Course Descriptions of Computer Engineering Bachelor Degree in Faculty of Engineering in Ferdowsi University of Mashhad

Course: **General Foreign Language: English**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **None**
Syllabus:
The aim of this course is teaching general English language to students.

Course: **Physics I**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **None**
Co-Requisite: **General Mathematics I**
Syllabus:
Measurement, vectors, one-dimensional motion, motion in one plane, dynamics of particle, work and energy, energy preservation, dynamics of particles systems, rotary kinematics, balance of solid bodies, temperature fluctuation, heat, 1st thermodynamics law, gas kinetic theory, and 2nd thermodynamics law.
References:
Fundamentals of physics, by D. Halliday & R. Resnick (1986), John Wiley & Sons, Inc.

Course: **General Mathematics I**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **None**

Syllabus:

Cartesian coordinates; polar coordinates; complex numbers; summation, multiplication, radical and geometrical representation of complex numbers; polar representation of complex numbers; function; algebra of functions; limit and related theories; infinite limit and limit at infinity; left-hand and right-hand limit; continuum; derivative; derivative rules; inverse function and its derivative; derivative of trigonometric functions and their inverse functions; Roll's theorem; mean; value theorem; geometrical and physical applications of derivative; curves and acceleration in polar coordinates; application of derivative in approximation of equations roots; definition of integral of continuous and piecewise continuous functions; fundamental theorems of differential and integral calculus; primary function; approximation estimate methods of integral; application of integral in calculation of surface area and volume and curve length and momentum and center of gravity and work, etc. (in Cartesian and polar coordinates); logarithm and exponential function and their derivatives; hyperbolic functions; integration methods such as change of variables and by parts and partial fractions decomposition; special variable replacement of sequence, numerical series and convergence theories; power series and Taylor theorem with residual, Taylor expansion.

Course: **Computer Lab**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **None**
Syllabus:

- Being familiar with main fields of computer engineering (each field is presented by an experienced professor in that field).
- Participating in Computer Assembly workshop (2 sessions).
- Working on a practical project in the favorite field (described in previous sessions).

Course: **Fundamentals of Computer and Programming**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **None**
Co-Requisite: **Computer Workshop**

Syllabus:

- Primary concepts of computer and its role in the modern world and applied examples, introduction to main parts of computer and its media (hardware and software), numerical systems in computers, numerical (fixed point, floating point) and non-numerical data representation, introduction to machine language (with an assumption language having 10 instructions and orders), concept of algorithm.
- Principles of algorithm design (sequence, selection and repeat) and problem solving, algorithm presentation using pseudo code, introduction to an organized programming language, constants, variables, computational and logical phrases, types of instructions, types of loops, conditional operations, vectors and matrices, subprograms (functions and procedures), input and output instructions, common algorithm such as methods of search and sort, introduction to advanced principles of program design.

References:

1- T. C. Bartee, Digital Computer Fundamental, McGraw Hall, 1981.
2- A. Catlin, Pascal for Engineers and Scientists with Turbo Pascal, Prentice Hall, 1990.
3- R. Bornat, Programming from First Principles, Prentice Hall, 1986.
4- A. Behfrooz and Onkar P. Sharma, An Introduction to Computer Science: A Structured Problem Solving Approach, 1985.

Course: **Computer Workshop**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **None**
Co-Requisite: **Fundamentals of Computer and Programming**
Syllabus:

- Developing Programs and practicing concepts presented in Fundamentals of Computer and Programming course.

References:

1- T. C. Bartee, Digital Computer Fundamental, McGraw Hall, 1981.
2- A. Catlin, Pascal for Engineers and Scientists with Turbo Pascal, Prentice Hall, 1990.
3- R. Bornat, Programming from First Principles, Prentice Hall, 1986.
4- A. Behfrooz and Onkar P. Sharma,An Introduction to Computer Science:A Structured Problem Solving Approach,1985.

Course: **Physics II**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Physics I**
Co-Requisite: **General Mathematics II**
Syllabus:

Charge & matter, electrical field, Gauss law, electrical potential, capacitors and dielectric, current & resistance, electrical kinetics and circuits, magnetic field, Ampere's law, Faraday induction law, matter magnetic properties & oscillations, alternate currents, Maxwell equations, electromagnetic waves.

References:

1- Fundamentals of physics, by D. Halliday & R. Resnick (1986), John Wiley & Sons, Inc.

Course: **Basic Physics II Lab**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **Physics II**
Syllabus:

Practical applications and experiments according to the syllabus presented in physics II.

