ISTINYE UNIVERSITY FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE

COURSE DESCRIPTIONS

1. Semester

Differential and Integral Calculus (2+2), 6

Analytic geometry, functions and limits, derivatives, techniques and applications of differentiation, logarithmic and trigonometric functions. Definite and indefinite integrals, techniques of integration, with applications in sciences and engineering.

Basic Programming 1 (Python) (3+2), 6

This course aims to prepare students for computational thinking as well as software development by using Python programming language. In addition to the elements that make up the Python language, the programming logic and the fundamentals of software development will be taught.

Engineering Physics (3+1), 6

Vectors. Motion in one and two dimensions. Newton's laws and its applications. Work and energy. Conservation of mechanical energy. Momentum and motion of systems. Static equilibrium of rigid bodies. Rotation and angular momentum. Newton's law universal gravitation.

Unix Operating System (3+1), 6

In this course, students study the fundamentals of the UNIX operating system, such as file editing, data retrieval, executing processes and creating directories. An introduction to shell scripts may be covered depending on the program.

Turkish 1 (2+0), 2

To teach the importance of language in human and social life; showing the classification of languages and teaching of Turkish among the world languages; to teach the characteristics of Turkısh in terms of phonology, morphology, syntax; to analyze spelling and writing rules of Turkish.

General English 1 (2+0), 2

The English of the terms and concepts encountered in various branches of engineering are examined in depth, and for the proper use of these terms, Turkish-English bilingual translations are explained. English language training for students starts from simple forms and intended to develop over time.

2. Semester

Computational Mathematics (3+2), 7

Introduction to the theoretical underpinnings of computer science and engineering. Topics include propositional predicate logic, set theory, functions and relations, counting, mathematical induction, recursion, generating functions, graph theory and algorithms. Fundamentals of groups, rings, fields, number theory and cryptography. Computational tools such as Python, Sage, and Mathematica will be be used to illustrate properties the mathematical objects and structures.

Basic Programming 2 (Python) (3+2), 7

The main goal of this course is to make students practice the programming skills that they acquired in Basic Programming I course. In particular, the course aims to teach widely used problem solving methodologies on real life examples. Course content spans from filtering spam to recommending movies, books and music to end-users, the course content is mainly based on practical applications. Practical data mining techniques (classification, clustering, etc.), collaborative filtering techniques, document retrieval and ranking, advanced Python concepts (lambda functions, iterators, generators, decorators, etc.) will be taught. The students will be able to apply learned data mining techniques on real life cases.

Unix Programming (3+1), 6

The course discusses the basic components of programming practice: testing, debugging, portability, performance, design alternatives and style. It also addresses the importance of simplicity, clarity and generality, the basic principles of programming process, which apply at all levels of computing. We stress the basic UNIX programming philosophy, based on the understanding of the relationship between programs. In other words one needs to know not only how to use programs but also how they fit into the environment. Parallel to discussing the features of languages we will deal with the principles of programming practice, the main theme of the course. Our main teaching objective is to apply these principles in practical tasks.

Electronics Physics and Systems (3+1), 6

In this course the principles of electrical and electronical engineering and basic technology will be introduced. Application examples of the knowledge on engineering will be given. In addition, electric machinery, power electronics, and electrical driving circuits will be introduced according to the mechanical engineering requirements

Turkish 2 (2+0), 2

To teach spelling, writing and punctuation rules, to teach basic elements of writing essays, to introduce Turkısh and World literary canons; to teach writing creative texts of literature especially story, poem and essay ; to teach writing scientific paper and texts; To analyse expression and punctuation disorders, to contribute lectures.

General English 2 (2+0), 2

The English terms and concepts encountered are examined in depth and Turkish-English bilingual translations are used in order to use the concepts correctly. To be able to master professional English language, the students are informed about grammatical structures of sentences, spelling and pronunciation.

3. Semester

Algorithms 1 (3+1), 6

This course provides an introduction to mathematical modeling of computational problems. It covers the common computer science algorithms, including searching, sorting, recursion, and graph theory. The course emphasizes the relationship between algorithms and programming.

Basic Data Structures 1 (3+1), 6

The course covers basic data structures like lists, stacks, queues, trees, diagrams and heaps .The aim of the course is to enable students to have knowledge and skills in designing, analyzing and developing basic data structures.

