Shiraz University
Computer Science and Engineering
Course Catalog

(Updated on 13 August , 2022)

Principles of Cryptography

Secure Programming

Laws in Security

Software Hardware Co-Design

Embedded System Design

Very Large Scale Integrated Circuits (VLSI)

Architecture of Kernel Accelerators

Principles of Computational Intelligence

Principles of Computer Vision

Principles of Natural Language Processing

Principles of Robotics

Human Computer Interaction

Software Testing

Formal Methods in Software Engineering

Object-Oriented Design

Database lab

Foundations of Operations Research

Mobile Programming

GPU Programming

User interface design lab

Linear Algebra

Green Computing

Randomized Algorithms

Configurable Architectures

Evolutionary Computing

Statistical Pattern Detection

Fuzzy Systems

Advanced Database Systems

Optimization Algorithms

Web Programming

Parallel Algorithms

Data Mining

Modeling of Biological Systems

Intro to Computational Biology and Bioinformatics

Text Mining

Information Networks

Advanced Engineering Mathematics

Business Intelligence

E-Commerce Security

Knowledge Management Systems

Scientific computations and data analysis

Computer Graphics

Information Technology Project Management

Special Topics In Software Engineering (Multi-Core and GPU programing)

Quick Programming Challenges

Information Storage and Retrieval

Calculus 1

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0

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

Suggested Textbooks:

1. Stewart, James. Calculus: Concepts and contexts. Cengage Learning, 2015.

Calculus 2

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Calculus 1

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

Suggested Textbooks:

1. Stewart, James. Calculus: Concepts and contexts. Cengage Learning, 2015.

Physics 1

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0

Description:
Measurement, vectors, one-dimensional motion, motion in one plane, dynamics of particle, work and energy, energy preservation, dynamics of particles systems, rotary kinematics, balance of solid bodies, temperature fluctuation, heat, 1st thermodynamics law, gas kinetic theory, and 2nd thermodynamics law.

Suggested Textbooks:

1. Halliday, David, Robert Resnick, and Jearl Walker. Principles of physics. John Wiley & Sons, New Jersey, 2020.

Physics 2

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Physics 1 (Heat and Mechanics), Calculus 1

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

Suggested Textbooks:

1. Halliday, David, Robert Resnick, and Jearl Walker. Principles of physics. John Wiley & Sons, New Jersey, 2020.

Engineering Probability and Statistics

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Calculus 2

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

Suggested Textbooks:

1. Papoulis, A., and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. 4th Edition, McGraw Hill, 2002 (Chapters 1 through 8).
2. Ross, Sheldon M. *A First Course in Probability*. 10th Edition, Prentice Hall, 2019.

Differential Equations

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Calculus 2

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

Suggested Textbooks:

1. Boyce, William E., Richard C. DiPrima, and Douglas B. Meade. *Elementary differential equations and boundary value problems*. 9th edition, 2008.

Computer Workshop

Credit Hours: 1.0
Lecture Contact Hours: 0 Lab Contact Hours: 1
Prerequisite: Fundamentals of Computer Programming

Description:
History, kinds and applications of computer including personal computer, working station, minicomputers, big and super-computers, structures and accessories including motherboard, output and input board, keyboard, screen, printer, scanner, platter, modem, series and parallel gates, secondary memories, introduction to media DOS, windows 95, windows NT, editors such as vi and edit, introduction to internet including mail, ftp, Telnet, web, introduction to some applied software such as Word, Latex, Excel and Corel.

Suggested Textbooks:

1. C. Newman, *SAMS Teach Yourself PHP in 10 Minutes*. Sams Publishing, 2005.
2. D. Hayes, *Sams Teach Yourself HTML in 10 Minutes*. 4th Edition, Sams publishing, 2006.
3. R. Weakley, *Sams Teach Yourself CSS in 10 Minutes*. Sams Publishing, 2005.
4. B. Forta, *Sams Teach Yourself Regular Expressions in 10 Minutes*. Sams Publishing, 2004.
5. R. Shimonski, *SAMS Teach Yourself Unix in 10 Minutes*. Sams Publishing, 2005.
6. J. Andrews, *A+ Guide to Managing & Maintaining Your PC*. 7th Edition, Course Technology, 2009.
7. Cisco Networking Academy, *IT Essentials PC Hardware and Software Course Booklet*. Version 4.1, 2nd Edition, Cisco Press, 2010.

