Computer Engineering Department

Amirkabir University of Technology
(Tehran Polytechnic)

Curriculum

Bachelor of Computer Engineering

Publication Date:

June 7, 2020

Abstract:

This curriculum is developed for a bachelor of computer engineering in the computer engineering department of Amirkabir university of technology. This curriculum is in accordance with the guidelines and regulations of the ministry of science research and technology, moreover it respects the IEEE/ACM Computer Engineering curricula and similar programs in top universities around the world. This program identifies the required capabilities of computer engineering students, and design the curriculum to meet the requirements.

Keywords:

Curriculum, Bachelor, Computer Engineering

Table of Content

Title **Page**

Title **Page**

Chapter 1:

Introduction

1-1 Objective

The purpose of the undergraduate program in computer engineering is to train students who have the ability and skills to solve computer engineering problems, including analyzing, designing, and building computer systems. This program prepares students for entrepreneurship, industry work, and graduate studies.

1-2 Graduates' Abilities

Students are expected to have the following abilities and capabilities after successful completion of this curriculum:

1) Analysis, design, and implementation of computer systems
2) Understand, formulate, and solve problems
3) Applying math, science, and engineering knowledge to solve real-world problems
4) Applying techniques, skills and world-class tools to solve engineering problems
5) Ability to learn and apply new knowledge after graduation
6) Effective and responsible performance as a member of a team
7) Communicate effectively spoken and written
8) Adherence to the principles of professional ethics and social responsibility

1-3 Program Structure

The undergraduate program in Computer Engineering consists of 140 practical and theoretical units in five categories: General (20 units), Basic (20 units), Main (55 units), Specialized (30 units), and Optional (15 units). The overall structure of the program is depicted in Figure 1.

Passing all the general, basic, and main courses are required for all computer engineering students. The specialized courses are offered as a set of packages. Each package consists of eight three-unit lessons (24 units in total). Each student selects two packages of the packages and receives 18 units of one of the packages and 12 units of the other one. Students must obtain 15 optional units from the list of courses specified in the optional lesson table.

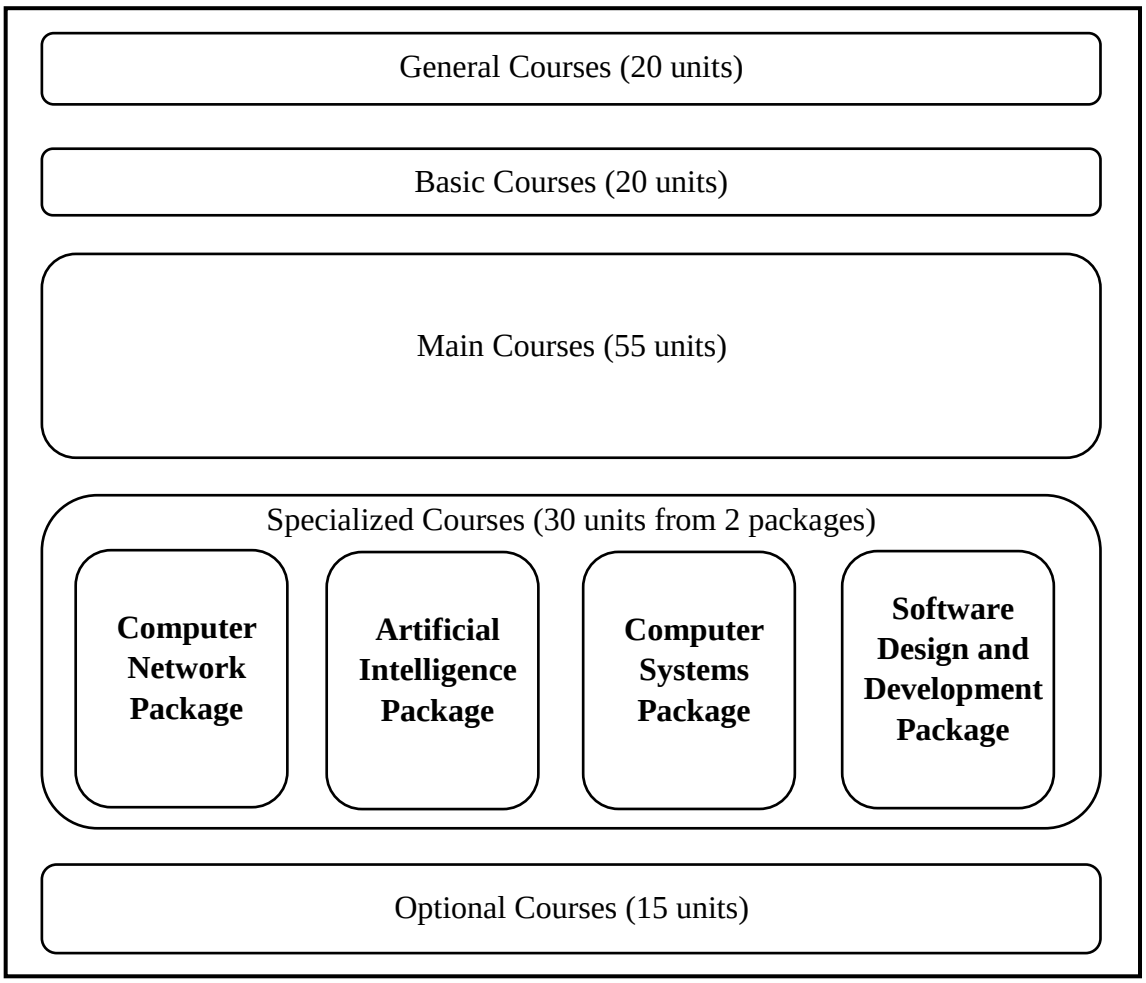

Figure 1 - Overall composition of the 140 units of computer engineering undergraduate courses

These packages are briefly described in the following.

Software Design and Development Package: The purpose of this package is to familiarize students with all the steps needed to design and develop a high-quality software system. Students in this pack will gain all the necessary knowledge about algorithm design techniques, database design principles, software project management, and software testing and design techniques. Understanding a few programming languages and compiler design principles are the other goals of this specialized package. Students who successfully complete courses in this specialized package can enroll in software development firms to pursue a job market or pursue a master's degree in related fields.

Computer Systems Package: Computer Systems package is related to the subjects of computer engineering that include design, manufacturing, implementation, and maintenance of software and hardware components of modern computing systems as well as computer-controlled devices. Students who take courses from this specialized package

can be involved in the computer systems industry and research groups from design to implementation of the systems. Examples of computer systems can be found in components of digital products such as digital cameras, tablets, automotive electronic control systems, digital medical devices, communication devices, and smart home appliances. In some of the courses of this package, the student will find the skills needed to design and build a complete digital system consisting of hardware and software. In addition, in some courses, students become familiar with the concepts of fast data processing in data centers, such as cloud computing.

Artificial Intelligence Package: The purpose of this package is to get students familiar with the concepts and basics of artificial intelligence, designing intelligent algorithms and intelligent data processing, moreover, it aims to teach problem-solving tools and methods such as logic, evolutionary algorithms, neural networks, and fuzzy systems. Students also become familiar with the applications of artificial intelligence in robotics, data mining, data recovery, and bioinformatics. The emphasis of the courses is based on the practical applications of artificial intelligence techniques. Interested students can choose some of their optional courses from the master of artificial intelligence courses, depending on their interests and abilities.

Computer Network Package: The purpose of this pack is to familiarize students with the concepts and fundamentals of computer networks and distribution systems. The courses of this package emphasize the theoretical and practical capabilities needed to prepare students for entrance into the labor market and entrepreneurship related to computer networks and network-based systems. Areas of interest in this package include the principles of wireless network design, web application development, applications for smart devices such as smartphones, cloud computing services, and computer network security. Students will be prepared to pursue higher education in the related fields and with choosing from master courses in computer networks and cloud computing rather than the package courses, as optional courses, they can deepen their knowledge and skills.

Chapter 2:

Courses List

2-1 General Courses

These courses are specified by the education committee of the university rather than this curriculum.

Each student should take 20 units of the courses in this category.

2-2 Basic Courses

Basic Courses				
No.	**Title**	**Units**	**Type**	**Prerequisite**
1	Mathematics I	3	Theory	---
2	Mathematics II	3	Theory	Mathematics I
3	Physics I	3	Theory	---
4	Physics II	3	Theory	Mathematics I
5	Differential Equations	3	Theory	Mathematics I
6	Engineering Probability and Statistics	3	Theory	Mathematics II
7	Physics Laboratory II	1	Practical	Physics II
8	General/Electrical Workshop	1	Practical	---
Total Units		**20**		

2-3 Main Courses

Main Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
1	CE101	Fundamentals of Computer and Programming	3	Theory	---	Fundamentals of Computer and Programming Workshop
2	CE102	Fundamentals of Computer and Programming Workshop	1	Practical	---	Fundamentals of Computer and Programming
3	CE103	Discrete Mathematics	3	Theory	Mathematics I, Fundamentals of Computer and Programming	---
4	CE104	Advanced Programming	3	Theory	Fundamentals of Computer and Programming	Advanced Programming Workshop
5	CE105	Advanced Programming Workshop	1	Practical	---	Advanced Programming
6	CE201	Logic Circuits	3	Theory	---	Discrete Mathematics, Logic Circuits Laboratory
7	CE202	Logic Circuits Laboratory	1	Practical	---	Logic Circuits
8	CE203	Data Structures and Algorithms	3	Theory	Discrete Mathematics, Advanced Programming	---
9	CE204	Electrical and Electronic Circuits	3	Theory	Physics II	Differential Equations, Electrical and Electronic Circuits Laboratory
10	CE205	Electrical and Electronic Circuits Laboratory	1	Practical	---	Electrical and Electronic Circuits

Main Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
11	CE206	Technical English	2	Theory	English Language II	---
12	CE207	Computer Architecture	3	Theory	Logic Circuits	Computer Architecture Laboratory
13	CE208	Computer Architecture Laboratory	1	Practical	---	Computer Architecture
14	CE209	Theory of Languages and Machines	3	Theory	Data Structures and Algorithms	---
15	CE210	Applied Linear Algebra	3	Theory	Mathematics II	---
16	CE301	Microprocessor and Assembly Language	3	Theory	Computer Architecture	Microprocessor and Assembly Language Laboratory
17	CE302	Microprocessor and Assembly Language Laboratory	1	Practical	---	Microprocessor and Assembly Language
18	CE303	Operating Systems	3	Theory	Computer Architecture	Operating Systems Laboratory
19	CE304	Operating Systems Laboratory	1	Practical	---	Operating Systems
20	CE305	Computer Networks	3	Theory	Engineering Probability and Statistics, Computer Architecture	Operating Systems, Computer Networks Laboratory
21	CE306	Computer Networks Laboratory	1	Practical	---	Computer Networks
22	CE307	Software Engineering I	3	Theory	Advanced Programming	---
23	CE308	Research and Presentation Methods	2	Theory	Technical English	---
24	CE309	Internship	1	Practical	Research and Presentation Methods	---

Main Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
25	CE401	Final Project	3	Practical	Research and Presentation Methods	---
Total Units			**55**			

2-4 Packages Courses

2-4-1 Software Design and Development Package

Software Design and Development Package Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
1	CE221	Algorithm Design	3	Theory	Data Structures and Algorithms	---
2	CE231	Principles of Database Design	3	Theory	Data Structures and Algorithms	---
3	CE232	Compiler Design	3	Theory	Data Structures and Algorithms	---
4	CE233	Programming Languages	3	Theory	Theory of Languages and Machines	---
5	CE331	Software Engineering II	3	Theory	Software Engineering I	---
6	CE431	Software Testing	3	Theory	Software Engineering II	---
7	CE432	User Interface Design	3	Theory	Software Engineering I	---
8	CE421	Information Retrieval	3	Theory	Data Structures and Algorithms, Engineering Probability and Statistics	---
Total Units			**24**			

2-4-2 Computer Systems Package

Computer Systems Package Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
1	CE222	Signal and Systems	3	Theory	Differential Equations	---
2	CE241	Interface Circuits Design	3	Theory	Microprocessor and Assembly Language	---
3	CE242	Programmable Digital Systems Design	3	Theory	Computer Architecture	---
4	CE243	Digital Electronics	3	Theory	Electrical and Electronic Circuits	---
5	CE341	Embedded and Real-Time Systems	3	Theory	Operating Systems, Microprocessor and Assembly Language	---
6	CE342	Multi-Processor Programming	3	Theory	Operating Systems	---
7	CE422	Fundamentals of Cloud Computing	3	Theory	Computer Networks, Operating Systems	---
8	CE441	Hardware-Software Codesign	3	Theory	Computer Architecture	---
Total Units			**24**			

2-4-3 Artificial Intelligence Package

Artificial Intelligence Package Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
1	CE221	Algorithm Design	3	Theory	Data Structures and Algorithms, Discrete Mathematics	---
2	CE222	Signal and Systems	3	Theory	Differential Equations	---
3	CE251	Principles and Applications of Artificial Intelligence	3	Theory	Data Structures and Algorithms	Applied Linear Algebra
4	CE351	Foundations of Computational Intelligence	3	Theory	Algorithm Design	---
5	CE352	Introduction to Robotics	3	Theory	Signal and Systems	---
6	CE451	Introduction to Bioinformatics	3	Theory	Foundations of Computational Intelligence	---
7	CE452	Data Mining	3	Theory	Foundations of Computational Intelligence	---
8	CE421	Information Retrieval	3	Theory	Data Structures and Algorithms, Engineering Probability and Statistics	---
Total Units			**24**			