Theory of Programming and Practice 1 (C) (3+2), 7

The course fully covers the basics of programming in the “C” programming language and demonstrates fundamental programming techniques, customs and vocabulary including the most common library functions and the usage of the preprocessor.

Linear Algebra (2+2), 6

Systems of linear equations, matrix algebra, determinants, vector spaces and subspaces, basis and dimension, linear transformations, eigenvalues and eigenvectors, diagonalization, and orthogonality; singular-value decomposition.

Ataturk's Principles and History of Revolution 1 (2+2), 6
The Collapse of the Ottoman Empire, Tanzimat and Reform Edict, I. and II. Constitutional Era, Tripoli and Balkan Wars, Worl War I, Mudros Armistice, War of Independence; Amasya Circular, National Congresses, Establishment of Turkish Grand National Assembly, Declaration of Republic

Manifest of Istinye 1 (0+1), 1

4. Semester

Algorithms 2 (3+1), 6
This course aims to give students the ability to solve any computing science problem using the techniques discussed in class. By the end of the course, you will have learned basic algorithm techniques including brute-force, greedy, divide and conquer, dynamic programming, and linear programming.

Basic Data Structures 2 (3+1), 6
This course presents advanced data structures used to represent complex data; enhances programming skills via implementation of these data structures, along with algorithms that apply to each; includes advanced uses of arrays and linked lists, as well as coverage of trees, priority queues, heaps, and graphs.

Theory of Programming and Practice 2 (C++) (3+2), 7
The goal is to teach the basics about C++ programming language such as variables, data types, arrays, pointers, functions and classes etc. At the end, the students are expected to have a good understanding about the concept of object-oriented programming using C++, be able to write and read basic C++ code.

Ataturk's Principles and History of Revolution 2 (2+0), 2
Revolutions in Turkish politics, political parties and multi-party system attempts, revolutions in law, reorganization of public life, reforms in economics, Turkish foreign policy 1923-1938, Turkish domestic and international politics in post-Atatürk era.

Manifest of Istinye 2 (0+1), 1

5. Semester

Operating Systems (3+1), 6
Issues such as process management, file management, memory management, input / output management, etc. constitute the basic knowledge that should be transferred to the students detailly. The basic structures of operating systems (virtual machines, paging, interprocess communication and synchronization, device drivers, etc.) are covered in this course. Additionally, problems encountered, and possible solutions devised while designing an operating system are explained at the entry level.

Principles of Machine Intelligence (3+2), 7
This course provides an overview of the core principles of machine learning and artificial intelligence. The goal is to gain an understanding of the basic principles of machine learning and artificial intelligence from both an intuitive and practical level and an understanding of common feature design principles for image and text data. Moreover, the students learn how to use popular machine learning and deep learning software packages in Python, as well as how to implement several popular machine learning algorithms (Linear/Logistic Regression; KMeans Clustering) from scratch. They are expected to have extensive experience applying machine learning algorithms to real data sets.

Theory of Programming and Practice 3 (C++) (3+2), 7

Further application of C++ programming techniques including subjects such as file access, abstract data structures, class inheritance, and other advanced techniques. The following C++ programming topics are covered: classes, objects, function and operator overloading, inheritance and dynamic polymorphism, templates, exception handling, reference counting, complex data structures, complex input/output standard and file handling techniques, program documentation and other advanced C++ techniques.

Manifest of Istinye 3 (0+1), 1

6. Semester

Theory of Programming and Practice 4 (iOS+Android) (3+2), 7

This course introduces you to the design and implementation of iOS and Android applications for mobile devices. You will develop an app from scratch, assuming a basic knowledge of Java, and learn how to set up Android Studio and XCode, work with various activities and create simple user interfaces to make your apps run smoothly.

Neural Networks and Machine Learning (3+2), 7

The aim of this course is to enable students to learn about artificial neural networks and how they're being used for machine learning, as applied to speech and object recognition, image segmentation, modeling language and human motion, etc. The basic algorithms and the practical tricks needed to get them to work well will be emphasized.

Manifest of Istinye 4 (0+1), 1

7. Semester

Deep Machine Learning (3+2), 7

The aim of this course is to introduce the students with knowledge representation, problem solving and learning methods which form the basis of machine intelligence. Students who successfully complete this course will be able to understand the importance of linguistics in understanding human intelligence, develop intelligent systems by developing solutions to concrete computational problems, understand the concept of knowledge representation in intelligent systems engineering.

