FINAL INTERNATIONAL UNIVERSITY

VOCATIONAL SCHOOL

Computer Programming Program of Courses

Course categories: UC = University Core; FC = Faculty Core; AC = Area Core; AE = Area Elective; FE = Faculty Elective; UE = University Elective

Semester	Course Code	Course Title	Course Category	Hours			Total Credit	Pre-requisite	ECTS Credit
				Lecture	Tutorial	Lab/Prac.			
1	ENGR103	COMPUTER PROGRAMMING-I	FC	2	0	2	3	-	5
1	ENGR101	INFORMATION TECHNOLOGY AND APPLICATIONS	FC	2	0	1	2	-	2
1	MATH121	CALCULUS-I	FC	3	2	0	4	-	6
1	MATH123	DISCRETE MATHEMATICS	FC	3	1	0	3	-	5
1	ENGL121	ENGLISH-I	UC	3	0	0	3	-	4
1	TARH101 / HIST111	ATATURK'S PRINCIPLES AND HISTORY OF TURKISH REFORMS-I	UC	2	0	0	2	-	3
1	TUOG101 / TURK131	TURKISH LANGUAGE-I / TURKISH AS A FOREIGN LANGUAGE-I	UC	2	0	0	2	-	3
		Total 7 courses	**TOTAL:**	**17**	**3**	**3**	**19**		**28**
2	ENGR104	COMPUTER PROGRAMMING-II	FC	2	0	2	3	ENGR103	4
2	CPRG102	INTRODUCTION TO DATABASES	AC	3	0	2	4		6
2	CPRG104	INTRODUCTION TO INTERNET PROGRAMMING	AC	3	0	2	4		6
2	CPRG106	INTRODUCTION TO MOBILE APPS DEVELOPMENT	AC	3	0	2	4		6
2	ENGL122	ENGLISH-II	UC	3	0	0	3	ENGL121	4
2	TARH102 / HIST112	ATATURK'S PRINCIPLES AND HISTORY OF TURKISH REFORMS-II	UC	2	0	0	2	-	3
2	TUOG102 / TURK132	TURKISH LANGUAGE-II / TURKISH AS A FOREIGN LANGUAGE-II	UC	2	0	0	2	- / TURK131	3
		Total 7 courses	**TOTAL:**	**18**	**0**	**8**	**22**		**32**
3	CPRG201	FURTHER TOPICS IN INTERNET PROGRAMMING, DATABASES, AND SQL	AC	2	0	2	3	CPRG104	6
3	CMPE215	ALGORITHMS AND DATA STRUCTURES	FC	3	0	1	3	ENGR104	6
3	CPRG200	SUMMER TRAINING	AC	0	0	0	0	-	2
3	CPRGXX1	AREA ELECTIVE	AE	X	X	X	3	-	6
3	CPRGXX2	AREA ELECTIVE	AE	X	X	X	3	-	6
3	UNIEXX1	UNIVERSITY ELECTIVE	UE	X	X	X	3	-	4
		Total 6 courses	**TOTAL:**	**5**	**0**	**3**	**15**		**30**
4	CMPE216	OBJECT ORIENTED PROGRAMMING	FC	2	0	2	3	ENGR104	6
4	CPRG202	C#	AC	2	0	4	4	ENGR104	4
4	CPRG204	GRADUATION PROJECT	AC	0	0	0	0	-	4
4	CPRGXX3	AREA ELECTIVE	AE	X	X	X	3	-	6
4	CPRGXX4	AREA ELECTIVE	AE	X	X	X	3	-	6
4	UNIEXX2	UNIVERSITY ELECTIVE	UE	X	X	X	3	-	4
		Total 6 courses	**TOTAL:**	**4**	**0**	**6**	**16**		**30**