Course: **General Mathematics II**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **General Mathematics I**
Syllabus:

Parametric equations; space coordinates; vector in space; scalar product; 3×3 matrices; linear equations system with 3 unknowns; operation on rows; inverse matrix; solving of linear equations system; linear independence; base in R^2, R^3;linear transformation and its matrix; 3×3determinant;characteristic vector and value; vector product; equations of line and plane; second order surface; vector function and its derivative; velocity and acceleration; curvature and normal vectors on curves; multivariable functions; total and partial derivative; tangent plane and normal line; gradient; chain rule for partial derivative; exact differential of double and triple integrals and their applications in geometrical and physical problems; change of variable in integration (without proof of accuracy); spherical and cylindrical coordinates; vector field; curvilinear integral; surface integral; divergence; curl; Laplacian; potential, Green and Stokes and divergence theorems.

Course: **Advanced Programming**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Fundamentals of Computer and Programming**
***Co-Requisite:* Advanced Programming Workshop**
Syllabus:

- Introducing the plan and presenting the course: life cycle, importance of programming, design methods and object-oriented.
- Object-oriented programming, object and its definition, diagnosis of objects in a problem and relations, and object-oriented languages, history and introduction to Java.
- Review of one of non-object-oriented routine language (C), class, polymorphism, heritage, program sample.
- Class: class and object relation, encapsulation and abstraction, different components of a class (private, public, protected), class intermediate, constructive and destructive, derived ranks and classes.
- Polymorphism and its necessity, Polymorphism of functions and operators.
- Heritage, usage, unique heritage and multiple heritage.
- Template: necessity of template, along with several examples, function templates, separation and template.
- Other facilities of Java programming.
- Other models for programming introduction and reviewing, illustrative and imaginary programming, windows, server and client programming.
- Test and documentation: necessity, automatic tools of test, documents on program, technical documents, and help document for users.

References:

1- Object Oriented Programming in Java for C Programmers, Paydar, Zarrin Kalam
2- Java, How to Program? by: Deitel & Deitel
3- Thinking in Java, by: Bruce Eckel R. S. Wiener, L. J. Pinson.

Course: **Advanced Programming Workshop**
***Credits:* 1 - *ECTS:* 3**
***Type of Course:* Practical**
***Prerequisite:* None**
***Co-Requisite:* Advanced Programming**
Syllabus:

- Developing Programs and practicing concepts presented in Advanced Programming course.

References:

1- Object Oriented Programming in Java for C Programmers, Paydar, Zarrin Kalam
2- Java, How to Program? by: Deitel & Deitel
3- Thinking in Java, by: Bruce Eckel R. S. Wiener, L. J. Pinson.

Course: **Discrete Mathematics**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Fundamentals of Computer and Programming, General Mathematics I**
Syllabus:

- Introduction: mathematical logic, algebra of expressions, well-structured formula, a review of theory of sets, proof methods.
- Relations and functions: binary relations, compatibility and equivalence relations, relations representation matrix, graphs of relations, functions, coverage functions, one to one functions.
- Recursive functions, deduction, generator functions.
- Algebraic structures: semi-groups and monoids, grammars and languages, Polish marking, groups, homomorphism,

isomorphism, lattices, Boolean Algebra, Carnot table, grammar and language, grammar as an example of monoids

- Combinatorial analysis, nest principles, introduction to combinatorial algorithm, regressive and reciprocal functions and applications.
- Graph theory: directional graphs, unidirectional graphs, Aurelian path and Hamiltonian path, optimal paths, finding algorithm for optimal paths, connected graphs, matrix of relation and related theorems, application of graphs in activity analysis.
- Trees: minimal overlapping trees, surveying of trees, application of trees, algebraic expressions and representation of their trees.

References:

1- R. Johnson Baugh. Discrete Mathematics, Macmillan Pub. Company, 1997.
2- W. K. Grassman and J. P. Tremblay, Logic and Discrete Mathematics: A Computer Science Perspective, Prentice Hall, 1996.
3- J. P. Tremblay, Discrete Mathematical Structures with Applications to Computer Science, McGraw-Hall, 1988.
4- Keneth. A, Rossen and Charles R. B., Wright, Discrete Mathematics, Third Edition, Prentice Hall, 1992.
5- Ralph P. Grimaldi, Discrete and Combinational Mathematics: An Applied Introduction, 1989.
6- Michael O. Alberison, Joan P. Hvtchinson, Discrete Mathematics with Algorithms, John Wiley, 1988.