2-4-4 Computer Network Package

Computer Network Package Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
1	CE222	Signal and Systems	3	Theory	Differential Equations	---
2	CE261	Web Programming	3	Theory	Computer Networks	---
3	CE361	Data Communication	3	Theory	Computer Networks, Signal and Systems	---
4	CE362	Information and Communications Security	3	Theory	Computer Networks	---
5	CE363	Multimedia Systems	3	Theory	Engineering Probability and Statistics, Signal and Systems	---
6	CE364	Mobile Device Programming	3	Theory	Advanced Programming	---
7	CE422	Fundamentals of Cloud Computing	3	Theory	Computer Networks, Operating Systems	---
8	CE461	Internet of Things Fundamentals	3	Theory	Microprocessor and Assembly Language, Computer Networks	---
Total Units			**24**			

2-4 Optional Courses

Optional Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
1	CE371	Computer Graphics	3	Theory	Advanced Programming	---
2	CE372	Human and Computer Interaction	3	Theory	Software Engineering I	---
3	CE373	Robot Building Workshop	3	Theory	Introduction to Robotics	---
4	CE374	Computer Games Design	3	Theory	Advanced Programming	---
5	CE375	Theory of Computation	3	Theory	Theory of Languages and Machines	---
6	CE376	Computer Simulation	3	Theory	Operating Systems, Engineering Probability and Statistics	---
7	CE377	Foundations of Computer Animation	3	Theory	Computer Graphics	---
8	CE378	Information Technology Project Management	3	Theory	---	---
9	CE379	E-Commerce	3	Theory	Engineering Economics, Computer Networks	---
10	CE380	Startup Business Development	3	Theory	Software Engineering I	---
11	CE381	Industrial Automation	3	Theory	Microprocessor and Assembly Language	---
12	CE382	Special Topics 1	3	Theory	---	---
13	CE383	Special Topics 2	3	Theory	---	---
14	---	Linear Control (Electrical Engineering)	3	Theory	---	---
15	---	Engineering Economics (Industrial Engineering)	3	Theory	---	---

Optional Courses						
No.	**Code**	**Title**	**Units**	**Type**	**Prerequisite**	**Corequisite**
16	---	Project Control (Industrial Engineering)	3	Theory	---	---
17	---	Graph Theory (Computer Science)	3	Theory	---	---
18	---	Numerical Computation	3	Theory	---	---
19	---	Up to 5 courses from the non-selected Packages	3	Theory	---	---
20	---	Up to 2 courses from other departments (need department approval)	3	Theory	---	---
21	---	Up to 2 master courses (need department approval)	---	---	----	---

Chapter 3:

Courses Syllabus

3-1 Main Courses

3-1-1 Fundamentals of Computer and Programming (CE101)

Fundamentals of Computer and Programming (CE101)		
Units	Prerequisites	Corequisites
3	---	Fundamentals of Computer and Programming Workshop

Objectives:

The main objective of this course is to learn how to solve problems with computers which is achieved by understanding how computer works as a computational machine, learning algorithmic thinking to solve problems, and learning how to describe an algorithm using a programming language (e.g., C)

Topics:

- Introduction to computer
  - History
  - Hardware components
  - Software components
- Problem solving by computer
  - Problem solving steps
  - Algorithm design
  - Algorithm description
- Programming basic steps
  - Programming languages types
  - Programming procedure
- Basic steps of the programming language
  - The main function
  - Variables
  - Types
  - Values
- Developing simple programs
  - Basic input/output
  - Mathematical calculations
- Loop and Condition statements
- Procedural programming and function
- Recursive functions

- Debugging
- Arrays
- Pointers
- Characters and Strings
- Complex data structures
- File
- Direct access to hardware (optional)

References:

[1] Paul Deitel, Harvey Deitel, *C How to Program*, 8th Edition, Pearson Education, 2015

[2] David Griffiths, Dawn Griffiths, *Head First C,* O'Reilly, 2012

3-1-2 Fundamentals of Computer and Programming Workshop (CE102)

Fundamentals of Computer and Programming (CE102)		
Units	Prerequisites	Corequisites
1	---	Fundamentals of Computer and Programming

Objectives:

This lab is corequisites of the "Fundamentals of Computer and Programming" course. Its objective to improve the programming skill of students. In this course, students develop programs based on concepts they are learning CE101.

Topics:

- Introduction to computer hardware components (I/O, CPU, Mainboard, Cards, …)
- Familiarize with different OSs (Linux, Windows, MacOS)
- Familiarize with applications (Browsers, Office, IDE, …)
- Developing basic programs in IDE
- Developing programs with mathematical calculations
- Developing programs to interact with user via formatted console I/O
- Developing programs need making decisions
- Developing programs need iterations
- Developing complex programs using functions
- Developing programs need arrays
- Developing programs need multiple functions and arrays
- Developing programs need dynamic memory (heap) using pointers
- Developing programs to work with struct
- Developing programs to work with files

References:

[1] Paul Deitel, Harvey Deitel, *C How to Program*, 8th Edition, Pearson Education, 2015

[2] David Griffiths, Dawn Griffiths, *Head First C*, O'Reilly, 2012

3-1-3 Discrete Mathematics (CE103)

Discrete Mathematics (CE103)		
Units	Prerequisites	Corequisites
3	Calculus I, Fundamentals of Computer and Programming	---

Objectives:

The course consists of the topics of discrete mathematics that are extensively used in other courses of bachelor of computer science and engineering. This course is meant to acquaint students with the logical foundations of reasoning and to make more effective the students' problem solving abilities. The course contains topics of set theory, logic, mathematical induction, counting, relations and functions, recurrence relations, number theory, and graph theory.

Topics:

- Counting
  - Principles of Counting
  - Inclusion and Exclusion
  - Generating Functions
- Logic
  - Propositional Logic
  - First-Order Logic
  - Boolean Algebra
- Induction
  - Mathematical Induction
  - Inductive Definitions
  - Structural Induction
- Integers
  - Divisibility
  - Diophantine Equations
  - Modular Arithmetic
- Relations and Functions
  - Partial Orders and Equivalence Relations
  - Countable and Infinite Sets
  - Time Complexity
- Recurrence Relations
  - Recursive Formulation of Problems
  - Linear Homogeneous and Nonhomogeneous Recurrence Relations

- Graph Theory
  - Euler Circuits
  - Hamilton Cycles
  - Graph Coloring
  - Tree

References:

[1] R. P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, 5th Edition, Pearson, 2017.

[2] K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th Edition, McGraw-Hill, 2011.

[3] S. S. Epp, *Discrete Mathematics with Applications*, 4th Edition, Brooks Cole, 2010

3-1-4 Advanced Programming (CE104)

Advanced Programming (CE104)		
Units	Prerequisites	Corequisites
3	Fundamentals of Computer and Programming	Advanced Programming Workshop

Objectives:

The purpose of the course is to study the fundamental concepts and techniques necessary to write high-quality programs. In order to deal with the complexities of medium to large scale programs, topics such as top-down design and basic concepts of object-oriented programming are covered. Additionally, the course focuses on program verification, testing, and debugging. An object-oriented programming language (e.g. Java) is utilized to convey the concepts and techniques of OOP stepwise design to students. Other advanced topics such as multi-threading, GUI libraries, and network programming are also covered.

Topics:

- Introduction to large scale programming and high quality program attributes
- Top-down design, modeling the real-world using object-oriented approach
- UML Modeling
- Object-Oriented Programming
  - Classes and Objects
  - Inheritance
  - Polymorphism
  - I/O Library Classes and other Library Classes
- Testing and debugging
  - Testing and Fault detection
  - Exception Handling
- Standard Data Structures
  - Collections
  - Queue and Stack
  - Text processing and Strings
- Advanced Concepts and Techniques
  - GUI Programming
  - Graphics Programming
  - Multi-Threaded Programming
  - Network Programming
  - Generic Programming
  - Reflection and its Uses
  - Connecting to Standard Databases

References:

[1] D. Barns, M. Kolling, *Object First with Java*, 5th Edition, Pearson, 2012.

[2] H. Deitel, P. Deitel, *Java How to Program*, 9th Edition, Prentice-Hall, 2012.

3-1-5 Advanced Programming Workshop (CE105)

Advanced Programming Workshop (CE105)		
Units	Prerequisites	Corequisites
1	---	Advanced Programming

Objectives:

Advanced Programming Lab is offered along with the Advanced Programming Course. It aims to provide the opportunity to implement concepts learned in the advanced programming course. In this lab, in addition to conducting small weekly projects, students carry out a major programming project throughout the semester

Topics:

- Understanding UML and building models using UML
- Creating and working with classes and objects
- Implementation of inheritance and polymorphic behavior among classes
- Implementation of common text processing concepts
- Creation of GUI with various components
- Implementation of a multi-threaded program
- Implementation of a program distributed over a network
- Implementing Generic classes and methods
- Implementation and application of reflection
- Connecting to a database
- A large-scale programming project

References:

[1] D. Barns, M. Kolling, *Object First with Java*, 5th Edition, Pearson, 2012.

[2] H. Deitel, P. Deitel, *Java How to Program*, 9th Edition, Prentice-Hall, 2012.

3-1-6 Logic Circuits (CE201)

Logic Circuits (CE201)		
Units	Prerequisites	Corequisites
3	---	Discrete Mathematics, Logic Circuits Laboratory

Objectives:

This course has two main objectives:

- Introduction to basic concepts and components of logic circuits and their operation.
- To develop skills in designing digital circuits and systems and analyzing them in terms of their functionalities and speed

Topics:

- Introduction to logic circuits and basic concepts
- Numerical systems
- Boolean algebra
- Logic gates
- Analysis and design of combinational circuits
- Simplification techniques for combinational circuits
  - Boolean algebra
  - Karnaugh map
- Basic block circuits and their applications
  - Multiplexer
  - Demultiplexer
  - Encoder
  - Decoder
  - 7-segment
- Arithmetic circuits
  - Adder
  - Subtractor
  - Comparator
  - Multiplier
- Electronic circuits of logic gates
- Basic electronics concepts of logic circuits
  - High-impedance
  - Pull-up/pull-down resistors
  - Wired logic
  - Delay

- Basic elements of sequential circuits
  - Latch
  - Flip-Flop
- Analysis and design of sequential circuits
- Sequential circuits optimization techniques

References:

[1] C. Roth, L. Kinney, *Fundamentals of Logic Design*, 7th Edition, Cengage Learning, 2014.

[2] F. Vahid, *Digital Design with RTL Design, VHDL, and Verilog*, 2nd Edition, Wiley, 2011.

[3] M. Mano, *Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog*, 6th Edition, 2017.

[4] R. H. Katz, G. Borriello, *Contemporary Logic Design*, 2nd Edition, Benjamin Cummings /Addison Wesley, 2004.

3-1-7 Logic Circuits Laboratory (CE202)

Logic Circuits lab rotary (CE202)		
Units	Prerequisites	Corequisites
1	---	Logic Circuits

Objectives:

This laboratory has two main objectives:

- To gain knowledge about commercial logic circuits components and their operations
- To develop skills in designing digital circuits and systems

Topics:

- Introduction to laboratory equipment
- Basic logic gates
- Implementation of logic functions by Karnaugh map
- Hardware description language
- Multiplexers, de-multiplexers and decoders
- Implementation of an adder/subtractor
- Code converter
- Arithmetic/logic unit
- Latches and flip-flops
- Timers
- A project on sequential circuit design

References:

[1] C. Roth, L. Kinney, *Fundamentals of Logic Design*, 7th Edition, Cengage Learning, 2014.

[2] F. Vahid, *Digital Design with RTL Design, VHDL, and Verilog*, 2nd Edition, Wiley, 2011.

[3] M. Mano, *Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog*, 6th Edition, 2017.

[4] R. H. Katz, G. Borriello, *Contemporary Logic Design*, 2nd Edition, Benjamin Cummings /Addison Wesley, 2004.

3-1-8 Data Structures and Algorithms (CE203)

Data Structures and Algorithms (CE203)		
Units	Prerequisites	Corequisites
3	Discrete Mathematics, Advanced Programming	---

Objectives:

The goal of this course is introduction to various types of data structures for management of data in main memory and secondary storage of computers, and also introduction to methods of designing of algorithms.

Topics:

- Analysis of Algorithms
  - Growth of functions
  - Recursive relations and methods of solving them (guess and induction, substitution, master theorem)
  - Amortized analysis
- Various Types of Lists
  - One-way Lists
  - Two-ways Lists
  - Queues, Stacks
  - Lists Operations
  - Applications of Lists (mathematical expressions, garbage collection, merge sort)
- Trees
  - Initial definitions
  - Expression tree
  - Various tree representations
  - Induction on trees
  - Exploration of trees
  - Structural induction
  - Binary trees
  - Operations on expression trees
  - Converting various types of expressions
  - Tries
  - Binary search trees
- Hashing
  - Cascading
  - Global hashing

- Open hashing
- Sorting and Order Statistics
  - Lower bound
  - Decision tree
  - Linear-time sorting (counting, radix, and bucket sorts)
  - Quick sort
  - Heap sort
  - Statistical sort
  - External sort
- Main methods of Algorithm Design
  - Divide and conquer
  - Dynamic programming
  - Greedy approach
  - Backtracking method
  - Branch and Bound

References:

[1] T. Cormen, C. Leiserson, R. Rivest, *Introduction to Algorithm*, McGraw-Hill, 2001

3-1-9 Electrical and Electronic Circuits (CE204)

Electrical and Electronic Circuits (CE204)		
Units	Prerequisites	Corequisites
3	Physics II	Differential Equations, Electrical and Electronic Circuits Laboratory

Objectives:

The goal of this course is to introduce students to basic concepts and principles of DC and AC circuits theory and analysis. It provides skills to develop and solve mathematical representations of circuits, as well as to analyze the dynamic behavior of electrical and electronic circuits.