Area and Faculty Elective Courses

No.	Course Code	Course Title	Course Category	Hours			Total Credit	Pre-requisite	ECTS Credit
				Lecture	Tutorial	Lab/Prac.			
1	CPRG209	ADVANCED TOPICS IN PYTHON	AE	3	0	0	3	-	6
2	CPRG210	PROGRAMMING FOR ANDROID	AE	3	0	0	3	-	6
3	CPRG211	PROGRAMMING FOR IOS	AE	3	0	0	3	-	6
4	CPRG212	JAVA FOR HANDHELD TERMINALS	AE	3	0	0	3	-	6
5	CPRG214	VISUAL PROGRAMMING	AE	3	0	0	3	-	6
6	CPRG215	SOFTWARE TESTING	AE	3	0	0	3	-	6
7	CPRG216	SOFTWARE DEVELOPMENT	AE	3	0	0	3	-	6
8	CPRG217	COMPUTER GRAPHICS APPLICATIONS	AE	3	0	0	3	-	6
9	CPRG218	CLOUD COMPUTING APPLICATIONS	AE	3	0	0	3	-	6

PROGRAM INFORMATION

General Goal of the Program	Our Computer Programming program aims to graduate highly skilled and knowledgeable professionals with hands-on experience who can be outstanding experts.
Program Outputs	1. Apply knowledge of Mathematics, Science, and Engineering to solve complex problems in Computer Programming.
	2. Identify, formulate, design, analyze and implement computer system, component, or process to meet desired needs.
	3. Design system components that meet economic, environmental, social, political, ethical, health and safety, and sustainability requirements.
	4. Conduct investigations of complex engineering problems including design of experiments, analysis, and interpretation of data, and synthesis of the
	5. Construct, select and apply appropriate techniques, resources, and modern simulation tools to solve complex computer related problems.
	6. Apply contextual knowledge to assess social, health, safety, and cultural issues and endure the consequent responsibilities relevant to professional
	7. Utilize core engineering knowledge in a global, economic, environmental, and societal context for sustainable development.
	8. Solve professional, legal, and ethical issues pertaining to core engineering and its related fields.
	9. Function effectively as a team member or a leader to accomplish a common goal in a multi-disciplinary team.
	10. Communicate effectively in both verbal and written forms.
	11. Apply knowledge of engineering and management principles to manage projects effectively in diverse environments as a member or leader of a
	12. Engage in independent and lifelong learning for continued professional development.

COURSE DESCRIPTIONS

Course Descriptions – I: All Area Core and Faculty/School Core courses offered by the department of the program.

Code	Course Title	Credit	Credit	Catego.	Pre-requisite	Teaching Language
ENGR103	COMPUTER PROGRAMMING-I	(2, 0, 2)3	5	FC	-	English
Course Content	The Computer Programming course introduces students to the concept of programming including designing algorithms and writing pseudo-code to solve engineering-related problems, creating flowcharts to represent the steps of a problem solution, and the basic elements of the Python programming language the implement their solution. The course covers common high-level programming concepts such as Data types, constants and variables, arithmetic and logical operators, decision-making expressions. Fundamental components of Python included in the course are; storing and manipulating input data, design and use of selection structures, repetition structures, various data structures such as lists, dictionaries and sets, functions, and modular design.					
ENGR101	Information Technology and Applications	(2, 0, 1)2	2	FC	-	English
Course Content	This course aims to introduce all students to the basic concepts of information technology and to train them in the skills needed to use office productivity tools. Course subjects include; History of Computing, Fundamental Hardware descriptions and functions, Software types and functions, Numbering Systems and Binary, Input, Output and Storage devices, Internet and the World Wide Web, Understanding Networks, Privacy while using Computers, Computer Crimes and Security, Computer Ethics, Cloud Computing fundamentals. The course also covers the usage of Microsoft Word, PowerPoint, and Excel.					
MATH121	CALCULUS-I	(3, 2, 0)4	6	FC	-	English
Course Content	Calculus-I covers differential and integral calculus, with applications in geometry, physics, and engineering. Students will learn to apply calculus concepts to various scientific and engineering applications. Topics include identifying function types, graphing functions, evaluating limits, handling elementary functions (polynomial, trigonometric, logarithmic, exponential, etc.), solving undefined limits, and evaluating derivatives. Derivatives of elementary functions, product, and quotient rules will be covered, along with applications of derivatives. Integration topics include evaluating integrals, defining integrals, and using methods like substitution, integration by parts, and integrating rational functions. The course will also explore the practical applications of integration.					
MATH123	DISCRETE MATHEMATICS	(3, 1, 0)3	5	FC	-	English
	Discrete mathematics is the first non-calculus course for mathematics, computer science, and engineering majors. This course introduces the mathematical tools and techniques used to study discrete processes as opposed to continuous processes. Topics covered include discrete concepts such as basic set theory, functions, relations, recurrences, counting principles, the fundamentals of propositional logic and Boolean algebra, graphs, and trees. The course also introduces proof techniques in mathematics, including proof by induction, proof by truth table, proof by Venn diagram, etc. This course is indeed a prerequisite for logic design, operational research, combinatorics, abstract algebra, mathematical modeling, geometry, and topology courses.					