Physics Lab

Credit Hours: 1.0
Lecture Contact Hours: 0   Lab Contact Hours: 1
Prerequisite: Physics 2

Description:
According to the syllabus presented in physics II.

Suggested Textbooks:

Fundamentals of Computer Programming

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0

Description:
problem solving, algorithm presentation using pseudocode, introduction to an organized programming language, constants, variables, computational and logical phrases, types of instructions, types of loops, conditional operations, vectors and matrices, subprograms (functions and procedures), input and output instructions, common algorithm such as methods of search and sort

Suggested Textbooks:

1. Downey, Allen, et al. *How to think like a computer scientist: learning with python 3*. 2016.
2. Wentworth, Peter, Jeffrey Elkner, Allen B. Downey, and Chris Meyers. *How to Think Like a Computer Scientist*. 2011.

Assembly and Machine Language

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Description:

- Numerical systems: representation of negative numbers, characteristics of binary complementary systems, representation of BCD numbers, representation of floating-point numbers.
- Assembly language: methods of addressing (implicit, real time, direct, indirect, benchmarking, base, shifting against contents of counter program, paging program), instructions of work with registers, instructions of work with machine memory (work with words and their combinations, work with characters and bits), instructions related to distribution and loop control, logic instructions, retrieval and transfer of routines, regressive type, assembler facilities at time of translation including macros, iteration and conditional blocks, dump assessment, application of assembly instruction in high-level programming languages, connection of high level programs and assembly language.

Suggested Textbooks:

1. Irvine, Kip R., and Lyla B. Das. *Assembly language for x86 processors*. 2014.
2. Kusswurm, Daniel. *Modern X86 Assembly Language Programming*. Springer, 2018.

Numerical Analysis

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Description:
Programming numerical solution of problems in linear algebra; system of linear equations, matrix inversion, and Eigenvalue problems, solution of equations, polynomial approximations, and initial value problems of ordinary differential equations.

Suggested Textbooks:

1. Burden, Richard L., J. Douglas Faires, and Annette M. Burden. *Numerical analysis. 10th Ed., Cengage learning*, 2015.
2. Young, Todd, and Martin J. Mohlenkamp. *Introduction to numerical methods and Matlab Programming for Engineers*. Department of Mathematics Ohio University, 2021.

Principles of Computer Programming

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0

Description:
More advanced problem solving, algorithm presentation using pseudocode, system level introduction to an organized programming language, constants, variables, computational and logical phrases, types of instructions, types of loops, conditional operations, vectors and matrices, subprograms (functions and procedures), input and output instructions, common algorithm such as methods of search and sort. Pointer and pointer manipulations. Basics of data structures.

Suggested Textbooks:
1. Ritchie, Dennis M., Brian W. Kernighan. *The C Programming Language*. 2nd Edition, Prentice Hall, 1988.
2. Deitel, Harvey M., Paul J. Deitel. *C: How to Program*. 8th Edition, Prentice-Hall, 2016.

Electrical Circuits

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Differential Equations

Description:
Compact circuits & Kirchhoff's Laws, approximation & modeling of circuit elements including: registers, unclosed and reclosed resources (voltage & current) capacitors, selves, power, energy, operational amplifiers (OPAMP) as a circuit element, simple circuits such as: resistor circuits, analytical methods of resistor circuits, labeling two terminals of a circuit, Tonen, Norton equivalent circuit & commutative theorem in resistor circuits, resources conversion, arranging selves, capacitors, application of spice in solving resistor circuits, first order circuits including RL & RC circuits, zero input responses, response at zero state, complete, transient and permanent responses, time coefficients and circuits with several time coefficients, switching, plateau & impulse responses, 2^{nd} order circuits, stability, oscillation negative resistance concepts, double circuits, similarity of electrical & mechanical systems.
Application of spice in solving 1^{st} and 2^{nd} logic circuits and OPAMP, analytical methods for linear circuits (network and node analysis(, importance of impulse response and estimation in general linear circuits (time domain analysis) & convolution theorem, permanent sinusoidal state analysis including: concepts of phasor and impedance, admittance, phasor diagram, concepts of resonance and series & parallel resonance circuits, network functions frequency responses, power at permanent sinusoidal state, average, real and reactive power, maximum power transfer theorem, effective values & RMS, scale change in a circuit, application of spice in solving permanent sinusoidal circuits, tri-phase circuit analysis-conjugated circuits including conjugated selves, circuits equivalent of T, π conjugated selves, inductance matrix, connecting conjugated selves, transformers, circuit models & their applications, application of spice in solving selves administrated circuits and transformers