Topics:

- Basic concepts
  - Lumped and distributed circuits
  - Kirchhoff's voltage and current laws
- One-port circuit elements
  - Resistor
  - Capacitor
  - Inductor
  - Diode
  - Independent voltage and current source
- Two-port elements
  - Dependent source and its application in modeling transistors and operational amplifiers
- Energy and power
  - Active elements
  - Passive elements
- Analysis of simple resistive circuits
  - Series and parallel connections
  - Voltage and current division
  - Source transformation
  - Basic nodal and mesh analysis
  - Thevenin and Norton equivalent circuits
  - Superposition principle
  - Maximum power transfer
- 1st order circuits
  - linear and time-independent elements

- Source-free and complete response
- Natural and forced response
- Step response and impulse response and their relation

- 2nd order circuits
  - Source-free and complete response
  - Natural and forced response
  - Step and impulse response
  - Dual circuits
  - Use of convolution to determine circuit response
- Sinusoidal steady-state analysis
  - Impedance and admittance
  - Phasor
  - Frequency response
- Non-linear resistive circuits
  - Introduction of diode
  - Mathematical and graphical analysis of resistive-diode circuits
  - Rectifier circuits
- Amplifiers
  - Introduction of BJT and MOSFET as amplifiers
  - Circuit modeling using dependent sources
  - Input/output impedance
  - Thevenin and Norton equivalent circuits
  - DC analysis and operating point
- Operational amplifiers
  - Circuit modeling using dependent sources
  - Analysis of simple OPAMP circuits
  - Practical examples (amplifier, current source, filter, integrator and differentiator, voltage follower, etc.)

References:

[1] W. H. Hayt, J. E. Kemmerly, and S. M. Durbin, *Engineering Circuit Analysis*, 8th Edition, McGraw Hill, 2011.

[2] C. K. Alexander, M. N. O. Sadiku, *Fundamentals of Electrical Circuits*, 5th Edition, McGraw Hill, 2012.

[3] A. Agarwal, J. H. Lang, *Foundations of Analog and Digital Electronic Circuits*, Morgan Kaufmann Publishers, 2005.

3-1-10 Electrical and Electronic Circuits Laboratory (CE205)

Electrical and Electronic Circuits Laboratory (CE205)		
Units	Prerequisites	Corequisites
1	---	Electrical and Electronic Circuits

Objectives:

The goal of this lab is to let students get familiar with electrical circuit elements and lab equipment (e.g. oscilloscope, multimeter, power supply, signal generator, …) and learn how to work with them. They will also learn how to simulate, implement, test and debug practical electrical and electronic circuits.

Topics:

- Getting familiar with circuit elements and lab equipment
- Examining the Ohm's law and Kirchhoff's current and voltage laws
- Examining the Norton and Thevenin equivalent circuits
- Getting familiar with Orcad Capture simulation tool
  - Running DC Bias analysis
  - DC Sweep analysis
  - AC analysis
  - Time Domain analysis
- RL and RC transient response analysis
  - Implementing RL and RC low-pass
  - Band-pass and high-pass filters
  - Examining unit-step response
- RLC transient response analysis
  - Implementing RLC circuits
  - Examining unit-step response
- RC frequency response analysis
  - Implementing and analyzing RC low-pass and band-pass filters and RC integrator circuit
- Getting familiar with diodes and its applications (rectifier, clamper, clipper)
- Getting familiar with MOSFET transistor as an amplifier and as a switch, learn how to bias a transistor
- Getting familiar with Opamps, its application as an inverting and non-inverting amplifiers, active filters and integrator
- Final project

References:

[1] W. H. Hayt, J. E. Kemmerly, and S. M. Durbin, *Engineering Circuit Analysis*, 8th Edition, McGraw Hill, 2011.

[2] C. K. Alexander, M. N. O. Sadiku, *Fundamentals of Electrical Circuits*, 5th Edition, McGraw Hill, 2012.

[3] A. Agarwal, J. H. Lang, *Foundations of Analog and Digital Electronic Circuits*, Morgan Kaufmann Publishers, 2005.

3-1-11 Technical English (CE206)

Technical English (CE206)		
Units	Prerequisites	Corequisites
2	English Language II	---

Objectives:

The purpose of this course is to improve student's abilities and skills in understanding technical articles and lectures in Computers Engineering. Variety of technical subjects are presented using motivating texts and talks to improve the students' English skills. Topics reflect the latest developments in technology in Computer and IT domain.

Topics:

- Operating Systems (Introduction to Unix)
- People in Computing (Familiarity with different fields of CE and IT, familiarity with writing CV)
- Software Engineering and Object Oriented Programming
- Recent Developments in CE/IT/AI
- The Future of CE/IT/AI
- Computing Support
- Data Security
- Multimedia
- Graphical User Interface
- Website Development
- Computer Architecture
- The Internet
- Applications Programs
- Web Design and Development
- Database Systems
- Cloud Computing
- Search Engine and Search Engine Optimization
- Social Networks
- Mobile Applications
- Computer Security
- Virtual Reality
- Computer Networks
- Artificial Intelligent

References:

[1] E. H. Glendinning , J. McEwan, *Oxford English for Information Technology*, Oxford University Press, 2006.

3-1-12 Computer Architecture (CE207)

Computer Architecture (CE207)		
Units	Prerequisites	Corequisites
3	Logic Circuits	Computer Architecture Laboratory

Objectives:

Computer Architecture courses introduces the structure and organization of a basic computer, memory units, central processing unit, and input/output devices. The detailed architecture of a processor, Register-Transfer Language (RTL), Arithmetic and Logic Unit (ALU), various arithmetic components (adders, subtractors, multipliers, and dividers), Von-Neumann algorithm, data-path and control unit design, memory hierarchy, and performance evaluation are discussed in this course. One of the popular Hardware Description Languages (HDL) is used for design and simulation of the components.

Topics:

- Introduction and basic concepts
  - Definitions and history
  - Performance
  - Amdahl's Law
- Instructions and computer language
  - Number and instruction presentations
  - ISA
  - Addressing mode
  - RISC and CISC concepts
- Computer arithmetic
  - ALU
  - Adders/Subtractors
  - Multipliers
  - Dividers
  - Floating-point unit
- Processing unit
  - RTL
  - Von-Neumann model
  - Data-path
  - Control unit
  - Micro-operation
  - Pipeline

- I/O
- DMA

- Memory hierarchy
  - Memory wall
  - Static and dynamic memory
  - Cache memory
  - Cache placement and replacement algorithms
- Parallel processing
  - Technology trend
  - Flynn taxonomy
  - GPUs
  - Clusters
  - Multicore

References:

[1] D. Patterson, J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Morgan Kaufmann, 5th Edition, 2013.

[2] M. Mano, *Computer System Architecture*, Prentice Hall, 3rd Edition, 1992.

[3] M. Mano, C.R. Kimi, *Logic and Computer Design Fundamentals*, 3rd Edition, Prentice-Hall, 2004.

[4] W. S. Stalling, *Computer Organization and Architecture*, 6th Edition, Prentice-Hall, 2003.

[5] V. C. Hamacher, Z. G. Vrasenic, and S. G. Zaky, *Computer Organization*, McGraw-Hill, 3rd Edition, 1990.

3-1-13 Computer Architecture Laboratory (CE208)

Computer Architecture Laboratory (CE208)		
Units	Prerequisites	Corequisites
1	---	Computer Architecture

Objectives:

The aim of this laboratory is to acquaint students with sequential circuit design, analysis and debugging, ALU design and implementation, control unit design, basic computer design and implementation, bus issues and IO systems. Moreover, students will learn to describe, synthesis and practical implementation on FPGA chips.

Topics:

- VHDL basics
  - Basic circuit description
  - Simulation
- Basic circuits
  - Examples: decoder, encoder, multiplexer and 4-bit comparator
- Sequential circuits
  - Process
  - Counter implementation
- Adders
  - Implementation
  - Delay-area analysis
- Multipliers
- Delay in digital circuits
- Memories
  - Memory design,
  - ROM, RAM and CAM
- Dividers
- Basic computer design and implementation (model I)
- Basic computer design and implementation (model II)
- Control unit design
- Pipeline
  - Implementation
  - Speedup analysis
- Syn./Async. uni-/bi- direction bus implementation
- Bus arbitration implementation

- Floating-point adders/subtractors
- Performance evaluation
  - How to find IPC, CPI, MIPS and DMIPS of a typical processor
- Memory hierarchy

References:

[1] S. Brown, Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*, McGraw-Hill, 2003.

[2] B. Parhami, *Computer Arithmetic – Algorithms and Hardware Designs*, Oxford University Press, 2000.

[3] D.A. Patterson, J.L. Hennessey, *Computer Organization and Design: The Hardware, Software Interface*, Morgan Kaufmann, 5th Edition, 2013.

[4] D.M Harris, *Digital Design and Computer Architecture*, 2nd Edition, Morgan Kaufman, 2012.

3-1-14 Theory of Languages and Machines (CE209)

Theory of Languages and Machines (CE209)		
Units	Prerequisites	Corequisites
3	Data Structures and Algorithms	---

Objectives:

This course familiarizes students with the foundations and principles of computer science and discusses material that is useful in subsequent courses. This course also strengthens students' ability to carry out formal and rigorous mathematical arguments.

Topics:

- Fundamental concepts in Formal Languages and Automata Theory
- Theory of Regular Languages
  - Acceptor for Regular Languages
  - Regular Expressions
  - Closure Properties of Regular Languages
  - Decidable Problems for Regular Languages
- Theory of Context Free Languages
  - Simplification of Context Free Languages
  - Normal Forms for Context Free Languages
  - Acceptor for Context Free Languages
  - Closure Properties of Context Free Languages
  - Decidable Problems for Regular Languages
- Theory of Context Free and Type Zero Languages
  - Standard Turing Machine
  - Non-standard Models of Turing Machines
  - Acceptor for Context sensitive Languages
- Turing Thesis and Computability

References:

[1] P. Linz, *Introduction to Formal Languages and Automata*, 6th Edition, Jones & Bartlet Learning, 2017.

[2] J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd Edition, Addison-Wiley, 2006.

[3] T. A. Sudkamp, *Languages and Machines*, 3rd Edition, Pearson Education Inc., 2006.

3-1-15 Applied Linear Algebra (CE210)

Applied Linear Algebra (CE210)		
Units	Prerequisites	Corequisites
3	Mathematics II	---

Objectives:
The objective of this course is studying topics of linear algebra, its applications and optimization that are extensively used in different fields of computer engineering. This course covers some theoretical topics and also, in simulation exercises, the students become familiar with the relevant software and applications in computer engineering.

Topics:

- Linear Equations in Linear Algebra
  - Linear systems and their solutions
  - Matrices
  - Matrix equation
  - Linear independence
  - Linear transformations
- Matrix Algebra
  - Matrix operations
  - Inverse of matrix
  - Matrix factorization
  - Determinants
- Vector Spaces
  - Vector spaces and subspaces
  - Null space
  - Column space
  - Bases
  - Dimension of a vector space
  - Rank
  - Change of basis
- Eigenvalues and Eigenvectors
  - Eigenvalues and eigenvectors
  - Characteristic equation
  - Diagonalization
    Applications
- Orthogonality and Least Squares
  - Inner products

- Orthogonal sets
- The Gram-Schmidt process
- Least squares problems
- Applications

- Singular Value Decomposition
- Principal Component Analysis
- Optimization
  - Vector functions
  - First and second order derivative
  - Introduction to different types of optimization problems
  - Linear programming
  - The simplex algorithm

References:

[1] D. C. Lay, S. R. Lay, and J. J. McDonald, *Linear Algebra and its applications*, 5th Edition, Pearson, 2015.

[2] P. N. Klein, *Coding the Matrix: Linear Algebra through Applications to Computer Science*, Newtonian Press, 2013.

[3] B. Noble, J. W. Daniel, *Applied Linear Algebra*, 3rd Edition, Pearson, 1987.

3-1-16 Microprocessor and Assembly Language (CE301)

Microprocessor and Assembly Language (CE310)		
Units	Prerequisites	Corequisites
3	Computer Architecture	Microprocessor and Assembly Language Laboratory

Objectives:

On successful completion of this course students will be familiar with hardware and software aspects of microprocessor- and microcontroller-based systems (using a specific architecture such as AVR or ARM) including their internal architecture and modules, programming in assembly and high level languages, interfacing with memories and Input/outputs, handshaking between microprocessor and peripherals, practical examples and case studies of microcontroller-based digital embedded and IoT systems.