Course Content						
ENGR104	COMPUTER PROGRAMMING II	(2, 0, 2)3	4	FC	ENGR103	English
Course Content	Review of the C programming language. Structured and modular programming using C. Local and global variables. Structured programming constructs. Arrays and array handling. Multi-dimensional arrays. Structures and Unions. Arrays of structures. Defining new data types in C. Functions in C. Call-by-value and call-by-reference. Character and string functions. Scope and extent. Recursion. Pointers and pointer arithmetic. Dynamic memory allocation and simple data structures in C. Arrays of pointers. Bit manipulation. Files; data and file processing. Conditional compilation and exception handling in C.					
CPRG102	DATABASE SYSTEMS	(3, 0, 2)4	6	AC	-	English
Course Content	This is a database management system introduction course. The lectures' primary goal is to show students how to conceptually model data and then implement that model in SQL. The focus of the lectures is on practical aspects of data modeling, including normalization and the creation of entity connection diagrams. Oracle is used in the labs to teach SQL. The purpose of lab work is to thoroughly introduce SQL and, in particular, the SQL data manipulation language statement. The learner will be able to create databases for use in industry after completing this course.					
CPRG104	INTERNET AND WEB PROGRAMMING	(3, 0, 2)4	6	AC	-	English
Course Content	This course is an introduction to programming for the World Wide Web. Students will learn about the relationship between clients and servers, how the internet works, and how web pages are constructed using several technologies. The following topics will be covered: HyperText Markup Language (HTML) for authoring web pages; Cascading Style Sheets (CSS) for applying stylistic information to web pages; JavaScript (JS) for creating interactive web pages; Asynchronous JavaScript and XML (Ajax) for enhanced web interaction and applications; PHP web services for handling and responding to web service requests; and Structured Query Language (SQL) for interacting with databases. After successfully completing this course, a student should be able to Support the development of web pages.					
CPRG106	INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT	(3, 0, 2)4	6	AC	-	English
Course Content	In this course, students will learn to develop cross-platform mobile applications to solve business problems using at least two modern programming languages. Topics covered include different software platforms, tools for development, the user interface, and database handling for mobile applications. Course Content includes an introduction to layouts, views, Intents on Android, menus, dialogs, data persistence, databases and their applications, and the navigation drawer. Upon completion of this course, the student will be able to: Develop, debug, and test a mobile application efficiently.					
CPRG201	FURTHER TOPICS IN INTERNET PROGRAMMING, DATABASES, AND SQL	(2, 0, 2)3	6	AC	CPRG104	English
Course Content	This course aims to encourage students to improve their team-working skills on web-based projects. The course focuses on designing, developing, and presenting real-life web projects. Students who take this course will use one or more web development tools in order to design and develop websites. They will also learn how to deal with different web development frameworks like Laravel, Codeigniter, and Django. The importance of web programming languages is discussed in a real-life project. On successful completion of this course, students should be able to: design professional web applications; develop database-driven component-based web applications; create event-driven applications; and design GUIs using frontend programming languages.					
CMPE215	ALGORITHMS AND DATA STRUCTURES	(3, 0, 1)3	6	FC	ENGR104	English