Suggested Textbooks:

1. Desoer, C. A., and E. S. Kuh. Basic Circuit Theory. 16th printing Singapore, 1987, translated in Farsi by Parviz Jabehdar Maralani. Published by Tehran University Press, *34th Ed, 2021*.
2. Desoer, C. A., and E. S. Kuh. Basic Circuit Theory. 16th printing Singapore, 1987.
3. Chua, Leon O., Charles A. Desoer, and Ernest S. Kuh. *Linear and nonlinear circuits*. McGraw-Hill College, 1987.

Discrete Mathematics (**Discrete Structures**)

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Description:
-Introduction: mathematical logic, algebra of expressions, well-structured formula, a review of theory of sets, proof methods.
-Relations and functions: binary relations, compatibility and equivalence relations, relations representation matrix, graphs of relations, functions, coverage functions, one to one functions.
-Recursive functions, deduction, generator functions.
-Algebraic structures: semi-groups and monoids, grammars and languages, Polish marking, groups, homomorphism, isomorphism, lattices, Boolean Algebra, Carnot table, grammar and language, grammar as an example of monoids
-Combinatorial analysis, nest principles, introduction to combinatorial algorithm, regressive and reciprocal functions and applications.
-Graph theory: directional graphs, unidirectional graphs, Eulerian and Hamiltonian paths, optimal paths, finding algorithm for optimal paths, connected graphs, matrix of relation and related theorems, application of graphs in activity analysis.
-Trees: minimal overlapping trees, surveying of trees, application of trees, algebraic expressions and representation of their trees.

Suggested Textbooks:

1. Rosen, Kenneth H. *Discrete Mathematics and Its Applications*. 8th Edition, McGraw Hill, 2018.
2. Grimaldi, Ralph P. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5th Edition, Pearson Addison Wesley, 2004.
3. Engel, Arthur. *Problem-Solving Strategies*. Springer, 1998.

Advanced Programming

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Fundamentals of Computer Programming
Description:
Basics of object-oriented design and analysis. Students are given a simple real-world problem and principles of quality design and analysis is taught by examples. The students are given variety of programming assignments to practice the fundamentals taught at the class.

Suggested Textbooks:

1. Eckel, Bruce. *Thinking in Java*. 4th Edition, Prentice Hall, 2006.
2. Fowler, Martin, K. Beck, J. Brant, W. Opdyke, D. Roberts. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999.
3. Deitel, H. M., and P. J. Deitel. *Java: How to Program*. 11th Edition, Pearson Education, 2017.

Data Structures

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Advanced Programming, Engineering Mathematics

Description:
Introduction to various data structures (stacks, queues, lists, hash tables, trees, heaps, and graphs); sorting and searching; design, analysis, and comparison of algorithms.

Suggested Textbooks:

1. Leiserson, Charles Eric, Ronald L. Rivest, Thomas H. Cormen, and Clifford Stein. *Introduction to Algorithms*. 3rd Edition, MIT Press, 2011.

Digital Circuits (**Logic Design**)

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: [Data Structures]

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

Suggested Textbooks:

1. Mano, M. Morris. *Digital Design*. 5th Edition, Prentice Hall, 2006.
2. Nelson, Victor P., H. Troy Nagle, Bill D. Carroll, and J. David Irwin. *Digital Logic Circuit Analysis and Design*. Prentice Hall, 1995.
3. Prosser, Franklin P., and David E. Winkel. *The Art of Digital Design: An introduction to top-down design*. Prentice Hall, 1987.