Topics:

- Introduction to microprocessors, their history, packaging methods, different kinds of microprocessors including general purpose microprocessors, digital signal processors, microcontrollers, and special purpose processors
- Understand basic microcontroller architecture modules
- Registers, control, data and address busses, instruction queue, pipelining, central processing unit, arithmetic logic unit, memory, stack, etc.
- Brief introduction of manufacturing companies, types of architectures, software development tools, programming and troubleshooting of microcontroller programs
- Microcontroller assembly language, addressing modes, instruction format, instruction sets, subroutine, macro, directives, expressions
- System clock and clock options
- System control and reset, Watchdog timer
- Input/Output ports in microcontrollers, Interrupts (external interrupts)
- Timers/counters (counting, waveform generation, PWM, etc)
- Analog Comparator (AC)
- Analog to digital convertor (ADC)
- Power management and sleep modes
- Serial port and synchronous and asynchronous communication, introduction of USART, familiarity with RS232 and RS485 protocols and daisy chain configuration
- Memory (SRAM, EEPROM, Flash, etc.), memory mapping and interfacing, address decoding, external memory interface timing
- Address decoding for Input/output ports, I/O interface timing
- Peripheral controlling (polling and interrupts), understand the operational principles of hardware support for processor interrupts

- Practical examples and case studies of microcontroller-based embedded and IoT systems

References:

[1] M. M. Homayounpour, F. Hessar, S. A. Asghari, A. Ghanbari, *Microprocessor and Assembly Language*, Sheikh Bahaie Publisher, 2015.

[2] M. A. Mazidi, S. Naimi, S. Naimi, *The AVR Microcontroller and Embedded Systems using Assembly and C*, Prentice Hall, 2010**.**

[3] M. A. Mazidi, D. Causey, R. D. McKinlay, *PIC Microcontroller and Embedded Systems using Assembly and C for PIC18*, Prentice Hall, 2008**.**

[4] R. H. Barnett, S. Cox, L. O'Cull, *Embedded C Programming and the Atmel AVR*, Delnmar Cengage Learning Publishing, 2011.

[5] B. van Dam, *Microcontroller System Engineering, 45 projects for PIC, AVR and ARM*, 2008.

[6] T. Wilmshurst, *Designing Embedded Systems with PIC Microcontrollers, Principles and Applications*, 2nd Edition, Newnes, 2010.

[7] *Atmel and ARM Microcontrollers' Datasheets*.

3-1-17 Microprocessor and Assembly Language Laboratory (CE302)

Microprocessor and Assembly Language Laboratory (CE302)		
Units	Prerequisites	Corequisites
1	---	Microprocessor and Assembly Language

Objectives:

By successfully completing this lab, students should be familiar with the issues outlined below in the design and implementation of microprocessor-based and microcontroller-based systems. In order to target the activities of this laboratory and to create interest and enthusiasm for the student, it is recommended that the final project be defined from the beginning and in the first sessions of the laboratory, and its characteristics are such that by the tests students perform during each session, he/she learns to do a part of the final laboratory project.

Topics:

- Familiarity with a simulation software (such as Proteus); Familiarity with a PCB design software (such as Altium software), and ultimately doing a simple electronic project.
- Familiarity with an assembler and a microcontroller compiler (such as AVR Studio and CodeVision for the AVR family).
- Providing or constructing a programmer, setting up reset circuit, scheduling fuse bits, producing clock signal for microcontroller
- Provide a reset interrupt service routine, set up stack pointer, work with ports, polling a pin from a port, generating software latencies, and working with watchdog timer
- Working with external interrupts and using power saving modes
- Working with ports, reading the value specified by a four or eight-bit Dip-Switch connected to a port, converting the read value to a BCD No., converting BCD digits to 7-seg codes and displaying the results by two or four 7-seg displays
- Matrix keyboard (keyboard design and display of digits read from a keyboard by 7-seg displays)
- Working with LCD (connecting a LCD to microcontroller and displaying information received from the keyboard)
- Writing and reading data in / from EEPROM microcontroller memory
- Work with timer/counter, microcontroller in normal operation mode and CTC (blinking with two LEDs connected to two pins of microcontroller that alternately turns on and off).
- Working simultaneously with two timer/counters (digital frequency meter)
- Working with timer/counter 2 in PWM mode (adjusting the intensity of a LED illumination or adjusting a small low voltage DC motor speed by PWM wave)
- Working with microcontroller analogue comparator (switching on and off a LED by increasing or decreasing the voltage of one of the analog comparator input relative to its other input)

- Working with an analogue to digital convertor (measuring the temperature or making a voltmeter or resistance meter and displaying the measured value on a LCD)
- Working with the microcontroller USART interface (communicating between two microcontrollers or communicating between a microcontroller and a computer via RS232 communication using the USART interface on the microcontroller side and a USART Serial Connection .NET Component terminal program on the computer side)
- Working with the SPI interface of the microcontroller and connecting two microcontrollers using SPI interface; writing and reading data on the SD RAM (Optional)
- Working with the TWI interface of the microcontroller and connecting two microcontrollers using the TWI interface, working with the real time clock (RTC) chip and displaying the actual time on the LCD (optional)

References:

[1] M. M. Homayounpour, F. Hessar, S. A. Asghari, A. Ghanbari, *Microprocessor and Assembly Language*, Sheikh Bahaie Publisher, 2015.

[2] M. A. Mazidi, S. Naimi, S. Naimi, *The AVR Microcontroller and Embedded Systems using Assembly and C*, Prentice Hall, 2010**.**

[3] M. A. Mazidi, D. Causey, R. D. McKinlay, *PIC Microcontroller and Embedded Systems using Assembly and C for PIC18*, Prentice Hall, 2008**.**

[4] R. H. Barnett, S. Cox, L. O'Cull, *Embedded C Programming and the Atmel AVR*, Delnmar Cengage Learning Publishing, 2011.

[5] B. van Dam, *Microcontroller System Engineering, 45 projects for PIC, AVR and ARM*, 2008.

[6] T. Wilmshurst, *Designing Embedded Systems with PIC Microcontrollers, Principles and Applications*, 2nd Edition, Newnes, 2010.

[7] *Atmel and ARM Microcontrollers' Datasheets*.

3-1-18 Operating Systems (CE303)

Operating Systems (CE303)		
Units	Prerequisites	Corequisites
3	Computer Architecture	Operating Systems Laboratory

Objectives:

The main objective of this course is to introduce structure and organization of operating systems. Students are being familiar with basic concepts of operations systems including internal structure, interrupts, system calls, processor internal capability, multicores and multi processors issues, process synchronization, CPU scheduling, memory management, protection and security issues, in this course.

Topics:

- Introduction and operating-system structures
- Process management
- Threads and concurrency
- CPU scheduling
- Process synchronization, tools and examples
- Deadlocks, livelock and starvations
- Main memory management
- Virtual memory management
- Mass-storage structure
- I/O systems
- File-system interface
- Protection and security
- Virtualization and virtual machines

References:

[1] A. Silberschatz, P.B. Galvin, G. Gagne, *Operating System Concepts*, 9th Edition, Wiley, 2013.

[2] W. Stallings, *Operating Systems: Internal and Design Principles*, 9th Edition, Pearson, 2015.

[3] A.S. Tanenbaum, *Modern Operating Systems*, 4th Edition, Pearson, 2014.

3-1-19 Operating Systems Laboratory (CE304)

Operating Systems Laboratory (CE304)		
Units	Prerequisites	Corequisites
3	---	Operating Systems

Objectives:

This course lab aimed to test and to do experiments of operating systems concepts. The course is divided into two parts: 1) an ability to work with well-known open-source operating systems like Linux and have kernel module programming (practical skills) and 2) an ability to implement operating system concepts and have experience what are problems and how to solve them.

Topics:

- Linux introduction
  - Definition and Installing
  - Filesystem and file management
  - Protection and permission
- Kernel module programming
  - Create a kernel module
  - Load and remove a kernel module
  - Kernel structure
- Bash scripting
  - Variable initialization and system variables
  - Condition statement, case statement, loop template
- Thread and process
- Inter-process communication
- Synchronization
- Deadlock
- Scheduling
- OpenMP
- Final project

References:

[1] M. Garrels, *Introduction to Linux, a Beginner's Guide*, 2010.

[2] M. Mitchell, J. Oldham, and A. Samuel, *Advanced Linux Programming*, 2001.

[3] A. Silberschatz, P.B. Galvin, G. Gagne, *Operating System Concepts*, 9th Edition, Wiley, 2013.

[4] W. Stallings, *Operating Systems: Internal and Design Principles*, 9th Edition, Pearson, 2015.

[5] A.S. Tanenbaum, *Modern Operating Systems*, 4th Edition, Pearson, 2014

3-1-20 Computer Networks (CE305)

Computer Networks (CE305)		
Units	Prerequisites	Corequisites
3	Engineering Probability and Statistics, Computer Architecture	Operating Systems, Computer Networks Laboratory

Objectives:

This course provides an introduction to fundamental concepts in the design, implementation and performance evaluation of computer networks. In this course, students get familiar with computer networks architecture, services, and applications. This course studies the application layer, the transport layer, the network layer, and the data link layer protocols with emphasis on the Internet and TCP/IP model.

Topics:

- Introduction
  - Review of Computer Networks Services
  - Definition of Computer Networks, Service, Quality of Service, and Protocol
- The Internet and Its Components
  - The Edge and Core Networks
  - Client-Server Model
  - Access Networks
  - Circuit Switching and Packet Switching Networks
  - Quality of Service Parameter in Packet Switch Networks
- Layered Architecture of Computer Networks
  - OSI Reference Model
  - Protocols and Services
  - Connection-Oriented and Connectionless service Models
  - Segmentation and Reassembly
  - Multiplexing and Demultiplexing
  - TCP/IP Model
- Application Layer
  - The Web and HTTP
  - FTP Protocol
  - Electronic Mail and SMTP Protocols
  - The Internet's Directory Service and DNS Protocol
  - Peer-to-Peer Applications
  - Socket Programming

- Transport Layer
  - Introduction and Transport-Layer Services
  - Connectionless Service of Transport Layer and UDP Protocol
  - Principles of Reliable Data Transfer
  - Connection-Oriented Service of Transport Layer and TCP Protocol
  - Principles of Congestion Control
  - TCP Congestion Control
- The Network Layer
  - Introduction and Network Layer Services (Forwarding and Routing)
  - Virtual Circuit and Datagram Networks
  - Router Architecture
  - Traffic Management in Packet Switch Networks
  - The Internet Protocols (IPv4, IPv6, ICMP, ARP Protocols)
  - DHCP and Mobile IP Protocols
  - Routing Algorithms (Link-State and Distance-Vector Routing Algorithms)
  - Routing Protocols in the Internet
- The Link Layer and Local Area Networks
  - Introduction and Data Link layer Services
  - Introduction to Error-Detection and Error-Correction Techniques
  - Introduction to Medium Access Control Techniques, Ethernet and Wireless LANs

References:

[1] J. F. Kurose, K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th Edition, Pearson, 2016.

[2] A. Leon-Garcia, I. Widjaja, *Communication Networks*, 2nd Edition, McGraw-Hill, 2003.

[3] A. S. Tanenbaum, *Computer Networks*, 5th Edition, Pearson, 2010.

[4] B. A. Forouzan and DeAnza College, *Data Communications and Networking*, 5th Edition, McGraw-Hill, 2012.

[5] W. Stallings, *Data and Computer Communications*, 10th Edition, Pearson, 2014.

3-1-21 Computer Networks Laboratory (CE306)

Computer Networks Laboratory (CE306)		
Units	Prerequisites	Corequisites
1	---	Computer Networks

Objectives:

The goal of the computer networks laboratory is to create the correct understanding of the basic concepts of computer networks. In this laboratory, the students will be familiar with tools and equipment needed to design, implement, and management of computer networks. They also perform some practical experiments based on theoretical concepts.

Topics:

- Introduction
- Simple tools for network management and performing the necessary experiments
- Packet Capturing tools and performing the necessary experiments
- Simulation and emulation tools of network and its equipment, and performing the necessary experiments
- Perform experiments to understand the application layer protocols
- Perform experiments to understand the transport layer protocols
- Perform experiments to understand the packet forwarding function
- Perform experiments to understand static and dynamic routing, and routing protocols
- Perform experiments for familiarity with wireless LANs
- Perform experiments for familiarity with Ethernet
- Perform experiments for familiarity with Ethernet network cabling

References:

[1] J. F. Kurose, K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th Edition, Pearson, 2016.

3-1-21 Software Engineering I (CE307)

Software Engineering I (CE307)		
Units	Prerequisites	Corequisites
3	Advanced Programming	---

Objectives:
The purpose of this course is to address engineering issues that must be followed in all stages of software development. In this course, students will get acquainted with the activities and tools needed to produce a software product.

Topics:

- Introduction
  - System Analysis and Design and Software Engineering
  - Components of information systems and dimensions of system analysis and design and software lifecycle
- Software development processes
  - Guiding principles in all software development activities
  - Structured processes
  - Agile processes
- System analysis
  - Requirements engineering
  - Scenario based modeling and use cases
  - Data analysis and modeling
  - Process analysis and modeling
  - Object-oriented analysis and modeling
  - Documentation using UML
  - Feasibility studies based on requirements engineering
- System design and implementation
  - Different design approaches
  - System architecture
  - Model-based structured design
  - Object-oriented design
  - Software development
- Umbrella activities
  - Different approaches and stages of design and implementation of software tests
  - Preliminary introduction to Project Management
- Configuration and documentation management

References:

[1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 7th Edition, 2011.

[2] J. Whitten, L. Bentley, *Systems Analysis and Design Methods*, McGraw-Hill, 7th Edition, 2005.

[3] M. Fowler, *UML Distilled*, 3nd Edition, OMG Standard Group.