Course: **Differential Equations**

Credits: **3 -** ***ECTS:*** **6.9**

Type of Course: **Theoretical**

Co-Requisite: **General Mathematics II**

Syllabus:

Nature of differential equations and solving, family of curves and normal trajectories, physical models, separable equation, first-order linear differential equation, homogeneous equation, second-order linear equation, homogenous equation with constant coefficient, method of undetermined coefficients, parameter changing method, application of second-order equations in physics and mechanics, solving differential equation with series, Bessel and Gamma functions, Legendre' polynomial, introduction to differential equations system, Laplace transformation and its application in solving differential equations.

Course: **Engineering Statistics and Probability**

Credits: **3 -** ***ECTS:*** **6.9**

Type of Course: **Theoretical**

Prerequisite: **General Mathematics I**

Syllabus:

Introduction to set theories, samples and table representation with mean, power, variance, conversion & probabilities combination with related theorems, intermediate random variables, average & variance, distributions, binomial Poisson' distribution, geometric difference, normal distribution, multivariate random distribution, random sampling and random numbers, sampling from small society, estimation of statistical parameters, confidence interval, hypothesis test of decision-making, assumption test, variance experience regression, correlation test, nonparametric methods, direct data fitting line, momentum generator functions, large number theorem, central limit test, sum of independent random variables, conditional probability, total probability theorem.

Course: **Logic Circuits**

Credits: **3 -** ***ECTS:*** **6.9**

Type of Course: **Theoretical**

Prerequisite: **None**

Co-Requisite: **Discrete Mathematics**

Syllabus:

Number representation system and coding, representation of negative numbers, key logic, negative test and triple state logic, overall structure of logical gates and its types, logic functions and simplification of them including: Carnot's methods and scheduling and tabulation method, computerized procedures for simplification of combination functions, decoding and coding circuits design, code converters, subtractions, summation, selectors, and comparators, collectors and destructors, logical and computational units, use of decoders, selectors and other packages for drawing combinatorial circuits, PAL, PLA & ROM, and other regular structures, leach and flip-flaps structures, synchronous circuits, comparison between state circuits under Moore & Mealy, numerators, shift registers, asynchronous circuits, study of hazards and race, allotting state without race, common chips in, sequential circuits, and designing or study of one type of machines or control and data section, modern designing methods.

References:

1- Victor P. Nelson, H Troy Nagle, Bill D. Carroll and David Irwin, Digital Logic Circuit Analysis & Design, Prentice-Hall Inc., 1996.
2- John F. Wakerley, Digital Design Principles and Practices, Prentice-Hall, 1993.
3- M. Morris Mano, Computer Engineering Hardware Design, Prentice-Hall, 1992

Software:

- Gate Level Schematic Capture and Simulation
- Language based Simulation Program

Course: **English for the Specific Purposes (Technical English)**
Credits: **2 -** ***ECTS:*** **4.6**
Type of Course: **Theoretical**
Prerequisite: **General Foreign Language: English**
Syllabus:

This subject aims at raising students' specific language ability in reading and writing academic texts of their own major disciplines. The subject will use reading texts from chapters of books or journal articles recommended by teachers of different majors for reading comprehension. These texts will also be used for analysis to enable students to develop an awareness of the genre in that particular discipline.

Course: **Data Structure**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Advanced Programming, Discrete Mathematics**
Syllabus:

Arrays, vectors, matrices and applications such as MAZE, matrices, arrays display, stacks and queues and their applications, linked lists (linear, loop, double linkage, multi-link) and their applications, definitions and basic principles of trees, method of representation and application of trees: decision making trees, binary trees, search trees, tree of the game and etc., graphs (representation, surveying and applications), coverage trees, dynamic memory allocation and their comparison, searching algorithms, sorting and merging algorithms

- Two hours in a week is allocated for problem solving in this course.
- Each chapter should include theoretical and programming exercises.

References:

1- E Horowitz and S. Sahni, Fundamentals of Data Structures and Computer Algorithms, Computer Science Press, 1995.
2- A. M. Tanenbaum, Data Structures Using Pascal, Prentice Hall, 1986.
3- N. Wirth, Algorithms + Data Structures = Programs, Prentice Hall, 1988.