Theory of Computation[1]

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Data Structures

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

Suggested Textbooks:

1. Sipser, Michael. *Introduction to the Theory of Computation*. 3rd Edition, Cengage Learning, 2013.
2. Linz, Peter. *An introduction to formal languages and automata*. Jones & Bartlett Learning, 2006.
3. Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. 2nd Edition, Addison-Wesley, 2001.
4. Denning, Peter J., Jack B. Dennis, and Joseph E. Qualitz. *Machines, languages, and computation*. Prentice-Hall, 1978.
5. Cameron, Peter J. *Sets, logic and categories*. Springer Science & Business Media, 2012.

[1] Aka, Theory of Automata and Formal Languages

English for Computer Engineers(**English for Computing**)

Credit Hours: 2.0
Lecture Contact Hours: 2 Lab Contact Hours: 0
Prerequisite: General English

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

Suggested Textbooks:

1. Fabré, Elena Marco, and Santiago Remacha Esteras. Cambridge University Press, 2007. (Main Textbook)
2. Glendinning, Eric H., John McEwan. *Oxford English for IT*. Oxford University Press, 2006.

Technical Writing and Presentation (Scientific and Technical Presentation)

Credit Hours: 2.0
Lecture Contact Hours: 2 Lab Contact Hours: 0
Prerequisite: English for Computer Engineers(English for Computing)

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

Suggested Textbooks:
1. Day, Robert A., and Barbara Gastel. *How to Write and Publish a Scientific Paper*. 8th Edition, Cambridge University Press, 2016.
2. Alley, Michael. *The Craft of Scientific Presentations Critical Steps to Succeed and Critical Errors to Avoid*. 2nd Edition, Springer, 2013.

Engineering Mathematics

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Calculus 2, Differential Equations

Description:

Fourier series and its integral and Fourier transform: definition of Fourier series, Euler's formula, expansion in half range, rebound oscillations, Fourier integral.

Partial derivatives equations: vibratory string, univariate wave equation, method of decomposition of variables, D'Alembert solution of the wave equations, heat propagation equation, univariate wave equation, equation in Cartesian, spherical and polar coordinates, elliptical, parabolic and hyperbolic equations, application of Laplace transformations in solving equations with partial derivatives, solution of partial derivatives using Fourier integrals.

Analytical functions and conformal mapping and complex integrals: limit and continuity, derivative of complex functions, exponential, trigonometric hyperbolic and logarithmic functions, inverse trigonometric and exponential functions with different power, conformal preposition-mapping.

Integral of line on complex plane, Cauchy's integral theorem, calculation of line integral using indefinite integral, Cauchy' formula, Taylor and Mac-Lauren' expansions, integration using residual methods, residuals theorems, calculation of certain real integrals.

Suggested Textbooks:

1. Kreyszig, Erwin. *Advanced engineering mathematics*, 10th edition, 2009.

Computer Architecture

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Digital Circuits(Logic Design)

Description:
Logic design, principles of operation of digital computers, and analysis of major components: arithmetic processing, memory, control and input/output units, instruction pipelining, SIMD and multiprocessor systems.

Suggested Textbooks:

1. Patterson, David A., and John L. Hennessy. *Computer organization and design*. 3rd Edition, Elsevier (Morgan Kaufmann), 2005.
2. Mano, M. Morris. *Computer system architecture*. Prentice-Hall, Inc., 1993.

Algorithm Design

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Data Structures

Description:
Algorithm analysis and design, heuristics; advanced tree structures; advanced hashing techniques; sorting and searching; graphs, sets. NP-Completeness, Time and Space complexities.

Suggested Textbooks:

1. Kleinberg, Jon, and Eva Tardos. *Algorithm Design*. Addison Wesley, 2005.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. 3rd Edition, MIT Press, 2009.
3. Manber, Udi. *Introduction to algorithms: a creative approach*. Addison-Wesley Longman Publishing Co., Inc., 1989.
4. Brassard, Gilles, and Paul Bratley. *Algorithmics: theory & practice*. Prentice-Hall, Inc., 1988.

Computer Aided Design[2]

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Computer Architecture

Description:
Applications of hardware description languages (HDLs) for the design of complex digital systems. Topics include designing and simulating using HDLs, logic synthesis into FPGAs and ASICs, optimization techniques, timing issues, hardware verification, and design for testability.