3-1-23 Research and Presentation Methods (CE308)

Research and Presentation Methods (CE308)		
Units	Prerequisites	Corequisites
2	Technical English	---

Objectives:

The purpose of this course is to develop students' skills for conducting research and presenting the results effectively. Steps in doing research are discussed and appropriate presentation of the results in both oral and written forms is taught. Each student practices the lessons with selecting a topic and doing different taught steps during the semester. Students get familiar with software tools aiding in research and in written and oral presentations.

Topics:

- Introduction
  - Research in engineering
  - Good presentation: factors and impact
- Basic concepts
  - Ethics in research
  - Research question and goals
  - Mind map
  - Fish bone diagram
  - Research methodology in engineering
  - Scientific methodology
  - Modeling and simulation
- Research proposal
  - Research topic and title
  - Preparing a research proposal
  - Undergraduate and postgraduate thesis proposals
  - Scheduling and budget estimation
- Conducting research
  - Acquiring information from the Internet
  - Variety of resources
  - Resource validation and selection,
  - Study of the subject background
  - Paper reading and note-taking,
  - Critical points in conducting empirical research
  - Writing the first draft of the report
  - Introducing software tools for research

- Oral Presentation
  - Organizing and scheduling oral presentations,
  - Presentation plan
  - Critical points to observe before and during presentation
  - Slide preparation
  - Software tools for oral presentation
- Special types of oral presentation
  - Seminars
  - Theses defense
  - Conference presentations
  - Interview tips
- Written presentation
  - Standard elements of an engineering report
  - Reference writing styles and citation methods
  - Plagiarism, writing quality
  - Editing the report
  - Software tools for written presentation
- Special types of written presentation
  - Seminars
  - Thesis report
  - Internship report
  - Papers
  - Resume

References:

[1] R. Safabakhsh, *Research and Presentation in Engineering (in Farsi)*, AUT Press, 2013..

[2] A. Aliahmadi, *A Comprehensive Description of Research Methods (in Farsi)*, Tolide Danesh Pub., 2007.

[3] G. Khaki, *Research Methods with Thesis Writing Approach (in Farsi)*, Baztab Pub., 2003.

[4] S. Taiebi, M.R. Maleki, and B. Delgoshaee, *Preparation of Thesis, Dissertation, Research Proposals, and Scientific Papers (in Farsi)*, Ferdows Pub., 2009.

3-2 Packages Courses

3-2-1 Algorithm Design (CE221)

Algorithm Design (CE221)		
Units	Prerequisites	Corequisites
3	Data Structures and Algorithms	---

Objectives:
The main goal of this course is teaching the basic concepts and typical methods of design and analysis of algorithms. In this course, students learn how to analyze a given problem and develop algorithms to solve it. Then, they learn how to analyze and compare the time complexity of the algorithms and select the best one. In this course, also the basic algorithms for solving some well-known problems are introduced.

Topics:

- Introduction
  - Concept of algorithm
  - Motivation of studying design and analysis of algorithms
  - Introducing some basic concepts
  - Asymptotic analysis
  - Amortized analysis
- Correctness proof of algorithms
  - Induction
  - Loop invariants
- Divide and conquer
  - Merge sort
  - Quick sort
  - Strassen approach for matrix multiplication
- Dynamic Programming
  - Longest Common Subsequence Problem
  - Optimal Binary Search Tree
  - Matrix Multiplication Problem
- Greedy Approach
  - Task Scheduling problem
  - Activity Selection problem
- Backtracking and Branch and Bound
  - n-Queens problem

- Graph coloring problem
- Knapsack Problem

- Graph Algorithms
  - Shortest path algorithms (Dijkstra, Floyd)
  - Minimum spanning tree algorithms (Prim, Kruskal)
  - Max Flow algorithms
- Computation complexity
  - Introduction to classes of P, NP, NP-complete, NP-hard

References:

[1] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to Algorithms*, MIT Press, 3rd ed., 2009.

[2] R. Neopolitan, *Foundation of Algorithms*, Jones & Bartlett Learning, 2015.

[3] E. Horowitz, S. Sahni, and S. Rajasekaran, *Fundamentals of computer algorithms*, Galgotia Pub., 2008.

3-2-2 Signal and Systems (CE222)

Signal and Systems (CE222)		
Units	Prerequisites	Corequisites
3	Differential Equations	---

Objectives:

This course introduces the methods of description and analysis for continuous-time and discrete-time signals and (linear time-invariant) systems, including time domain analysis using convolution, frequency domain analysis using the Fourier series and the Fourier transform, the Laplace transform and the Z-transform. Students will also learn examples of applications of signal and system analysis methods to communications and control systems.

Topics:

- Introduction and Preliminary Concepts
  - Discrete-Time and Continuous-Time Signals
  - Systems Types
  - Basic Signals
  - Systems Properties
- Linear Time-Invariant Systems
  - Continuous and Discrete Convolution
  - Convolution Properties
  - Impulse Response and Its Properties
  - System Block Diagram
- Continuous-Time Fourier Series
- Continuous-Time Fourier Transform
- Applications of Continuous-Time Fourier Transform
  - Sampling
  - Filtering
  - Communication Systems
- Discrete-Time Fourier Series
- Discrete-Time Fourier Transform
- Z-Transform
- Laplace Transform

References:

[1] A.V. Oppenheim, A.S. Willsky, S.H. Nawab, *Signals and Systems*, 2nd Edition, Prentice-Hall, 1997.

3-2-3 Principles of Database Design (CE231)

Principles of Database Design (CE231)		
Units	Prerequisites	Corequisites
3	Data Structures and Algorithms	---

Objectives:

Database management has evolved from a specialized computer application to a central component of a modern computing environment, and, as a result, knowledge about database systems has become an essential part of an education in computer science. In this course, the fundamental concepts of database management including aspects of database design, database languages, and database-system implementation will be presented.

Topics:

- Introduction to Database
- Introduction to the Relational Model
- Formal Languages
- Introduction to SQL
- Intermediate and Advanced SQL
- Database Design and the Entity Relationship Model
- Normalization (1st, 2nd, 3rd and BCNF)
- Higher Order Normalization
- Advanced Topics in Database
    - Selecting at least two of the following topics: Object-based Data Models, XML, NoSQL, OLAP, Data Warehouse

References:

[1] A. Silberschatz, H. F. Korth, S. Sudarshan: *Database System Concepts*, 6th Edition, McGraw-Hill, 2010.

[2] C. J. Date, *Introduction to Database Systems*, 8th Edition, Addison-Wesley, 2003

3-2-4 Compiler Design (CE232)

Compiler Design (CE232)		
Units	Prerequisites	Corequisites
3	Data Structures and Algorithms	---

Objectives:

The objective of this course is to learn the principles, techniques and tools for constructing a compiler. In this course, after introduction of the basic concepts of compiler design, lexical and syntax analyses are discussed. After that, run-time environment as well as intermediate code generation concepts are explained.

Topics:

- Introduction to compiler
- Lexical Analysis
- Syntax Analysis
  - Top down parser
  - bottom up parser
- Symbol table and Run-Time Environment
- Intermediate Code Generation
- Project: Design and implementation of a front end of a compiler using Compiler-Compilers

References:

[1] A. Vlahos, R. Sethi, J. D. Ullman, *Compilers: Principles, Techniques, and Tolls*, 2nd Edition, Addison-Wesley, 2007.

[2] D. Grune, H. Bal, C. Jacobs, K. Langendoen, *Modern Compiler Design*, JohnWiley & Sons, 2000.

3-2-5 Programming Languages (CE233)

Programming Languages (CE233)		
Units	Prerequisites	Corequisites
3	Theory of Languages and Machines	---

Objectives:

This course acquaints students with the concepts and constituents of programming languages. It also surveys the constructs and capabilities typically found in modern programming languages with attention to design trade-offs and implementation considerations. The course also develops the skills necessary to learn new programming languages quickly. Learning the capabilities of different classes of programming languages as well as familiarizing the students with instances of each class is among the objectives of this course.

Topics:

- Introduction
  - Programming Languages
  - Computational Models
  - Classes of Programming Languages
  - Language and Computability
- Syntax
  - Formal Syntax
  - Concrete and Abstract Syntax
- Dynamic Semantics
  - Operational
  - Denotational
  - Axiomatic Semantics
- Lambda Calculus
  - Syntax
  - Semantics
  - Recursive Functions
- Functional Languages
  - Lisp
  - ML
  - Haskell
- Static Semantics
  - Type Systems
  - Safe Languages

- Type Safety
- Typed Lambda Calculus
- Type Inference

- Scope, Functions, and Memory Management
  - Block-Structured Languages
  - Activation Records
  - Closures
- Control in Sequential Languages
  - Structured Control
  - Exceptions
  - Continuations
  - Functions and Evaluation Order
- Modularity, Abstraction, and Object-Oriented Programming
  - Structured Programming,
  - Modules
  - Concepts in Object-Oriented Languages
  - Simula, Smalltalk, C++, and Java
- Concurrent and Distributed Programming
  - Basic Concepts in Concurrency
  - Actor Model
  - Core Calculus of Concurrency
  - Concurrent ML
  - Concurrency in Java
- Logic Programming
  - Concepts of First-Order Logic
  - Resolution
  - Prolog

References:

[1] J. C. Mitchell, *Concepts in Programming Languages*, Cambridge University Press, 2002.

[2] R. W. Sebesta, *Concepts of Programming Languages*, 11th Edition, Pearson, 2015.

[3] B. C. Pierce, *Types and Programming Languages*, MIT Press, 2002.

3-2-6 Interface Circuits Design (CE241)

Interface Circuits Design (CE241)		
Units	Prerequisites	Corequisites
3	Microprocessor and Assembly Language	---

Objectives:

The objectives of this course is

- To provide basic concepts and principles of hardware and software interfaces without a focus on a particular architecture or processor
- To learn the main principles of commonly used interface standards and protocols
- To develop skills to design and implement interface circuits

Topics:

- Introduction to interfaces
  - Types
  - Hardware/Software
  - Analog/Digital
- Buses
  - Theoretical aspects
  - Different types
  - Applications
  - Comparison
- Different bus architectures
  - CPU-Memory
  - I/O
  - DMA
  - etc.
- Universal Serial Bus (USB)
- IEEE 1394 Standard (Firewire)
- AMBA AHB/APB
  - Overall architecture and design with a focus on FPGA
- PCI family of interfaces
- Storage device interfaces (SATA)
- Memory chip interfaces (DDR)
- Multimedia interfaces (HDMI)
- Debugging interfaces (SPI and I2C)

- IEEE 1149.1 Standard (JTAG)
- Device drivers (firmware interfaces)
- Motors (stepper, servo, DC) and their drivers
- Sensors and actuators
  - Theoretical aspects
  - Design and analysis

References:

No single textbook covers all of the material presented in this course. The main source of the course will be the text of standards and protocols. Only the last topic in the course will be based on Chapter 7 of the following textbook:

[1] E. Lee and S. Seshia, *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, 2nd Edition, 2015.

3-2-7 Programmable Digital Systems Design (CE242)

Programmable Digital Systems Design (CE242)		
Units	Prerequisites	Corequisites
3	Computer Architecture	---

Objectives:

The objectives of this course are:

- To develop skills in design, verification, debugging and optimization of large digital circuits and systems
- To develop skills in describing digital circuits and systems in hardware description languages
- To enable students to implement a complete system on a development board based on programmable devices
- To gain familiarity with architectures of programmable logic devices

Topics:

- Digital systems design flow
- Hardware description: basic concepts and applications
- Levels of abstraction
  - Behavioral level
  - Register-transfer level
  - Gate level
- Describing hardware in one of hardware description languages
- Synthesis and synthesizable description
- Designing in register-transfer level; control path/data path
- Verification and simulation
- Placement and routing
- Digital systems analysis
  - Timing analysis
  - Power analysis)
- Design techniques for optimizing designs and design process
  - Pipelining
  - Retiming
  - Ad-hoc techniques)
- Introduction to hardware/software co-design
- Commercial programmable logic devices and their architectures

References:

[1] P. Simpson, *FPGA Design: Best Practices for Team-based Reuse*, Springer, 2015.

[2] P. Chu, *RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability*, Wiley, 2006.

[3] P. Wilson, *Design Recipes for FPGAs*, 2nd Edition, Elsevier Science & Technology, 2015.

[4] C. Maxfield, *The Design Warrior's Guide to FPGA*, Elsevier, 2004

3-2-8 Digital Electronics (CE243)

Digital Electronics (CE243)		
Units	Prerequisites	Corequisites
3	Electrical and Electronic Circuits	---

Objectives:

Digital Electronics course provides basic concepts and principles of semiconductor devices with an emphasis on Metal Oxide Semiconductor Field Effect Transistors (MOSFET) along with an in-depth knowledge of CMOS semiconductor devices. In addition, it provides skills to model, analyze, and simulate digital electronic circuits at logic and transistor levels.

Topics:

- Introduction to MOSFETs
  - Behavior and characteristics
  - Shockley model
  - I-V curves
  - A brief overview of CMOS fabrication process
- NMOS inverters
  - Circuit models
  - Advantages and disadvantages
  - Comparison between resistive load vs. active load
- CMOS inverter
  - Circuit model
  - DC characteristics
  - Delay and timing behavior
  - Parasitic capacitances
  - Power consumption
- CMOS complex gates
  - Theoretical and practical analysis
- CMOS buffers and drivers
  - Theoretical and practical analysis
- CMOS tristate and Schmitt trigger inverters
- Pass transistors and transmission gates
- CMOS sequential circuits
- Dynamic CMOS, Differential CMOS circuits
- Introduction to BiCMOS technology

References:

[1] M. Sedighi, A. Valizadeh, *Digital Electronic Devices*, Amirkabir University Press, 2016.

