, Merkle trees); block structure and chaining; consensus mechanisms: Proof of Work (PoW), Proof of Stake (PoS), and others; the Ethereum platform and the EVM (Ethereum Virtual Machine); smart contracts and their life cycle; the Solidity programming language; token standards (ERC-20, ERC-721); decentralized application (DApp) development tools (Remix, Truffle/Hardhat, Web3.js); blockchain security and attack vectors; applications in finance (DeFi) and supply chain.						
References:	<ol style="list-style-type: none"> 1. Antonopoulos, A. M., & Wood, G. (2018). Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly Media. 2. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press. 3. Antonopoulos, A. M. (2017). Mastering Bitcoin: Programming the Open Blockchain (2nd Edition). O'Reilly Media. 						

Course Name	Course Code	Semester	Theory Hours/Week	Practice Hours/Week	Laboratory Hours/Week	Local Credit	ECTS
Realtime Systems for Software Engineers	SFT462	8	3	0	0	3	5
Prerequisite(s)							
Course Type	Elective						
Course Objectives	To teach software engineering students the characteristic properties of real-time systems, scheduling algorithms, and synchronization mechanisms; and to enable them to design predictable, reliable, and high-performance software architectures for systems with strict time constraints by using real-time operating systems (RTOS).						
Course Content	Introduction to real-time systems and hard/soft real-time concepts; reference models for real-time systems; task scheduling approaches (time-driven, priority-driven); scheduling algorithms: Rate Monotonic (RM) and Earliest Deadline First (EDF); resource sharing and the problem of priority inversion; Priority Inheritance Protocol; architecture and kernel structure of Real-Time Operating Systems (RTOS); inter-task communication and synchronization; fault tolerance and reliability; real-time software design patterns.						
References:	<ol style="list-style-type: none"> 1. Liu, J. W. S. (2000). Real-Time Systems. Pearson. 2. Buttazzo, G. C. (2011). Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (3rd Edition). Springer. 3. Laplante, P. A., & Ovaska, S. J. (2011). Real-Time Systems Design and Analysis (4th Edition). Wiley-IEEE Press. 						