Suggested Textbooks:

1. Bobda, Christophe, and Reiner Hartenstein. *Introduction to reconfigurable computing: architectures, algorithms, and applications*. Vol. 1. No. 1.5. Netherlands: Springer, 2007.
2. Palnitkar, Samir. *Verilog HDL: A Guide to Digital Design and Synthesis*. 2nd Edition, SunSoft Press, 2003.
3. Brown, Stephen, and Jonathan Rose. *FPGA and CPLD Architectures: A Tutorial*. IEEE Design and Test of Computers, pp. 42-57, 1996.
4. *Altera Data Sheets*. available at www.altera.com.
5. *Xilinx Data Sheets*. available at www.xilinx.com.
6. *Actel Data Sheets*. available at www.actel.com.

[2] Also known as: Design of Digital Systems

Microprocessors and Assembly Language

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Computer Architecture

Description:
Reviewing history, types and developing trend of microprocessors, organization of a microprocessor and method of executing instructions, introduction to the architecture of 8085 bit processor and Z80 and their differences, introduction to programming methods, addressing modes, calculating timing of processing and designing system relying on microprocessors such as clock pulse producing circuit, types of EPROM and RAM memories, decoders circuits, output and input, surveying 8086 microprocessor and properties of 16 bites microprocessors, system supporting chips including timer 8254 and 8255 parallel bus, 8251 USART bus, interrupt control.

Suggested Textbooks:

1. Triebel, Walter A., and Avtar Singh. *The 8088 and 8086 Microprocessors*. Prentice-Hall, 2003.
2. Mazidi, Muhammad Ali, and Janice Gillispie Mazidi. 80x86 IBM PC and Compatible Computers, The, 2000.
3. Mazidi, Muhammad Ali, et al. *The AVR Microcontroller and Embedded Systems: Using Assembly and C*. Prentice Hall, 2011.
4. Kirk, David B., and W. Hwu Wen-Mei. *Programming massively parallel processors: a hands-on approach*. Morgan kaufmann, 2016.

Computer Networks

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Operating Systems

Description:
Data communications; network protocols and architecture; local and wide-area networks; internetworking.

Suggested Textbooks:

1. Peterson, Larry L., and Bruce S. Davie. *Computer Networks: A Systems Approach*. 5th Edition, 2011.

Artificial Intelligence and Expert Systems

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Data Structures

Description:
Topics include search techniques, reasoning with logic, planning, decision making, machine learning.

Suggested Textbooks:

1. Russell, Stuart, and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd Edition, 2009.

Principles of Compiler Design

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Data Structures

Description:

Preliminaries (various types of translators including compilers and interpreters), structure and components of compilers (tokenizers, syntactic analyzer, semantic analyzer, symbols table management, middle and final code generation, optimization, errors management).

Parsers: LR and LL(1), SLR(1), LALR(1) and similar parsing algorithms. Syntax errors.

Symbol table management and their structure. Memory allocation techniques (static and dynamic). Code generation. Translation of several structures of imperative languages such as statements, control structures, routines call.

A brief discussion of code optimization.

Suggested Textbooks:

1. Aho, Alfred V., Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. 2nd Edition, Addison Wesley, 2007.
2. Grune, Dick, Kees Van Reeuwijk, Henri E. Bal, Ceriel JH Jacobs, and Koen Langendoen. *Modern Compiler Design*. Springer Science & Business Media, 2012.
3. Tremblay, Jean-Paul, and Paul G. Sorenson. *Theory and Practice of Compiler Writing*. McGraw-Hill, Inc., 1985.
4. Fischer, Charles N., and Richard J. LeBlanc Jr. *Crafting a Compiler with C*. Benjamin-Cummings Publishing Co., Inc., 1991.

Operating Systems

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: Data Structures, Computer Architecture

Description:
Operating system; sequential processes, concurrent processes, deadlock, mutual exclusion, semaphores; memory management, processor management, peripheral device management.

Suggested Textbooks:

1. Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. *Operating System Concepts*. 10th Edition, Wiley Publishing, 2018.
2. Anderson, Thomas, and Mike Dahlin. *Operating Systems: Principles and Practice, volume 1: Kernel and Processes*. Recursive books, 2014.