Course: **Computer Architecture**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Logic Circuits**
Syllabus:

Definition of Computer architecture, a reference to history of computer and its generations, introduction to main units of computer, design of a series of instructions, assessing criteria and issues, mechanism of execution of instructions with hardware description language i.e. RTL, design methods of control unit by wired method, control unit structure, control of different types of buses and data routing, ALU unit design and specify the latencies, micro-procedure control unit design, memory organizations and its hierarchy, static and dynamic memories, introduction to cache and virtual memory, computational algorithms (add, subtraction, multiplication, division), floating point algorithms, access methods to IO devices (interrupt and so on), direct memory access (DMA), BUS sharing, introduction to developing trend in computers and differences between CISC and RISC.

Course: **Logic Circuits and Architecture Lab**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **Computer Architecture**
Syllabus:

Practical applications according to the Computer Architecture course.

***Course:* Algorithm Design**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Data Structure**
Syllabus:

Review of essential points of Data Structures and its completion by mathematical deduction and regression methods, complexity of algorithms and analysis of them, symbols O, o, Ω, θ, problem solving methods, divided & conquer (maximum and minimum of an array, matrices product with Strassen method, triangulation of polygonal, plays tournaments, quicksort algorithm), dynamic method of programming, greedy algorithms (problems of timing, money crash, Huffman code), full searching methods, and limitation of searching space, α-β pruning (puzzle, tic-tac-tac Games), graph algorithms including deep and superficial, Dijkstra algorithms, minimal covering tree, correlative components, side graphs, Floyd algorithms, topological sorting of components of 2 connections, maximum flow network and related problems.

References:

1- R. E. Neapolitan and K. Naimipour, Foundations of Algorithms Using C++ Pseudo Code, Second sedition, Jones and Barlett publishers, 1998.
2- Corman, Leisersen and Rivert, Introduction to Algorithms. MIT Press, 1990.
3- E. Horowitz and S. Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 1987.
4- Aho Hopetoft, Ullman, Data Structures & Algorithms, Addison-Wesley, 1985.
5- Udi Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley, 1987.
6- G. Brassard and P. Bratley, Fundamentals of Algorithms, Prentice-Hall, 1996.

***Course:* Theory of Languages and Machines**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Data Structure**
Syllabus:

Finite automata, Pushdown automata, touring machine, different types of grammars and languages, Chomsky classification, relation between languages and machines and the relevant theorems.

References:

1- P. Linz, Introduction to Formal Languages and Automata, Second Edition, D. C. Heath Company, 1996.
2- D. Wood, Theory of Computation, Prentice-Hall, 1986.
3- P. Reveseze, Theory of Formal Languages, McGraw-Hill, 1985.
4- D. I. A. Cohen, Introduction to Computer Theory, John Wiley & Sons, Inc., 1991.

***Course:* Research Methods and Presentation**
***Credits:* 2 - *ECTS:* 4.6**
***Type of Course:* Theoretical**
***Prerequisite:* English for the Specific Purposes**
Syllabus:

Different types of scientific and technical subjects (letters, reports, pamphlets, manual and etc.), common points in all scientific and technical writings: specifying the objective of writing and its eventual readers, organizing the subjects, abstract of essay together with report, the role of a good introduction, dividing the subjects into parts and chapters, discussion and conclusion, preparing source and reference index, attachments, preparing the pictures and diagrams and tables. Important points in translation of scientific and technical subjects, writing style, marking and its importance, preparing final format of writing by typing machine or computer, foot-article, notes and other lateral subjects, an introduction to research methods, presenting subjects orally, effective use of audio-visual devices, the rules and process of drawing up graduation diploma including the main parts of thesis and details of each part, preparing and presenting a scientific essay (as assignment).

***Course:* Applied Linear Algebra**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* General Mathematics II**
Syllabus:

- Linear equations and Gaussian elimination, matrix operations
- Factors, inverses and transposes; special matrices
- Vector spaces, linear independence and dimension

- Subspaces, linear transformations, applications to networks
- Orthogonality and projections; Gram-Schmidt orthogonalization
- Determinants: theory and applications
- Eigenvalues, eigenvectors, diagonalization, matrix exponentials
- Positive definite matrices, second derivative test
- Singular value decomposition, Rayleigh quotient
- Linear programming
- Duality, game theory
- Computations, norm and condition number

References:

1- Linear Algebra and Its Applications, 4th edition, by Gilbert Strang. Published by Thompson Brooks

***Course:* Electrical and Electronic Circuits**

***Credits:* 3 - *ECTS:* 6.9**

***Type of Course:* Theoretical**

***Prerequisite:* None**

***Co-Requisite:* Basic Physics II Lab, Differential Equations, Electrical and Electronic Circuits Lab**

Syllabus:

- Characteristics of ideal diode, π type and P type semiconductor, linkage between Pπ, electrical characteristics of actual diode, diode rectifier circuits, diode jumping and chopper circuits, Zener' diode specifications, Zener voltage modulator.
- BJT bipolar transistor, structure and electrical behavior, biasing of BJT (dc analysis), small signal model, BJT amplifier, CB, CE, CC amplifier (small signal analysis), multi layers amplifiers and magnitude.