Computer Security (3+1), 6

This course covers the concept of information security at the beginning level, cryptographic algorithms and systems, as well as the creation of protection mechanisms using the techniques and approaches required to achieve network security. Students who complete this course successfully will have learned security algorithms, cryptographic algorithms, and simple encryption systems, electronic signatures, together with secure messaging and authentication topics.

Capstone Project 1 (3+2), 7

Student groups design a significant computerbased project. Multiple groups may cooperate toward one large project. Each group works independently; interaction among groups is via interface specifications and informal meetings.

8. Semester

Network Programming (3+1), 6

The aim of this course is to familiarize the students with the general concepts of network programming and to give experience on network programming. Topics include; Introduction to network layers, TCP and UDP socket programming, client and server-side programming, consistency, latency, scalability, security and compression algorithms.

Theory of Computer (3+1), 6

This course is an introduction to the theory of computing. Topics include design of finite state automata, pushdown automata, linear bounded automata, Turing machines and phrase structure grammars; correspondence between automata and grammars; computable functions, decidable and undecidable problems, P and NP problems, NP-completeness, and randomization.

Capstone Project 2 (3+2), 8

Student groups design a significant computerbased project. Multiple groups may cooperate toward one large project. Each group works independently; interaction among groups is via interface specifications and informal meetings.

DEPARTMENT ELECTIVE COURSE DESCRIPTIONS

Syber-Physical System Security (2+2), 5

This course provides an introduction to security issues relating to various cyber-physical systems including industrial control systems and those considered critical infrastructure systems. The students will learn the state of the art of designing protections for cyber-physical systems like the power grid, industrial control networks, and autonomous vehicles.

Computational Biology (2+2), 5

This course covers the algorithmic and machine learning foundations of computational biology combining theory with practice. It will be covered both foundational topics in computational biology, and current research frontiers. The students will learn fundamental techniques, recent advances in the field, and be able to work directly with current large-scale biological datasets.

Embedded Sytems (2+2), 5

The object of this course is to gain information about the structure of embedded systems and ability of programming embedded system by using cross compiler. This course will provide the students with the ability of programming and develepment of embedded systems used in mobile phones, modems, etc.

Parallel Computers (2+2), 5

Subjects of this course: Modern computer elements, evolution of computer architectures, Flynn classification, multiprocessors and multiple computers, memory sharing multiprocessors, memory multiple computers, taxonomy of MIMD computers, distributed shared memory, message passing architectures, networked computers as multi-SIMD architectures, sequential computations and computing platforms for sequential applications, linear sequential processors, and nonlinear sequential processors.

Theory of Programming Languages (2+2), 5

In this course students will have a general knowledge of the basic concepts of syntactic and semantic structures of programming languages through comparative analysis of several programming languages. Additionally, awareness will be given to the students on the characteristics of several programming languages so that the gain the ability to understand general conceptual grammatical issues necessary for designing new programming languages and compilers.

Compliers (2+2), 5

This course explores the implementation of modern programming languages by looking at compiler design and construction. The course focuses mainly on object-oriented programming languages, although it also looks briefly at compilation of languages from other programming paradigms. Major topics in compilation are covered, including scanning, parsing, semantic analysis, and code generation. Time permitting, the course also covers some advanced topics, including garbage collection and optimization.

Real-Time Systems (2+2), 5
The topics covered in this course: Theory, algorithmic and protocol concepts, mechanisms, and implementations of real-time computer systems. Introduction to real-time systems, real-time scheduling, real-time synchronization, real-time operating system kernels, and real-time programming languages. Design and analysis of real-time resource management algorithms (e.g., scheduling, synchronization), their implementations in production operating system kernels, experimental studies of those implementations, and real-time application development.

Computer Games (2+2), 5
This course will help the students to develop strong technical skills suitable for professional programming roles in the game industry, specialist knowledge in computer graphics, AI, physics and audio, the ability to design and build game engines from scratch in industry standard languages, including C++, and knowledge of the games development process, including the pitch, design, and use of a game engine to build a demo.

Multi Media Programming (2+2), 5
This course aims to teach the design and programming of computer-based interactive products that incorporate text, graphics, sound, animation and video.