[2] N. Weste, D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th Edition, Addison-Wesley Press, 2011.

3-2-9 Principles and Applications of Artificial Intelligence (CE251)

Principles and Applications of Artificial Intelligence (CE251)		
Units	Prerequisites	Corequisites
3	Data Structures and Algorithms	Applied Linear Algebra

Objectives:

The goal of this course is to familiarize students with the basics, methods and applications of artificial intelligence. Artificial intelligence definitions, rationality concept, task environment properties and agent architectures are introduced in the first part of the course. Students learn how to model real world problems as search/optimization problem and how to solve them using informed and uninformed search strategies. Propositional and first order logic are also covered as knowledge representation and inference tools. A number of artificial intelligence applications including text processing, visual perception, and robotics are also introduced.

Topics:

- Introduction
- Intelligent Agents
- Problem solving with search
- Adversarial search
- Constraint satisfaction problems
- Non-classical search algorithms
- Text, speech and natural processing
- Visual perception
- Logical Agents
- First-order logic
- Knowledge representation
- Text, speech and natural processing
- Visual Perception
- Robotics

References:

[1] S. J. Russel, P. Norvig, *Artificial Intelligence, A Modern Approach*, 3rd Edition, Pearson Education, 2009.

[2] A. Konar, *Computational Intelligence: Principles, Techniques and Applications*. Springer, 2007.

3-2-10 Web Programming (CE261)

Web Programming (CE261)		
Units	Prerequisites	Corequisites
3	Computer Networks	---

Objectives:

As the web is the main application on the Internet, the objective of this course is to learn the concepts, technologies, tools and programming languages used in both client and server side of web. In client side, students gradually learn from static page development to complex interactive pages. In server side, the main requirements of a typical application including input data handling, session management, and DB access are discussed.

Topics:

- HTTP
- HTML
- CSS
- XML & JSON
- JavaScript
- HTML5
- Ajax
- A client side framework (e.g. Vue)
- A server side programming language (e.g. PHP)
- Input processing
- Session management
- Database Access and ORM
- Concurrency Control
- Error Handling & Debugging
- Security
- Web Application Architectures

References:

[1] J. Duckett, *HTML & CSS: Design and Build Websites*, Wiley, 2011.

[2] J. Duckett, *JavaScript and jQuery: Interactive Front-End Web Development*, Wiley, 2014.

[3] K. Yank, *PHP & MySQL: Novice to Ninja*, 6th Edition, SitePoint, 2017.

3-2-11 Software Engineering II (CE331)

Software Engineering II (CE331)		
Units	Prerequisites	Corequisites
3	Software Engineering I	---

Objectives:

This is the second of two courses in the Software Engineering. While Software Engineering I focuses on software production topics such as processes, requirements and architectures, Software Engineering II focuses on how the quality software will be. Much of the content is organized around the software processes, software planning, risk, estimation, and architecture and detailed design.

Topics:

- Introduction
- WebApp Design
- MobileApp Design
- Quality Concepts
- Review Techniques
- Software Quality Assurance
- Software Testing Strategies
- Testing Conventional Applications
- Testing Object-Oriented Applications
- Testing Web Applications
- Testing Mobile Apps
- Product Metrics
- Process and Project Metrics
- Estimation for Software Projects
- Risk management

References:

[1] R. Pressman, *Software Engineering -- A Practitioner's Approach*, 8th Edition, McGraw Hill, 2014.

[2] I. Sommerville, *Software Engineering*, 9th Edition , Addison-Wesley, 2011.

3-2-12 Embedded and Real-Time Systems (CE341)

Embedded and Real-Time Systems (CE341)		
Units	Prerequisites	Corequisites
3	Operating Systems, Microprocessor and Assembly Language	---

Objectives:

Embedded systems play a decisive role in various applications including healthcare, transportation, traffic control, financial and information systems, avionics, and home appliance. This course focuses on design, implementation, and analysis of embedded and real-time systems. Hardware and software components of an embedded system, task scheduling approaches, process and resource management in real-time systems, and programming techniques are covered in this course.

Topics:

- Preliminaries and basic concepts
  - Applications
  - Trends
  - Characteristics
  - Challenges
  - Computing paradigm
- Hardware components
  - Processors
  - Memory structure
  - Sensors and actuators
  - Communication
- State-machine model
  - System design
  - Interrupts
- Communications in embedded systems
  - Network protocols
  - Distributed embedded systems
  - Integrity in distributed embedded systems
- Operating system and embedded software
  - Real-time scheduling
  - Compilers and Low-level programming
  - Software optimizations
  - Internet-of-Things

References:

[1] E. A. Lee, S. A. Seshia, *Introduction to Embedded Systems – A Cyber-Physical System Approach*. MIT Press, 2016.

[2] P. Marwedel, *Embedded System Design (Embedded System Foundation of Cyber-Physical System)*, Springer, 2011.

[3] R. Love, *Linux Kernel Development*, 2nd Edition, Novell, 2005.

[4] S. Seiwert, *Real-Time Embedded System and Components*, Thomson Learning Inc, Charles River Media, 2007.

3-2-13 Multi-Processor Programming (CE342)

Multi-Processor Programming (CE342)		
Units	Prerequisites	Corequisites
3	Operating Systems	---

Objectives:

The goal of this course is to give students the skills to design, implement, debug and optimize multithreaded programs, as well as vectorized programs and GPU codes. Students will learn the basics of multicore and GPU architectures and their memory models. They also learn how to profile a given application, identify its performance bottlenecks and how to resolve them, and gain multicore programming skills.

Topics:

- Introduction to parallel systems architecture and memory models including shared memory and distributed memory models
- Introduction to multithreaded programming, programming models and related programming languages
- Introduction to vector processing, SIMD, SSE, AVX, and how to write a vectorized program
- Multithreaded and vectorized programming using multicore programming languages (OpenMP)
- Introduction to thread synchronization methods, lock, barrier, etc.
- Practical multithreaded applications (matrix operation, sorting, etc.)
- Introduction to GPU architecture and GPU memory hierarchy
- Introduction to multithreaded programming for general purpose application development on GPUs
- Implementing general purpose applications on GPU using CUDA
- Practical GPU applications (matrix operation, n-body simulation, etc.)

References:

[1] T. Rauber, G. RüngerParallel, *Programming for Multicore and Cluster Systems*, Springer, 2013.

[2] N. Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*, Addison-Wesley, 2013.

[3] D. Kirk, *Programming Massively Parallel Processors: A Hands-on Approach*, Elsevier, 2010.

[4] J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General–Purpose GPU Programming*, Addison-Wesley, 2010.

3-2-14 Foundations of Computational Intelligence (CE351)

Foundations of Computational Intelligence (CE351)		
Units	Prerequisites	Corequisites
3	Algorithm Design	---

Objectives:

This course is designed to familiarize the students with fundamental concepts and principles of intelligent methods for data analysis and engineering problem solving, such as fuzzy methods, evolutionary techniques, and neural networks. To achieve this goal, students get hands-on experience with software tools related to these areas.

Topics:

- Introduction to Computational Intelligence
- Fuzzy Methods
  - Introduction, Basic fuzzy set theory (Membership functions, Fuzzy operators)
  - Fuzzy relations and fuzzy logic inference (Fuzzy rules, Extension Principle)
  - Fuzzy Knowledge-based Systems
- Evolutionary Methods
  - Introduction, Genetic algorithm (Representation, Crossover, Mutation, Selection)
  - Evolution strategies (Adaptation, Self-adaptation, Representation, Crossover, Mutation, Selection)
  - Particle swarm optimization (Global best, Local best, Inertia weight)
  - Ant Algorithms (Ant Systems, Ant Colony, Max-Min Ant)
- Neural Networks
  - Introduction,
  - Artificial neurons (Activation Function, Learning, Perceptron, Adaline)
  - Supervised neural networks (Feedforward Neural Networks)
  - Unsupervised neural networks (Self-organizing feature maps, Learning Vector Quantization networks)
- Hybrid CI Algorithms

References:

[1] J. M. Keller, D. Liu and D. B. Fogel, *Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems and Evolutionary Computation*, Wiley, 2016.

[2] P. Engelbrecht, *Computational Intelligence: An Introduction*, Wiley, 2007.

[3] Konar, *Computational Intelligence: Principles, Techniques and Applications*, Springer, 2007.

3-2-15 Introduction to Robotics (CE352)

Introduction to Robotics (CE352)		
Units	Prerequisites	Corequisites
3	Signal and Systems	---

Objectives:

The purpose of this course is to introduce basics of mechanism, design, planning, programming and intelligent control of robot systems. The main focus is on different aspect of mobile robots including, kinematics, sensors, actuators, perception, localization, motion planning and navigation. This course is a laboratory based course which students practice via real and simulated robots.

Topics:

- Introduction to mobile robotics and robotic manipulation
- Robot simulators and mobile robot design
- Spatial descriptions, homogeneous transform, and mobile robot kinematics
- Robot sensors, measurement and calibration
- Actuators and drives, DC motors, stepper motors, servomotors, gears, pulse width modulation (PWM)
- Robot Control basics and principles and PID controllers
- Robot perception, localization, planning, and navigation
- Introduction to Kalman filter, program a robot using ROS

References:

[1] R. Siegwart, I. Nourbakhsh, D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, The MIT Press, 2004.

[2] G. McComb, *Robot Builder's Bonanza*, 4th Edition, McGraw-Hill, 2011.

[3] T. Braünl, *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*, 3rd Edition, Springer, 2008.

3-2-16 Data Communication (CE361)

Data Communication (CE361)		
Units	Prerequisites	Corequisites
3	Computer Networks, Signal and Systems	---

Objectives:

This course examines the technologies for the implementation of computer and communication networks. Students will be introduced to the principles of digital transfer, shared media technologies, Ethernet and wireless local area networks, and advanced computer network architectures.

Topics:

- Introduction
  - Overview of computer networks
  - Internet networks
  - OSI and TCP / IP layer models
- Principles of digital transmission
  - Digital representation of information
  - Digital channel characteristics
  - Nyquist rate for signal transmission
  - Shannon’s channel capacity
  - Digital modulation
  - Media features and digital transmission systems
- Error detection and correction methods
  - FEC and ARQ error detection methods
  - Error detection and correction codes
  - ARQ error control protocols
- Data Link Layer
  - Introduction of Data Link Layer Services
  - Point-to-Point Protocol
  - HDLC Protocol
- Medium Access Control
  - Introducing shared channels and point-to-point channels
  - Random access methods
  - Scheduling methods
  - Channelization methods

- Local Area Networks
  - IEEE 802 standards for local area networks
  - Ethernet networks
  - Wireless local area networks
  - Local area network bridge
  - Ethernet switches and VLANs
- Advanced computer network architectures
  - Label switching and MPLS
  - Software-Defined networks

References:

[1] A. Leon-Garcia, I. Widjaja, *Communication Networks*, 2nd Edition, McGraw-Hill, 2003.

[2] W. Stallings, *Data and Computer Communications*, 10th Edition, Pearson, 2014.

[3] J F. Kurose, K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th Edition, Pearson, 2016.

[4] A. S. Tanenbaum, *Computer Networks*, 5th Edition, Pearson, 2010.

[5] B. A. Forouzan, D. College, *Data Communications and Networking*, 5th Edition, McGraw-Hill, 2012.

3-2-17 Information and Communications Security (CE362)

Information and Communications Security (CE362)		
Units	Prerequisites	Corequisites
3	Computer Networks	---

Objectives:

The main objective of the course is security in information technology and network communications. In more details, the students will learn foundations of security and security mechanisms, specially cryptography, authentication, and security challenges and solutions in network layers.

Topics:

- Introduction
  - Security definition and dimensions
  - Threats and attacks categories
- Symmetric Cryptography
- Message Authentication and Public key cryptography
- User Authentication
- Email Security (PGP, S/MIME)
- Transport Layer Security
- IP security and VPN
- Intrusion detection systems and firewalls
- Malicious codes and malwares

References:

[1] W. Stallings, *Network Security Essentials, Application and Standard*, 6th Edition, Prentice-Hall, 2017.

[2] T. R. Peltier, *Information Security Fundamentals*, 2nd Ed., CRC Press, 2014.

[3] M. Goodrich, R. Tamassia, *Introduction to Computer Security*, Pearson Education, 2014.

3-2-18 Multimedia Systems (CE363)

Multimedia Systems (CE363)		
Units	Prerequisites	Corequisites
3	Engineering Probability and Statistics, Signal and Systems	---

Objectives:

Multimedia content is one of the main ingredients of any network-connected computer system. Multimedia systems play an ever-increasing role in most aspects of information technology, computer architecture, computer software, computer networks and application programs. The main goal of this course is to introduce the principles of multimedia systems, applications and communication. The students are to learn the acquisition, representation, compression and transfer of multimedia data. In addition, current standard data compression techniques and standards are covered. We also discuss challenges such as resilience against packet loss and quality of service in transferring multimedia content over wireless and IP networks.