Field affect transistor, structure and electrical behavior, introducing CMOS, biasing FET, FET amplifiers.

***Course:* Electrical and Electronic Circuits Lab**

***Credits:* 1 - *ECTS:* 3**

***Type of Course:* Practical**

***Prerequisite:* None**

***Co-Requisite:* Electrical and Electronic Circuits**

Syllabus:

Practical applications and experiments according to the syllabus presented in Electrical and Electronic Circuits.

***Course:* Computer Networks**

***Credits:* 3 - *ECTS:* 6.9**

***Type of Course:* Theoretical**

***Prerequisite:* Engineering Statistics and Probability, Computer Architecture**

***Co-Requisite:* Operating Systems**

Syllabus:

- Structure of networks, network architectures, reference model of ISO, networks of ARPA, SNA, DECNET and general. Network topology, connectivity analysis, delay analysis, design of network with local access.
- Design of physical layer, fundamentals of theory for data transfer, transfer telephone systems and multiplexing, survey on terminal, transfer errors.
- Data relation layer, primary protocols for data relation, internet protocol standards, internet IP protocol, IPv6 protocol, internet ISO protocol, ISO routing protocol, sliding window protocol, protocols analysis.
- Primary layer of network, point-point networks, routing algorithms, density, Transport & Application layers.
- Secondary layer, satellite and radio networks, broadcasting satellite packages, radio packages.

References:

1- F. Halsail, Data Communications, Computer Networks and Open Systems, 4th edition, Addison Wesley, 1996.
2- A. S. Tannenbaum, Computer Networks, 3rd edition, Prentice-Hall, 1996.
3- W. Stallings, Data and Computer Communications, Prentice-Hall, 1996.

***Course:* Computer Networks Lab**

***Credits:* 1 - *ECTS:* 3**

***Type of Course:* Practical**

***Prerequisite:* Computer Networks**

Syllabus:

Investigation and simulation of some basic Computer Networks concepts using CISCO packet tracer and presenting practical applications of Computer Networks course.

Course: **Operating Systems**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Computer Architecture**
Syllabus:

Definition of operating system and its fundamental duties as a virtual machine and manager of resources, types of resources, brief history of operating systems and its classifications, operating system in view of user, concept of processing, function, tasks, types of processor, concept of interrupts, different types of interrupts and their processing, interrupts priority and next coming, programming and control of I/O, concurrency in I/O, memory management, introducing multiprogramming environments, static memory allocation, dynamic memory allocation, commutative memory allocation, paging according to demand, partitioning, paging part, memory hierarchy, Processor management, tasks scheduling and its policies, processes scheduling and its policies, multiprocessor systems, weak communication, stable communication, allocation of resources to processes, competition mode, blocked mode and methods of releasing, mutual exclusion, concurrency of processes by use of semaphore, files management, assessment of deadlock problem, security and protection of operating systems, introduction to operating system in network and distributed operating system.

References:

1- A. S. Tanenbaum, Modern Operating Systems, Prentice-Hall, 1992.
2- W. Stallings, Operating Systems, 3rd edition, Prentice-Hall, 1998.
3- H. M. Deitel, Operating Systems, Addison Wesley, 1993.
4- A. Silberschatz and J. L. Peterson, Operating Systems Concepts, Addison Wesley, 1998.

Course: **Operating Systems Lab**
***Credits:* 1 - *ECTS:* 3**
***Type of Course:* Practical**
***Prerequisite:* None**
***Co-Requisite:* Operating Systems**
Syllabus:
Working with Linux Operating system (Ubuntu distribution) and investigation of its commands. In addition, Investigating some practical Operating Systems applications and concepts such as processes and threads.

Course: **Signals and Systems**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Differential Equations**
Syllabus:
This covers the fundamentals of signal and system analysis, focusing on representations of discrete-time and continuous-time signals (singularity functions, complex exponentials and geometrics, Fourier representations, Laplace and Z transforms, sampling) and representations of linear, time-invariant systems (difference and differential equations, block diagrams, system functions, poles and zeros, convolution, impulse and step responses, frequency responses). Applications are drawn broadly from engineering and physics, including feedback and control, communications, and signal processing.
References:
Oppenheim, Alan, and Alan Willsky. Signals and Systems. 2nd ed. Prentice Hall, 1996. ISBN: 9780138147570.