Operating Systems Lab

Credit Hours: 1.0
Lecture Contact Hours: 0 Lab Contact Hours: 1
Prerequisite: Operating Systems

Description: Hands-on exercises based on topics covered in the Operating Systems course

Suggested Textbooks:

1. Salzman, Peter Jay, Michael Burian, and Ori Pomerantz. *The linux kernel module programming guide*. 2007.
2. Wall, Kurt, Mark Whitis, and Mark Watson. *Linux Programming Unleashed, Macmillan Computer Publishing*. 1999.
3. Mitchell, Mark, Jeffrey Oldham, and Alex Samuel. *Advanced linux programming*. New Riders Publishing, 2001.
4. Rodriguez, Claudia Salzberg, and Gordon Fischer. *The Linux® Kernel Primer: A Top-Down Approach for x86 and PowerPC Architectures*. Prentice-Hall, 2005.
5. Corbet, Jonathan, Alessandro Rubini, and Greg Kroah-Hartman. *Linux device drivers*. O'Reilly Media, Inc, 2005.

Digital Circuits and Computer Architecture Lab

Credit Hours: 1.0
Lecture Contact Hours: 3   Lab Contact Hours: 1
Prerequisite: Digital Circuits(Logic Design), Computer Architecture

Description:
Hands-on exercises based on topics covered in the courses.

Suggested Textbooks:

1. Patterson, David A., and John L. Hennessy. *Computer organization and design ARM edition: the hardware software interface*. Morgan kaufmann, 2016.
2. Mano, M. Morris. *Computer system architecture*. Prentice-Hall, Inc., 1993.

Microprocessors Lab

Credit Hours: 1.0
Lecture Contact Hours: 0 Lab Contact Hours: 1
Prerequisite: Microprocessors and Assembly Language

Description:

Hands-on exercises based on topics covered in the Microprocessors course

Suggested Textbooks:

Computer Networks Lab

Credit Hours: 1.0
Lecture Contact Hours: 0 Lab Contact Hours: 1
Prerequisite: Computer Networks

Description:
Hands-on exercises based on topics covered in the course.

Suggested Textbooks:

1. Kurose, James, and Keith Ross. *Computer Networking: A Top-Down Approach*. 7th Edition, Pearson, 2016.
2. Peterson, Larry L., and Bruce S. Davie. *Computer Networks: A Systems Approach*. 5th Edition, 2011.
3. Tanenbaum, Andrew S. *Computer Networks*. 5th Edition, Pearson, 2010.

Electronic Circuits

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Electrical Circuits

Description:

Characteristics of ideal diode, π type and P type semiconductor, linkage between Pπ, electrical characteristics of actual diode, diode rectifier circuits, diode jumping and chopper circuits, Zener' diode specifications, Zener voltage modulator.

BJT bipolar transistor, structure and electrical behavior, biasing of BJT (dc analysis), small signal model, BJT amplifier, CB, CE, CC amplifier (small signal analysis), multi layers amplifiers and magnitude.
Field effect transistor, structure and electrical behavior, introducing CMOS, biasing FET, FET amplifiers.

Suggested Textbooks:

Digital Electronics

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisites: Electronic Circuits , Electrical Circuits

Description:

Digital electronics are circuits dealing with binary values in which “true/false” logical operations are used. This stands in contrast to analogue circuits, in which signals are free to vary continuously between the limits imposed by power supply voltage and circuit resistances. These circuits find use in binary logical operations and digital computation. Basic Boolean functions such as AND, OR, and NOT are essential in building functionality for digital systems, and there might ten billion or more of these gates in the modern integrated circuits (ICs).

These Boolean functions might be implemented in different designs of electronic elements. This course provides complete qualitative descriptions of digital circuit operation followed by in-depth analytical analyses of utilized elements and designs. The circuit families described in detail are diode logic (DL), resistor-transistor logic (RTL), transistor-transistor logic (TTL, STTL, and ASTTL), emitter-coupled logic (ECL), NMOS logic, CMOS logic, dynamic CMOS, BiCMOS structures and various GASFET technologies. In addition to detailed presentation of the basic physics, inverter circuits for each digital logic family and complete details of some other logic circuits for these families are presented.

Suggested Textbooks:

1. DeMassa, Thomas, and Zack Ciccone. Digital Integrated Circuits. Wiley, 1996.
2. Ayers, John. Digital Integrated Circuits: Analysis and Design, Second Edition. 2nd ed., CRC Press, 2009.