Topics:

- Principles of color and digital imaging
- Basics of lossless compression
  - Entropy coding
  - Huffman coding
  - Dictionary coding such as LSW, LZ77
  - Run Length coding
- Basics of lossy compression
  - Vector quantization
  - Rate-distortion relationship
- Fundamentals of compression standards
  - DCT, JPEG, DWT, JPEG2000
- Rate Allocation
- Principles of digital video
- Basics of video compression
- Video compression standards
  - H.261, H.263, H264, MPEG-1, MPEG-2, MPEG-4
- Principles of digital audio
- Standards of audio compression
  - AC-3, AAC, MP3
- MPEG Systems
- Principles and standards of multimedia conferencing

- SIP and H.32x
- Principles of multimedia transfer over computer and telephone networks
  - Error detection and recovery
  - Error concealing
  - Resilience
  - Quality of service (QoS)
  - Quality of experience (QoE)
  - Multicasting, and streaming

References:

[1] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, 3rd Edition., Institution of Engineering and Technology, 2011.

[2] R. Steinmetz, K. Nahrrstedt, *Multimedia Systems*. Springer, 2010.

[3] T. E. Richardson. *The H.264 Advanced Video Compression Standard*, 2nd Edition, Willey, 2010.

3-2-19 Mobile Device Programming (CE364)

Mobile Device Programming (CE363)		
Units	Prerequisites	Corequisites
3	Advanced Programming	---

Objectives:

The purpose of this course is to familiarize students with open source initiative and software for mobile devices. After general introduction to mobile development in different platforms, in particular, students will learn about Android operating system as well as Android programming environment. Wide range of topics including mobile devices hardware architecture, software architecture, operating systems, programming languages and development tools are covered. Through exercises and projects, students would gain skills necessary to develop and run a software application on a mobile device.

Topics:

- Introduction
  - Mobile devices
  - Mobile applications
- Introduction to Android
  - Android Features
  - Architecture
  - Application Components
- Android programming basics
  - Android Installation and development tools
- Android UI, Views, and Widgets
- Intent, Permissions, and Fragment concepts
- Storing, retrieving, and managing information
- Threads and Asynchronous tasks
- Notifications and Alarms
- Content Providers
- Event handling
- Camera handling
- State management
- Service
- Broadcast Receiver
- Graphic and animation

- Multimedia
- Geo-Location handling
- Custom Components

References:

[1] D. Griffiths, D. Griffiths, *Head First Android Development: A Brain-Friendly Guide*, 1st Edition, O'Reilly, 2015.

[2] Z. Mednieks, L. Dornin, *Programming Android: Java Programming for the New Generation of Mobile Devices*, O'Reilly, 2012.

[3] M. L. Murphy, *The Busy Coder's Guide to Advanced Android Development*, 8th Edition, CommonsWare, 2009.

[4] J. Iversen, M Eierman, *Learning Mobile App Development*, Addison-Wesley, 2011.

3-2-20 Information Retrieval (CE421)

Information Retrieval (CE421)		
Units	Prerequisites	Corequisites
3	Data Structures and Algorithms, Engineering Probability and Statistics	---

Objectives:

Information retrieval is the process through which a computer system can respond to a user's query for text-based information on a specific topic. Web search is one of the most important applications of information retrieval techniques and an area in which most people interact with IR systems. The goal of this course is to introduce students with the basics, models, tools and applications of the modern information retrieval.

Topics:

- Text Preprocessing and vocabulary construction
  - Document and word separation
  - Normalization
  - Stemming and lemmatization
  - Spelling correction
- Indexing
  - Index construction
  - Index compression
- Retrieval and ranking methods
  - Boolean, Vector-based, Probabilistic Retrieval
- Performance evaluation for information retrieval methods
- Query languages and operators
- Document classification and clustering
- Web search
  - Basics
  - Crawling
  - Link analysis
- IR-based applied systems

References:

[1] C. D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.

3-2-21 Fundamentals of Cloud Computing (CE422)

Fundamentals of Cloud Computing (CE422)		
Units	Prerequisites	Corequisites
3	Computer Networks, Operating Systems	---

Objectives:

In this course, students will learn the basic concepts of cloud computing, the architecture of cloud computing infrastructure and knowhow of cloud-based applications creation. Virtualization technologies in computing, networking and data storage and fundamental concepts of cloud computing are also covered.

Topics:

- The basic concepts of cloud computing
- A brief history and evolution of cloud computing
- Service models and standards
- OpenStack
  - Architecture and building blocks
  - Console and controller, Scheduler, Nova cert., Cinder services, Swift object storage, Neutron networking, Keystone, Glance image services, Heat configuration service
- Hadoop HDFS distributed file system
  - Hadoop MapReduce
  - Hadoop Zookeeper, Oozie, Flume, Sqoop
- Cloud databases and storage technologies
  - HiveQl, Hive, Cassandra, MongoDB, HBase
- Cloud computing infrastructure
  - Data center architecture
  - Storage and networking technologies
- Virtualization
  - Virtualization concepts and techniques
  - Virtual SAN, Distributed virtual switch, Virtual machine, Containers

References:

[1] K. Chandrasekaran, *Essentials of cloud computing*, CRC Press, 2014

[2] N. B. Ruparelia, *Cloud Computing*, MIT Press, 2016

[3] T. Erl, Z. Mahmood, R. Puttini, *Cloud Computing, Concepts, Technology and Architecture*, Prentice Hall, 2013.

[4] I. Ganelin, E. Orhian, K. Sasaki, B. York, *Spark: Big Data Cluster Computing in Production*, Wiley, 1st Edition, 2016.

[5] T. White, *Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale*, 4th Edition. O'Reilly Media, 2015.

[6] H. Geng, *Data Center Handbook*. Wiley, 1st Edition, 2014.

3-2-22 Software Testing (CE431)

Software Testing (CE431)		
Units	Prerequisites	Corequisites
3	Software Engineering II	---

Objectives:
The goal of this course is to familiarize students with software testing method, especially those methods that generate test data. Furthermore, in this course, tools required for software testing are introduced.

Topics:

- Introduction to testing methods
- Coverage criteria
- Graph based criteria
- Logic-based criteria
- Inventory-based criteria
- Syntax-based criteria
- Practical considerations for software testing
- Software testing and adjustment tools for testing

References:

[1] P. Ammann, J. Offutt, *Introduction to software testing*, Cambridge University Press, 2016.

[2] P. C. Jorgensen, *Software Testing: A Craftsman's Approach*, 2nd Edition, CRC Press, 2002.

3-2-23 User Interface Design (CE432)

User Interface Design (CE432)		
Units	Prerequisites	Corequisites
3	Software Engineering I	---

Objectives:

This course provides an introduction to design user interfaces. It covers design principles, prototyping techniques, evaluation techniques, implementation of user interfaces, wireframes, color schemes, tones, and design templates. Moreover, it describes how to obtain information from a client to design and manufacture interfaces for different platforms, such as mobile, tablet and desktop. Finally, it briefly investigates well-known tools for generating and upgrading user interfaces.

Topics:

- Introduction to user interface design
  - Evolution of user interfaces
  - Current trends in user interfaces design
- Principles of user interface design
  - Usability, Learnability, Visibility, Efficiency, …
- User Interface basic concepts
  - Color schemes and tones
  - Typography
  - Layouts
  - Wireframes and mockups
  - Navigation
- User interface design process
  - User interface design methodology
  - Iterative design
  - User-centered design
- User interface design techniques
  - Sketching, scenarios, storyboards
  - Design patterns
  - Prototyping
- User interface validation and verification
- User interface evaluation techniques
- User interface tools

References:

[1] H. R. Hartson, P. S. Pyla, *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*, Morgan Kaufmann / Elsevier, 2012.

[2] B. Shneiderman, C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 4th Edition, Addison Wesley, 2004.

[3] J. Preece, Y. Rogers, H. Sharp, *Interaction Design: Beyond Human-computer Interaction*, 4th Edition, John Wiley & Sons Ltd, 2002.

[4] D. A. Norman, *The Design of Everyday Things*. Basic Books, 2002.

[5] A. J. Dix, J. E. Finlay, G. D. Abowd, R. Beale, *Human-Computer Interaction*. 2nd Edition. Prentice Hall, 1998.

[6] D. R. Olsen, *Developing User Interfaces (Interactive Technologies)*. Morgan Kaufmann, 1998.

3-2-24 Hardware-Software Codesign (CE441)

Hardware-Software Codesign (CE441)		
Units	Prerequisites	Corequisites
3	Computer Architecture	---

Objectives:

This course provides a basic insight on designing computing systems by joint consideration of software and hardware subsystems based on state-of-the-art system design methodologies. In addition, students learn designing models of hardware and software components in different abstraction layers as well as communication between application program and hardware.

Topics:

- Introduction
- System-level design (approaches and applications)
- An introduction to a system-level language (SystemC)
- Dataflow and control models
- Transaction-level models
- System-level synthesis and verification
- Hardware-software interface design
- Codesign optimization approaches
- Codesign implementation platform
- Behavioral synthesis

References:

[1] P. Schaumont, *A Practical Introduction to Hardware/Software Codesign*, Springer, 2013.

[2] F. Vahid, T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley & Sons, 2002.

[3] G. De Micheli, R. Ernst, W. Wolf, *Readings in Hardware/Software Codesign*, Morgan Kaufman, 2002.

3-2-25 Introduction to Bioinformatics (CE451)

Introduction to Bioinformatics (CE451)		
Units	Prerequisites	Corequisites
3	Foundations of Computational Intelligence	---

Objectives:

The main goal of this course is to introduce basic concepts and problems in bioinformatics and related algorithms and solutions. Because of wide and diversity of concepts in bioinformatics, this course intends to make students familiar with various concepts and sub domains in the field, so students become ready to enter in more specialized fields and carry out state-of-the-art research. On the other hand, this course, via practical assignments and projects, helps students develop basic skills required for practical careers, such as working with common bioinformatics databases and software.

Topics:

- Introduction to bioinformatics and its major domains
- Review of some basic concepts in biology
- Access to sequence data and related information
- Pairwise sequence alignment
- BLAST
- Advanced database searching
- Multiple sequence alignment
- Molecular phylogeny and evolution
- DNA: The eukaryotic chromosome
- Analysis of next-generation sequence data
- Bioinformatic approaches to RNA
- Gene expression: Microarray and RNA-seq data analysis
- Protein analysis and proteomics
- Protein structure
- Functional genomics
- Introduction to systems biology

References:

[1] J. Pevsner, *Bioinformatics and Functional Genomics*, John Wiley & Sons, 2015.

[2] A. Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2013.

[3] M. Zvelebil, J. Baum. *Understanding Bioinformatics*, Garland Science, 2007.

3-2-26 Data Mining (CE452)

Data Mining (CE452)		
Units	Prerequisites	Corequisites
3	Foundations of Computational Intelligence	---

Objectives:

Data mining provides methods for exploring data types and discovering knowledge from data. After general overview of data mining problems, this course deals with data preparation methods, mining association rules, clustering, classification and outlier detection. The main objectives of this course are as follow:

- Understanding algorithms and data mining techniques
- Developing data mining applications with existing data mining tools and programming languages
- Learn to analyze and visualize data mining results

Topics:

- Introduction (motivations and challenges)
- Data warehouse
- Types and characteristics of data sets
- Data preprocessing and preparation
  - Data cleaning
  - Outlier detection
  - Normalization,
  - Discretization
  - Dimensionality reduction
  - Regression
- Association rule mining
  - Apriori
  - FP-growth algorithms
- Clustering
  - Partitioning-based
  - Hierarchical
  - Density-based algorithms
- Classification
  - Decision tree and naïve Bayes
  - Evaluation methods
- Data summarization and visualization

- Outlier detection methods
- Recommender systems
  - Collaborative filtering
  - Content-based
- Advanced topics in data mining

References:

[1] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, 3rd Edition, McGraw-Hill 2012.

[2] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997

3-2-27 Internet of Things Fundamentals (CE461)

Internet of Things Fundamentals (CE461)		
Units	Prerequisites	Corequisites
3	Microprocessor and Assembly Language, Computer Networks	---

Objectives:

This course introduces Internet of Things (IoT) concepts, applications, architecture and its corresponding technologies as well as network protocols. Students will also learn how to choose an appropriate network for IoT applications, and acquire skill for designing, implementing, and debugging an IoT-based system. This course provides students with foundational skills by doing mini-projects.

Topics:

- Introduction to IoT
  - IoT Applications
  - IoT ecosystem
  - IoT architecture and standards
- IoT Hardware
  - Embedded systems and their application to IoT
  - Arduino platform and C programming
  - Introduction to the Raspberry platform, Raspbian operating system, and python programming, Connecting Raspberry board to Internet
  - IoT lab exercises
- IoT Connectivity and networks
  - Physical and Mac layers technologies
  - Network, transport, and application layers protocols
  - IoT lab exercises
- IoT platforms
  - Introduction of existing open-source and commercial platforms
  - Connecting devices to platform
  - Creating an application for IoT
  - Data analytics
  - IoT lab exercises
- Security and privacy in IoT

References:

[1] O. Hersent, D. Boswarthick, O. Elloumi, *The Internet of Things: Key Applications and Protocols*, 2nd Edition, Wiley, 2013.

[2] M. Margolis, *Arduino Cookbook*, 3rd Edition, O'Reilly, 2017.

[3] S. Monk, *Raspberry Pi Cookbook*, 2nd Edition, O'Reilly, 2016.

3-3 Optional Courses

3-3-1 Computer Graphics (CE371)

Computer Graphics (CE371)		
Units	Prerequisites	Corequisites
3	Advanced Programming	---

Objectives:

The objective of this course is to introduce the basic principles of computer graphics with the emphasize on game development. In this course, the techniques of the real-time animation are covered. Students learn the main techniques including pipeline, lighting, shader, post processing, HDR, Bloom, and etc.