Course: **Fundamentals and Applications of Artificial Intelligence**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Algorithms Design**
Syllabus:

- What is artificial intelligence? Fundamentals, history and review of knowledge borders in this area.
- Intelligent agents: structures and operations of intelligent agents, and their environments.
- Problem solving: problem solving via search, problem formulation and approaches with some example for search, search types.
- Informed searching methods: best-first searching, heuristic functions, limited space search and another optimal methods.
- Knowledge based agents: agents with logical inference, representation of logic, sentence based logic, inference.
- First order logic: inference in this logic, deduction rules, forward-backward chaining deduction.
- Planning: problem solving with planning, a simple representation for planning and knowledge engineering for planning.

- Uncertainty: mechanism of performance under uncertainty conditions and obtaining probabilities.
- Representation of some application in expert systems, natural language processing, machine and robotics vision.

References:

1- Russel and Norwig, “Artificial Intelligence: A Modern Approach”, Prentice-Hall, 1995.
2- E. Rich. “Artificial Intelligence”, McGraw-Hill, 2nd edition, 1992.
3- I. Bratko, “Prolog Programming for AI”, Addison Wesley, 1986.
4- N. J. Nilsson, Principles of Artificial Intelligence, Springer-Verlag, 1980.
5- L. Sterling and E. Shapiro, Art of prolog, MIT Press, 1986.
6- I. Bratko, Prolog Programming for AI, Addison-Wesley, 1986.

Course: **Microprocessors and Assembly Language**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Computer Architecture**
Co-Requisite: **Microprocessors Lab**
Syllabus:

This course introduces students to basic computer organization and architecture concepts. It covers: number systems, Boolean algebra, digital logic circuits and their design, simple machine architecture, genealogy of microprocessors, von Neumann architecture, the system bus model, data representation and manipulation, organization of instruction sets and program execution, microprocessor organization, memory organization, organization of input and output subsystem, I/O interface; instruction set design philosophies, parallel processing, symmetric multiprocessing and clustering; case study of at least two microprocessor families and other components of computing system..

References:

1- Y. Bai, Practical Microcontroller Engineering with ARM® Technology, 1st Edition, Wiley-IEEE Press, 2015.
2- J. W. Valvano, Embedded Systems: Introduction to Arm Cortex-M Microcontrollers, 5th Edition, 2017.
3- M. A. Mazidi, S. Chen, and E. Ghaemi, STM32 Arm Programming for Embedded Systems, 1st Edition, 2018.

Course: **Microprocessors Lab**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **None**
Co-Requisite: **Microprocessors and Assembly Language**
Syllabus:

According to the syllabus presented in Microprocessors and Assembly Language.

Course: **Principles of Compiler Design**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Theory of Languages and Automata**
Syllabus:

Getting familiar with the basic components of compiler, their duty and relations. Getting familiar with the how the compiler works in detail.

References:

Compilers, principles, techniques and tools, by Aho, Sethi and Ullman, 2nd edition, 2007

Course: **Fundamentals of Computational Intelligence (Introduction to Machine Learning)**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Artificial Intelligence Fundamentals and Applications**
Syllabus:

Teaching and explanation of various Supervised and Unsupervised Machine Learning Algorithms (including K-Means, DBSCAN, Mean Shift, etc. for Unsupervised Learning and Clustering and including Decision Trees, Support Vector Machines (SVM) and Neural Networks for Supervised Learning). Deep Learning algorithms such as Perceptron, MLP and CNNs are of great importance in this course. Furthermore, Ensemble Learning methods such as ADABOOST and Random Forest were explained and Natural Language Processing (NLP) concepts, tools and methods such as Word2Vec and Bag of Words were also instructed.

References:

1- An Introduction to Statistical Learning with Applications in R, G. James, D. Witten, T. Hastie, R. Tibshirani.

2- Machine Learning: An Algorithmic Perspective, S. Marsland.
3- Machine Learning: A Probabilistic Perspective, K. P. Murphy.

Course: **Fundamentals of Information Security**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Computer Networks**
Syllabus:

Getting preliminary knowledge about the principles of security, authentication, access control, fundamentals of cryptography, software security, database security, cloud security, IDS, risk management, voting system security.