Computer Architecture (2+2), 5
This course emphasizes today's computer architectures, especially the basic principles modern computer architecture, and the critical role of performance in computer design. Topics to be covered include number systems, computer arithmetic, evolution and performance of computers, memory, storage, input / output, processors, multi-core processors and clustered computer structures.

Data Mining (2+2), 5
The aim of this course is to teach data mining techniques for both structured data which conform to a clearly defined schema, and unstructured data which exist in the form of natural language text. Specific course topics include pattern discovery, clustering, text retrieval, text mining and analytics, and data visualization.

Distributed Systems (2+2), 5
This course introduces students to the distributed system algorithms and technologies. The students will learn the structure, types and usage areas of distributed systems. They will be able to make the infrastructure, software, hardware, language and operating systems applicable in distributed system and parallel computing. Distributed system process structures which are client, server, multiplex (thread) and code handling approaches, and process communication, protocol layers, remote process execution and naming, distributed processing and algorithm synchronization approaches in distributed structures will be learned.

Artifical Vision (2+2), 5
The course covers the physics of image formation, motion vision, and recovering shapes from shading. Binary image processing and filtering are presented as preprocessing steps. Further topics include photogrammetry, object representation alignment, analog VLSI and computational vision.

Advanced Web Programming (2+2), 5
This course is designed to give students the opportunity to enhance and enrich their skills in Web programming. Students will learn to develop Web applications that use three-tier architecture, session management, object-oriented techniques, and advance database interactions. Concepts such

as advanced CSS concepts, rich interactive Web environments, authentication, and security will also be explored.

Advanced Cryptography (2+2), 5
This course investigates advanced topics in cryptography. It begins with an overview of necessary background in algebra and number theory, private- and public-key cryptosystems, and basic signature schemes. The course will cover number theory and basic theory of Galois fields used in cryptography; history of primality algorithms and the polynomial-time test of primality; discrete logarithm based cryptosystems including those based on elliptic curves; interactive protocols including the role of zero-knowledge proofs in authentication; construction of untraceable electronic cash on the net; and quantum cryptography.

Advanced Network Programming (2+2), 5
Subjects to be covered: Overview of TCP/IP layers, transport layer API, network programming issues, related RFCs and standards. Socket programming : UNIX sockets, IPC issues, multiplexing, Winsock sockets. XML. . TLI programming : Fundamentals, STREAMS subsytem, drivers, modules. . RPC programming : Basics, XDR, high/low level programming, rpcgen. WML. Perl programming : How to run PERL programs, program elements, operators etc, loops, i/o, pipes, system calls, matching, parsing, CGI applications. Java programming : SDK, writing applets, classes and JFC, GUI development with AWT/Swing, graphics and sound. Java Beans, JDBC database, servlets, security. . Programming of mobile agents, then distributed computing and CORBA (RMI). PHP.

Image Processing (2+2), 5
This course is an introduction the general principles of image processing. Areas examined include: Image sampling and quantization, color, point operations, segmentation, morphological image processing, linear image filtering and correlation, image transforms, eigenimages, multiresolution image processing, noise reduction and restoration, feature extraction and recognition tasks, image registration. Students learn to apply material by implementing and investigating image processing algorithms in Matlab and optionally on Android mobile devices.

Cryptography (2+2), 5
This course features a rigorous introduction to modern cryptography, with an emphasis on the fundamental cryptographic primitives of public-key encryption, digital signatures, pseudo-random number generation, and basic protocols and their computational complexity requirements.

Computer Graphics (2+2), 5
The aim of this course is to enable students to learn three-dimensional computer graphics concepts and user interfaces and to be able to use advanced imaging techniques effectively, to gain the ability to use advanced graphic programming interfaces (OpenGL etc.) and user interfaces effectively, to create fast and high-quality animations and graphics. They are expected to gain the ability to design and use the necessary data structures in an effective and correct way, and to be exposed to two and three-dimensional game programming.

Database Systems (2+2), 5
This course covers the fundamental concepts of database systems. Topics include data models (ER, relational, and others); query languages (relational algebra, SQL, and others); implementation techniques of database management systems (index structures, concurrency control, recovery, and query processing); management of semistructured and complex data; distributed and noSQL databases.