Data Transmission

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisites: Engineering Probability and Statistics, Signals and Systems

Description:

1. Layered network architecture
   1.1 Data Communication
2. Physical Layer Basics and Protocols
   2.1 Transmission Media
   2.2 PSTN & Mobile
   2.3 Signal Types
   2.4 Attenuation & Distortion
   2.5 MODEM
   2.6 Physical Layer Interface Standards

3. Data-Link Layer Fundamentals
   3.1 Transmission Modes
   3.2 Asynchronous Transmission
   3.3 Synchronous Transmission
   3.4 Error Control
   3.5 Compression

4. Data-Link Layer Protocols
   4.1 Idle RQ
   4.2 Continuous RQ

Suggested Textbooks:

1. Stallings, William. *Data and computer communications*. Pearson Education India, 2007.
2. Halsall, Fred. *Data Communications, Computer Networks, and Open Systems*. 4th Edition, Addison Wesley, 1996.
3. Tanenbaum, Andrew S. *Computer networks*. Pearson Education India, 2002.

Linear Control Systems

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Signals and Systems

Suggested Textbooks:

Electronic Circuits Lab

Credit Hours: 1.0
Lecture Contact Hours: 0 Lab Contact Hours: 1
Prerequisites: Electronic Circuits

Description:
Hands-on exercises based on topics covered in the Electronic Circuits course

Suggested Textbooks:

Digital Electronics Lab

Credit Hours: 1.0
Lecture Contact Hours: 0   Lab Contact Hours: 1
Prerequisites: Digital Electronics

Description:
Hands-on exercises based on topics covered in the Digital Electronics course

Suggested Textbooks:

1. DeMassa, Thomas, and Zack Ciccone. Digital Integrated Circuits. Wiley, 1996.
2. Ayers, John. Digital Integrated Circuits: Analysis and Design, Second Edition. 2nd ed., CRC Press, 2009.

Computer Aided Design Lab

Credit Hours: 1.0
Lecture Contact Hours: 0 Lab Contact Hours: 1
Prerequisites: Computer Aided Design

Description:

Hands-on exercises based on topics covered in the course.

Suggested Textbooks:

1. Palnitkar, Samir. *Verilog HDL: a guide to digital design and synthesis*. Vol. 1. Prentice Hall Professional, 2003.
2. ACEX 1K Programmable Logic Family Data Sheet. Available at www.altera.com.
3. ModelSim User's Manual. Available at www.actel.com.
4. Introduction to the Quartus II Software. Available at www.altera.com.

Signals and Systems

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Engineering Mathematics

Description:

Signals and system representation. Convolution and impulse response. Fourier series, Fourier transform, and Laplace transform. State variable analysis of continuous and discrete systems.

Suggested Textbooks:

1. Oppenheim, Alan V., et al. *Signals & systems*. Pearson Educación, 1997.

Undergraduate Project

Credit Hours: 3.0
Lecture Contact Hours: 0   Lab Contact Hours: 3

Description:
Developing a software system using all or part of the knowledge and materials that were gained through bachelor program based on personal interests.

Systems Analysis and Design

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Advanced Programming

Description:

Software development processes, requirement elicitation and tracing techniques, functional and non-functional requirements, modeling languages.

Suggested Textbooks:

1. Bentley, Lonnie D., Kevin C. Dittman, and Jeffrey L. Whitten. *Systems analysis and design methods*. Irwin/McGraw Hill, 2007.
2. Rubin, Kenneth S. *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.
3. M. Fowler, C. Kobryn, and K. Scott. *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.
4. Fowler, Martin. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.
5. Bass, Len, Ingo Weber, and Liming Zhu. *DevOps: A software architect's perspective*. Addison-Wesley Professional, 2015.
6. Fowler, Martin. *Patterns of Enterprise Application Architecture: Pattern Enterpr Applica Arch*. Addison-Wesley, 2012.

Database Systems (**Database Design**)

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Data Structures

Description:

Concepts and definitions of database management (definition of data and information, database definition and necessity, data independence, various models of database systems). Architecture of a database system (three-level architecture, external level, internal level, conceptual level, database and data administrator, data relationship administrator). Different models of database systems (Hierarchical model, Relational model, network model). Relational model of database (relations, tables, base tables and non-base tables, query language). Relational model elements (domain, relation and its types). Integrity of relational model (candidate key, primary key, foreign key and its rules, Null foreign key and primary key). Relational algebra. SQL language. Functional dependencies (partial dependencies definitions, a collection of dependencies, irreducible collection of dependencies). BCNF, 3NF, 2NF, 1NF normalization, multivalued dependency (MVD), 4NF, join dependency (JD), SNF.