References:

Computer Security, principles and Practice, Stallings and Brown, 3rd edition, 2014

Course: **Internship**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **Research Methods and Presentation**
Syllabus:

Working as an intern in a company relevant to the Major of the student.

Course: **Software Engineering I**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Advanced Programming**
Co-Requisite: **Software Engineering Lab**
Syllabus:

- Software crisis, reasons of needing to methodology and development process, system's life cycle (including requirements analysis, general design, partial design, implementation and maintenance of system).
- Concepts of systems analysis, structural information systems (introducing some of the structural methods like Demarco, Yourdon, Gane & Sarson). Physical and logical models of available data-current systems, physical and logical model of data-current suggested systems, exact specifications of activities and data, feasibility study of system with regards to technology and financial resources and time, overall system design including databases, input forms, final reports, and software structure design, determining process specifications with system operations.
- An introduction to methods of gathering information, expense and time estimation needed for each stage, project management methods and tools, system analysis tools, system design tools, introduction to first part of CASE. In this course students must do a group type project.

References:

1- Bentley, Barlow and Toppan, Systems Analysis and Design Methods, 1990.
2- Yourdon, Modern Structured Analysis, Prentice-Hall, 1989.
3- J. Fitsgerald and A. Fitzgerald, Fundamentals of Systems Analysis, 3rd edition, John Wiley, 1987.
4- E. M. Awad, Systems Analysis and Design, 2nd edition, 1985.
5- Hawryszkiewgez, Introduction to Systems Analysis and Design, 2nd edition, Prentice-Hall, 1990.
6- K. E. Kendall and J. E. Kendall, Systems Analysis and Design, 2nd edition, Prentice-Hall, 1992.
7- B. Boehin, Software Engineering Economies, Prentice-Hall, 1981.
8- A. Summerville, Software Engineering, 4th edition, Addison-Wesley, 1996.

R. S. Pressman, Software Engineering, A Practitioner Approach, 4th edition, McGraw Hill, 1996.

Course: **Software Engineering Lab**
Credits: **1 -** ***ECTS:*** **3**
Type of Course: **Practical**
Prerequisite: **None**
Co-Requisite: **Software Engineering I**
Syllabus:

Teaching basic concepts of drawing UML diagrams and practical applications according to the syllabus presented in Software Engineering I.

Course: **Project**

***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Research Methods and Presentation**
Syllabus:

Working on an academic project under supervision of a professor chosen based on student’s choice and professor’s availability; Providing a comprehensive report for the intended project and presenting it.

***Course:* Fundamentals of Computer Vision**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Artificial Intelligence Fundamentals and Applications**
Syllabus:

Familiarizing students with the basic concepts of image, how to process images and explaining algorithms for finding similarities between images. The course covers these chapters: Image processing in Spatial domain, Image Processing in Frequency domain, Wavelet transform, Color, image matching, finding features in images using SIFT and SURF algorithms and explanation of how these algorithms are taken into account to find similarities between images and match them.

References:

R.C. Gonzalez; R.E. Wood, "Digital Image Processing, 4th Edition, 2018

***Course:* Database**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* Data Structure**
Syllabus:

- Introduction and review of discussions on Storage and Retrieval.
- Concepts and definitions of database management (definition of data and information, database definition and necessity, data independence, various models of database systems).
- Architecture of a database system (three-level architecture, external level, internal level, conceptual level, database and data administrator, data relationship administrator).
- Different models of database systems (Hierarchical model, Relational model, network model).
- Relational model of data base (relations, tables, base tables and non-base tables, query language).
- Relational model elements (domain, relation and its types).
- Integrity of relational model (candidate key, primary key, foreign key and its rules, Null foreign key and primary key).
- Relational algebra.
- SQL language.
- Functional dependencies (partial dependencies definitions, a collection of dependencies, irreducible collection of dependencies).
- BCNF, 3NF, 2NF, 1NF normalization, multi-value dependency (MVD), 4NF, join dependency (JD), SNF.
- Review of more advanced topics (protection, reparation, coherency, object-oriented data bases and inference data bases).

References:

1- C. J. Date, Introduction to Database Systems, Sixth Edition, Addison-Wesley, 1995.
2- Elmasri, Fundamental of Database Systems, 2nd edition, Addison-Wesley, 1994.
3- S. Abitebout, R. Hull and V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
4- H. Korth and A. Silberchatz, Database System Concepts, 3rd edition, McGraw-Hill, 1997.