Course Content	The objective of this course is to provide the basics of data structures and data organization. The course will introduce C/C++ and algorithms for the implementation of data structures which are stack, queue, linked list, and tree. Also, the applications of data structures cover stack applications which are parenthesis checker, infix to postfix and prefix conversions, recursion, dynamic stack and queue, and tree traversals. Theoretical aspects of the most widely used data structures will be covered during the lectures. Programming assignments and lab works cover the C/C++ implementations of applications of data structures that are discussed in the lectures.					
CPRG200	SUMMER TRAINING	(0, 0, 0)0	2	AC	-	English
Course Content	Computer Programming summer training is a 30-day internship for students to apply theoretical knowledge from their studies in a professional setting. The training can take place at any institution related to Computer Programming and Development. Students work on real-life tasks, interact with professionals, and explore their interests within the industry. After the first year of their studies, they write summer training reports summarizing their experiences. A committee evaluates these reports to assess the students' internship performance. The training aims to bridge the gap between academia and industry, enabling students to better prepare for future career opportunities and make informed decisions about their professional path.					
CMPE216	OBJECT ORIENTED PROGRAMMING	(2, 0, 2)3	6	FC	ENGR104	English
Course Content	This course introduces the concepts of object-oriented programming to students with a background in the procedural paradigm. The course begins with a brief review of control structures and data types with emphasis on structured data types and array processing. It then moves on to introduce the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Other topics include an overview of programming language principles, simple analysis of algorithms, basic searching and sorting techniques, memory management, an introduction to software engineering issues, and ethics in software development.					
CPRG202	C#	(2, 0, 4)4	6	AC	ENGR104	English
Course Content	This course introduces computer programming using the Visual Programming Language with object-oriented programming principles. The emphasis is on event-driven programming methods, including creating and manipulating objects and classes and using object-oriented tools such as the class debugger. Visual programming languages are widely used for the rapid development of graphical applications. This subject will introduce students to the fundamental principles of event-driven programming and to programming in a visual environment through the use of the Visual C# programming language. An additional aim of this subject is to give students an understanding of the main ideas of human-computer Interaction (HCI). Upon completion, students should be able to design, code, test, and debug at a beginning level.					
CPRG204	GRADUATION PROJECT	(2, 0, 2)3	6	AC	-	English
Course Content	Engineering Design is a crucial activity for engineering students, involving various phases of the design process. Students work in teams under supervision to complete interdisciplinary capstone projects over one academic year, spanning ELEE401 and ELEE402 courses. ELEE401 covers problem formulation, technical surveys, detailed problem study, analysis, and methodical initial solution formulation. The course requires comprehensive preliminary design documentation for solving a realistic and complex computer engineering problem, applying skills gained throughout the undergraduate program. Students present progress through reports and presentations during the semester and at its conclusion. This extended exercise aims to cultivate professional application and experience in engineering design.					