Topics:

- Mathematics of computer graphics
- Introduction to graphic pipeline
- Sprite tiling
- Basic geometry
- Ray tracing
- Shader
- Rasterization
- Global Illumination
- HDR ،DOF ،Tome Mapping
- Deferred rendering

References:

[1] T. Akenine-Moller, E. Haines, N. Hoffman, *Real-Time Rendering*, 3rd Edition, O'Reilly, 2008

[2] A. Sherrod, *Game Graphics Programming*, O'Reilly, 2008

3-3-2 Human and Computer Interaction (CE372)

Human and Computer Interaction (CE372)		
Units	Prerequisites	Corequisites
3	Software Engineering I	---

Objectives:

The goal of this course is to familiarize students with foundations of human-computer interaction (HCI), complexity of humans and their attitude to digital instruments, computer and digital tools and accessories, different stages of simple and reliable software design based on intrinsic perception of humans, and management issues at different stages of software design. The course also discusses future of HCI and its impact on software development, and HCI concepts and issues.

Topics:

- Human-Computer Interaction (HCI) concepts
- Modeling at HCI
- Framework concepts
- Designing interactive frameworks
- Design language
- Design evaluation (assessment)
- Designing details
- Methods to improve design details

References:

[1] B. Shneiderman, C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* 4th Edition, Pearson Education, 2005.

3-3-3 Robot Building Workshop (CE373)

Robot Building Workshop (CE373)		
Units	Prerequisites	Corequisites
3	Introduction to Robotics	---

Objectives:

This workshop is designed to familiarize the students with different steps of building a robot, including the mechanical and electrical parts. In addition to building the robot components, the students learn the assembling and testing of the built system. The testing phase includes programming and evaluating various components of the system, including the operation of different robot sensors and drives.

Topics:

- Familiarization with technical documents of electrical and mechanical components
- Capabilities and constraints of metal shaping machines
- Product assembly and test steps
- Selecting electronic components and building printed circuit boards
- Electric AC and DC motor drives and their control via microcontrollers
- Capabilities of microcontrollers and their programming
- Step by step testing and fault detection of a mechatronic system

References:

[1] *C Programming Reference Manual*

[2] *Introduction to Micro AVR*

[3] *Lab manual*

3-3-4 Computer Games Design (CE374)

Computer Games Design (CE374)		
Units	Prerequisites	Corequisites
3	Advanced Programming	---

Objectives:

The objective of this course is to introduce the principles of computer game design and development. This course covers these topics: Process and pipeline of game development, game engines, fundamental aspect of computer games, mathematical and physical sciences of computer games.

Topics:

- Analysis of the structure of computer games
- Introduction to the game development pipeline
- Architecture of game engines
- Mathematical foundations of computer games
- Introduction to Engine Support System
- Introduction to Game Loop
- Programming and development of game
- Artificial intelligence role in games
- Game development middleware

References:

[1] J. Gregory, *Game Engine Architecture*, A K Peters LTD, 2009.

[2] E. Lengyel, *Game Engine series*, CRC Press, 2007-2010

[3] D. H. Eberly, *3D Game Engine Design*, 2nd Edition, Morgan Kaufman, 2006

3-3-5 Theory of Computation (CE375)

Theory of Computation (CE375)		
Units	Prerequisites	Corequisites
3	Theory of Languages and Machines	---

Objectives:

The goal of this course is to familiarize students with the foundations of theory of computation and fundamental concepts of computability models, unsolvable problems, mathematical logics and theory of automata on finite string or tree as input. This course provides necessary theoretical foundations for graduate students who are interested in theory of computation and algorithms, formal methods in software engineering, verification of systems and artificial intelligence.

Topics:

- Computability Theory and Computation Complexity
  - Turing computable Model, Church-Turing Thesis, Functions and Decidable Languages, Functions and Recognizable Languages, Uncomputable Functions, Halting Problem, Universal Turing Machine, Multi-tape Turing Machine, Non-deterministic Machine and Theorems for Their Equivalency
  - Proof Methods for Undecidability and Unreconizabilty of Languages (Method for Reduction to Halting Problem and Functional Reduction Method)
  - Introduction to other Computational Models (von Neumann Random Access Model, Kleene's Recursive Theory, Lambda-Church Calculus, Post's Correspondence Problem)
  - Recursive Theorem and Self-Reference
  - Information-Theoretic Computational Complexity and String Complexity
  - Introduction to Complexity Theory, Review of Classes of Time and Storage Complexity Classes and Hard Problems
- Mathematical Logic from Computation Theory Point of View
  - Propositional Logic and its Syntax and Semantic, Axiomatic Inference System and its Soundness and Completeness Theorems, Undecidable Theorems for Propositional Logic
  - First Order Logic and its Syntax and Semantic, Compression and Löwenheim–Skolem Theorems
  - Axiomatic Inference System of First Order Logic and its Soundness Theorem
  - Godel's Completeness Theorem for First Order Logic Inference System
  - Church's Theorem in Undecidability of First Order Logic
  - Axiomatic System for Theory of Numbers and Godel's Uncompleteness Theorem
- An Introduction on Theory of Automata on Infinite inputs
  - Buchi and Rabin Automata on Finite Strings

- Theorems for Complementation and Empty Test for Language of Buchi Automata, Deterministic Buchi Automata, Safra's Theorem
- An Introduction to Relationship of Decidable Problems in Logic with Theory of Automata
- An Introduction to Automata on Tree as input

References:

[1] M. Divis, R. Sigal, E. Weyuker, *Computability, Complexity, and Languages*. 2nd Edition, Academic Press, 1997.

[2] M. Sipser, *Introduction to the Theory of Computation*. 2nd Edition, Thompson, 2006.

3-3-6 Computer Simulation (CE376)

Computer Simulation (CE376)		
Units	Prerequisites	Corequisites
3	Operating Systems, Engineering Probability and Statistics	---

Objectives:

This course covers the concepts and principles of computer simulation. A system behavior is implemented in a required level of abstraction in a virtual environment in simulation. This provides a flexible and low cost approach to study the behavior of systems. Application of computer simulation covers a wide range from educational purposes to evaluating system design and tuning. In this course, students are acquainted with simulator design and implementation using programming languages and also learn to use an existing simulation package. The statistical techniques used in simulations will also be covered including input-output modeling and simulation validation. Variance reduction techniques and rare event simulation are introduced as well.

Topics:

- Basic concepts and ten main steps in a simulation study
- Basics of discrete event simulation (DES)
- Examples of software simulation
- A review of simulation packages
- Introducing a simulation package
- Statistical models in simulation
- Validating a simulation model
- Analysis of simulation outputs
- Evaluation and optimization of alternative system designs
- Case studies: computer systems and networks
- Variance reduction techniques and rare event simulation

References:

[1] J. Banks, J. S. Carson, B. L. Nelson, D. M. Nicol, *Discrete-Event System Simulation*, 5th Edition, Pearson Publishing, 2010.

3-3-7 Foundations of Computer Animation (CE377)

Foundations of Computer Animation (CE377)		
Units	Prerequisites	Corequisites
3	Computer Graphics	---

Objectives:

The objective of this course is introducing the basic rules and foundations of animation which is the indispensable part of computer games. Introduction to principles of generating, processing and displaying animations are the main section of the course. Along the theoretical aspects, students learn to combining, processing and displaying animations in practice.

Topics:

- History of computer animation
- Hierarchical rigid body animation
- Skeleton meshes
- Vertex based animation
- Kinematics
- Blending
- Layering
- Bone placement
- Mixing
- Natural motions
- Pipeline

References:

[1] C. Grenberg, *Character Animation with DirectX*, Charles River Media, 2009.

[2] J. Gregory, *Game Engine Architecture*, A K Peters LTD, 2009.

3-3-8 Information Technology Project Management (CE378)

Information Technology Project Management (CE378)		
Units	Prerequisites	Corequisites
3	---	---

Objectives:

The objective of this course is to provide a foundation to prepare students, as future IT project managers, IT engineers, or system architects, to play leading roles in the application and management of e-business system construction. Upon successful completion of the course, students will be able to:

- Understand the job roles of an IT project manager;
- Develop Work Breakdown Structures (WBS)
- Manage overall change control
- Control project execution processes
- Build up the baseline knowledge for further career in IT project management fields

Topics:

- Introduction to Project Management
- The Project Management and Information Technology Context
- The Project Management Process Groups: A Case Study
- Project Integration Management
- Project Scope Management
- Project Time Management
- Project Cost Management
- Project Quality Management
- Project Human Resource Management
- Project Communications Management
- Project Risk Management
- Project Procurement Management
- Project Stakeholder Management
- Guide to Using Microsoft Project

References:

[1] K. Schwalbe, *Information Technology Project Management*. 7th Edition, Course Technology, 2014.

[2] J. T. Marchewka, *Information Technology Project Management*, 4th Edition, Wiley, 2012.

3-3-9 E-Commerce (CE379)

E-Commerce (CE379)		
Units	Prerequisites	Corequisites
3	Engineering Economics, Computer Networks	---

Objectives:

In this course, students are introduced to the basic concepts of e-commerce. They will learn about the application of information technology in e-commerce. Moreover, in this course, the concepts and techniques of internet marking is discussed.

Topics:

- Introduction to E-commerce
- Knowledge based business
- Value in network economy
- Virtual enterprise and factory
- Product development in digital economy
- Marketing in digital economy
- Product management and trading services
- Strategic planning and trading process
- Security and E-commerce
- E-commerce infrastructure
- E-commerce software
- Search strategies
- Application of software agents in commerce

References:

[1] G. W. Treese, and L.C. Stewart, *Designing System for Internet Commerce*. 1st Edition, Addison Wesley, 1998.

[2] D. Coyle, *The Weightless World: Strategies for Managing the Digital Economy*. The MIT Press, 1998.

[3] A. Chaudhury, J.P. Kuilboer, *E-Business & E-Commerce Infrastructure: Technologies Supporting the E-Business Initiative*. McGraw-Hill, 2011.

3-3-10 Startup Business Development (CE380)

Startup Business Development (CE380)		
Units	Prerequisites	Corequisites
3	Software Engineering I	---

Objectives:

The purpose of this course is introducing new methods, tools, and concepts in business development especially startups. We tried to outline starting a personal business for students by presenting practical concepts and internal and external examples and increase success rate of such businesses by teaching some of the methods and sharing experiences in main centers of technology growth. In addition to subjects, entrepreneurs or successful managers will be invited in some sessions not only to teach some of the subjects or related topics but to answer students' questions. During this course, students start developing a business plan for a startup practically and in groups.

Topics:

- Introduction
- Startup businesses
- Risk evaluation
- Resource evaluation
- Market analysis
- Financial analysis
- Proposed value
- Team work skills
- Internal and international business regulations
- Software development methods and skills
- Presentation of successful entrepreneurs in Iran and over the world
- Presentation of great successes and failures in launching startups
- Lectures from successful startup launchers

References:

[1] A. Osterwalder, Y. Pigneur, *Business Model Generation*, John Wiley, 2010.

[2] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, 2011.

[3] S. Blank, S. G. Blank, B. Dorf, *The Startup Owner's Manual: The Step-by-step Guide for Building a Great Company*, K&S Ranch, 2012.

[4] A. Maurya, *Running Lean: Iterate from Plan A to a Plan That Works*, O'Reilly, 2012.

3-3-11 Industrial Automation (CE381)

Industrial Automation (CE381)		
Units	Prerequisites	Corequisites
3	Microprocessor and Assembly Language	---

Objectives:

The objectives of this course is to provide students with the concepts of process control, different kinds of industrial controllers, their components and capabilities, sensors, actuators, high definition media interfaces, communications and networks used for transfer of industrial data.

Topics:

- Introduction and basic concepts
- Process control and its variants
- PID controllers
- Programmable Logic Controllers (PLC) for automation and process control
- Direct Digital Control (DDC)
- Supervised Control and Data Acquisition (SCADA)
- Distributed Control Systems (DCS)
- Human and Machine Interface in Industrial Automation systems (HMI)
- Programming methods for Industrial Automation Planning Design Method
- Sensors: measuring terms, pressure, temperature, flow, proximity sensors, capacitive, inductive, resistive, magnetic and optical sensors
- Relays and actuators
- Data communication, protocols, and industrial networks

References:

[1] J. Stenerson, *Industrial Automation and Process Control*, Prentice Hall, 2002.

[2] T. L. M. Bartelt, *Industrial Automated System: Instrumentation and Motion Control*, Delmar, Cengage Learning, 2010.

[3] M. Madhuchhanda, G. S. Sen, *Programmable Logic Controllers and Industrial Automation*, 2005.

[4] R. Shell, *Handbook of Industrial Automation*, Taylor & Francis, 2000.

[5] J. A. Rehg, G. J. Sartori, *Programmable Logic Controllers*, Prentice Hall Higher Education, 2009.

[6] G. Michel, *Programmable logic controllers: architecture and application*, Wiley, 1990.

[7] M. P. Lukas, *Distributed control systems: their evaluation and design*, Van Nostrand Reinhold Co., 1986.

[8] C. Gerber, *Implementation and Verification of Distributed Control System*, 2011

[9] S. A. Boyer, *Scada: Supervisory Control and Data Acquisition*, International Society of Automation, 2010.

[10] M. S. Nardone, *Direct Digital Control Systems: Application Commissioning*, Springer, 1999.