***Course:* Database Lab**
***Credits:* 1 - *ECTS:* 3**
***Type of Course:* Practical**
***Prerequisite:* None**
Syllabus:

Learning to manage SQL based databases and presentation of how to setup and work with various DBMS by students.

***Course:* Game Theory**
***Credits:* 3 - *ECTS:* 6.9**
***Type of Course:* Theoretical**
***Prerequisite:* None**

Syllabus:

The course, Game Theory, provides a mathematical framework which makes possible the analysis of cooperative and non-cooperative decision-making process. The main topic of this course is to develop an essential tool for analyzing strategic interactions between rational agents. By electrical and computer engineering point of view, the game theory plays an important role in different research area with designing algorithms for modeling of multi-agent systems, decision making in competitive environment and understanding rival behavior.

References:

1- An introduction to game theory, Martin J. Osborne, Oxford University Press (2004).
2- Introduction to game theory, Peter Morris, Springer Universitext, 1994.
3- Several recently published papers related to special topics.

Course: **Software Testing**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Software Engineering I**
Syllabus:

In addition to theoretical knowledge, including the basics of software testing, test methods and techniques, types of test coverage and test tools are also examined in this course.

References:

1- P. Ammann, J. Offutt, Introduction to software testing, Cambridge University Press, 2017.
2- G. J. Myers, The Art of Software Testing, 3rd Edition, John Wiley & Sons, 2012.

Course: **Fundamentals of Datamining**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Database**
Syllabus:

Familiarity with the process of knowledge discovery, data recognition and evaluation, data preprocessing and cleaning, frequent patterns and association rules, classification and clustering.

References:

Data Mining – Concepts and Techniques, J. Han, M. Kamber, J. Pei, 3rd edition, 2012

Course: **Fundamentals of Internet of Things**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Microprocessors and Assembly Language, Computer Networks**
Syllabus:

Teaching the latest technologies available in the field of Internet of Things and similar fields. Instruction and explanation of Domain Specific IoTs, IoT Characteristics and Protocols, how to work with Arduino and ESP32, IoT Sensors, IoT communication protocols including IEEE 802.15.4-Zigbee-Zwave, LPWAN, COAP, Wi-Fi, Bluetooth, NFC and MQTT.

Course: **Web Programming**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Computer Networks**
Syllabus:

Familiarity with back-end and front-end programming on the web and the basic principles of full stack developer.

References:

1- Programming the World Wide Web, R.W.Sebesta
2- Beginning HTML, XhTML, CSS and JavaScript, J. Duckett
3- www.w3schools.com

Course: **Object-Oriented System Design**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **Advanced Programming, Software Engineering I**
Syllabus:

In this lesson, we are looking for familiarity with the purpose, necessity, structure and application of design patterns in software engineering, especially GoF design patterns and microservice and cloud space patterns. We will learn how to use patterns in different situations and problems. For this purpose, various details related to the implementation of patterns are also discussed in detail and in depth. In this course, in addition to paying attention to the practical applications of models, the research topics of this field will also be introduced.

References:

1- E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Pattern: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1st ed., 1994.
2- E. Freeman, E. Freeman, B. Bates, and K. Sierra, Head first design patterns. O’Reilly & Associates, Inc., 1st ed., 2004.
3- A. Shalloway, J. R. Trott, Design Patterns Explained, A new Prospective on object-oriented Design, Addison Wesley, 2nd ed., 2004.
4- G. Booch, J. Rumbaugh, and I. Jacobson, Unified Modeling Language User Guide, Addison-Wesley Professional, 2nd ed., 2005.
5- M. Fowler, Analysis Patterns: Reusable Object Models, Addison-Wesley Professional, 1st ed., 1996.
6- A. Rasoolzadegan, Software Modeling Using UML based on RUP, 1st ed. (in Persian): Olom Rayaneh, 1st ed., 2014.

Course: **Marketing**
Credits: **3 -** ***ECTS:*** **6.9**
Type of Course: **Theoretical**
Prerequisite: **None**
Syllabus:

Getting to know the necessity and learning the concepts and skills needed to enter the business space: the dos and don'ts of entering the entrepreneurial space - the principles and steps of designing and setting up a business based on the lean startup model - generating ideas and evaluating ideas - How to implement ideas and develop a business model - Converting an idea into a product and commercializing it.

References:

1- Pure Startup, Eric Ries
2- Entrepreneur's Guide, Steve Blank
3- Business Model Creation, Alexander Osterwalder
4- Mom Test, Rob Fitzpatrick
5- Evaluation of Ideas, Alexander Osterwalder