Course Descriptions – III: All Area Elective and Faculty/School Elective courses offered by the department of the program.

Code	Course Title	Credit	Credit	Catego.	Pre-requisite	Teaching Language
CPRG209	ADVANCED TOPICS IN PYTHON	(3, 0 ,0)3	6	AE	-	English
	This course covers the Python programming language, which has advanced features with an emphasis on programming practice. A graphical user interface, data analysis and visualization, web data extraction, and web applications are examples of programming tasks. Students who successfully complete this course will be acquainted with the following principles and will be able to use them in appropriate situations: Create legible, efficient programs that take advantage of Python's built-in capabilities and adhere to					

	Python's best practices; Recognize the variations in implementation and performance considerations associated with various Python data structures; Handle missing or inconsistent values, manipulate, and analyze huge datasets; Create, test, and implement Python applications for data analysis and visualization, online data extraction, and database interactions.					
Course Content						
CPRG210	**PROGRAMMING FOR ANDROID**	(3, 0 ,0)3	6	AE	-	English
Course Content	The course covers the design and development of mobile applications utilizing the open-source Android platform. It will be a lecture and laboratory course that will help students grasp the philosophy of programming for Android by examining its core application development building elements and how they interact with one another. This course encourages students to learn by developing increasingly sophisticated and meaningful Android mobile applications. Upon completion of this course, each student will have created their own entire Android application, incorporating the majority of the platform's important features.					
CPRG211	**PROGRAMMING FOR IOS**	(3, 0 ,0)3	6	AE	-	English
Course Content	The course covers the design and development of mobile applications utilizing the Open-Source iOS Libraries. It will be a lecture and laboratory course that will help students grasp the philosophy of programming for iOS by examining its core application development building elements and how they interact with one another. This course encourages students to learn by developing increasingly sophisticated and meaningful iOS mobile applications. Upon completion of this course, each student will have created their own entire iOS application, incorporating the majority of the platform's important features.					
CPRG212	**JAVA FOR HANDHELD TERMINALS**	(3, 0 ,0)3	6	AE	-	English
Course Content	This course covers computer programming with object-oriented programming ideas using the Java programming language. The course emphasizes event-driven programming approaches, database connectivity, socket programming, and distributed programming, as well as the creation and manipulation of objects and classes and the use of object-oriented tools such as the class debugger. After completing this course, the student will be able to: design, create, build, and debug Java applications and applets; write Java programs using object-oriented programming techniques such as classes, objects, methods, instance variables, composition, inheritance, and polymorphism; and write programs using graphical user interface (GUI) components and Java's Event Handling Model.					
CPRG214	**VISUAL PROGRAMMING**	(3, 0 ,0)3	6	AE	-	English
Course Content	This course covers computer programming concepts through the use of the Visual Programming Language and object-oriented programming techniques. Students will learn how to use advanced VB.NET and.NET Framework technologies to create sophisticated, scalable, and high-performance applications. Students will design a Visual Basic project using inheritance, interfaces, and polymorphism. Students will build attractive ASP.NET web and Windows user interfaces. Students will learn how to use LINQ to integrate SQL or other databases into Visual Basic. Students will also learn how to deploy Windows and ASP.NET applications.					
CPRG215	**SOFTWARE TESTING**	(3, 0 ,0)3	6	AE	-	English
Course Content	The goal of this course is to teach students about software validation and testing concepts and theories. It is primarily concerned with examining whether a software system meets specifications and requirements so that it fulfills its intended purpose. White box, black box, integration, system and acceptance, performance, regression, object-oriented, usability, and accessibility testing will be covered. Students who successfully complete the course will be aware of a wide range of software testing techniques and have the ability to apply the right techniques in the process of software validation and testing.					
CPRG216	**SOFTWARE DEVELOPMENT**	(3, 0 ,0)3	6	AE	-	English
Course Content	The objective of this course is to generate dependable, safe, and effective software products by focusing on software product development. This involves looking at the general organization of the software's development and release phases, how the software is broken down into components, how the servers are organized, and the technologies that were utilized to create the software. With a focus on the practical concerns inherent in software project management, students will master the fundamentals of software architectural designs, patterns, and views. In addition, a brief introduction to microservices architecture and cloud-based applications will be covered.					
CPRG217	**COMPUTER GRAPHICS APPLICATIONS**	(3, 0 ,0)3	6	AE	-	English
Course Content	The primary goal of this subject is to introduce students to computer graphics concepts. It begins with an overview of interactive computer graphics, two-dimensional systems, and mapping before moving on to the most important drawing algorithms, two-dimensional transformation, clipping, filling, and an introduction to three-dimensional graphics. The course discusses strategies for rendering a complicated scene efficiently in real-time. The course covers the following topics: As a rendering architecture, the shader development languages at the highest level Algorithms for rendering shadows, reflections, and refraction efficiently and techniques for increasing speed, such as culling and level-of-detail detection of intersections and collisions.					
CPRG218	**CLOUD COMPUTING APPLICATIONS**	(3, 0 ,0)3	6	AE	-	English
Course Content	This course focuses on the use of the most popular cloud computing applications and services that run on a distributed network using virtualized resources and are accessed by common Internet protocols and networking standards. Its architecture, abstraction, virtualization, infrastructure, scaling deployments, machine learning in the cloud, data management, security, and privacy in the cloud will be discussed in detail. On successful completion of this course, students should be able to: Explain Cloud Computing abstraction and virtualization; Describe cloud storage services, pros and cons; Use different cloud storage services; Work with cloud APIs and SDKs; Describe machine learning in the cloud; Secure data in the cloud; and Build their own cloud with open stack.					