Suggested Textbooks:

1. Elmasri, R., Shamkant B. Navathe. *Fundamentals of Database Systems*. 7th Edition, Pearson, 2015.
2. Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. *Database System Concepts*. 6th Edition, McGraw-Hill, 2010.
3. C. J. Date. *An Introduction to Database Systems* 8th Edition, Pearson, 2003.
4. Connolly, Thomas M., and Carolyn E. Begg. *Database Systems*. 6th Edition, Pearson, 2014.
5. Ramakrishnan, Raghu, Johannes Gehrke, and Johannes Gehrke. *Database Management Systems*. 4th Edition, McGraw-Hill, 2014.

Design of Programming Languages

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisites: Principles of Compiler Design

Description:
General characteristics of programming language, processors of programming language and their comparison. Specifications of data types and their implementation in high-level programs, methods of determining sequence of instruction execution and their implementation, data control, arguments of a function and its implementation, memory management methods in higher level languages, data abstraction.

Suggested Textbooks:

1. Friedman, Daniel P., Mitchell Wand, and Christopher Thomas Haynes. *Essentials of Programming Languages*. 3rd Edition, MIT Press, 2008.
2. Krishnamurthi, Shriram. *Programming Languages: Application and Interpretation*. 2nd Edition, 2017.
3. Felleisen, Matthias, et al. "A programmable programming language." *Communications of the ACM* 61.3 , 2018.

Software Engineering

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Analysis of Design of Software Systems(Systems Analysis and Design)

Description:
Introduction to the concepts of software engineering. Identification of problems related to the development of large software systems. Software project planning, requirements analysis, design, implementation, quality assurance and maintenance.

Suggested Textbooks:

1. pressman, Roger S. *Software Engineering: A Practitioner's Approach.* 8th Edition, McGraw-Hill, 2014.
2. Ammann, Paul, and Jeff Offutt. *Introduction to software testing*. Cambridge University Press, 2016.
3. Woodcock, Jim, and Jim Davies. *Using Z: Specification, Refinement, and Proof*. Prentice-Hall, 1996.

Internet Engineering

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Computer Networks

Suggested Textbooks:

Network Security

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Computer Networks

Suggested Textbooks:

1. Matt Bishop. *Computer Security*. Addison-Wesley, 2017.
2. John Erickson. *The Art of Exploitation* 2nd Edition, No Starch Press, 2008.
3. Robert C. Seacord. *Secure Coding in C and C++*. 2nd Edition, Pearson Education, 2005.
4. A. Sotirov. *Bypassing Browser Memory Protections*. 2008.
5. T. Garfinkel. *Traps and Pitfalls: Practical Problems in System Call Interposition Based Security Tools*. NDSS, 2003.
6. Adam Barth, Collin Jackson, and John C. Mitchell. *Securing Browser Frame Communication*. Usenix, 2008.
7. Adam Barth, Collin Jackson, Charles Reis, and the Google Chrome Team. *The Security Architecture of the Chromium Browser*. 2008.
8. Bortz et al. *Origin Cookies: Session Integrity for Web Applications*. 2011.
9. Enck, Ongtang, and McDaniel. *Understanding Android Security*. 2009.
10. Allan Tomlinson. *Introduction to the TPM: Smart Cards, Tokens, Security and Applications*. 2008.
11. Andrew Baumann, Marcus Peinado, and Galen Hunt. *Shielding Applications from an Untrusted Cloud with Haven*. OSDI 2014.

Principles of Secure Computing

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Operating Systems

Suggested Textbooks:

System Security

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisites: Database Systems, Operating Systems

Suggested Textbooks:

Management of Data Security

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisites: Principles of Secure Computing

Suggested Textbooks:

Principles of Cryptography

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Principles of Secure Computing

Suggested Textbooks:

Secure Programming

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Pre-requisites: Analysis and Design of Software Systems

Suggested Textbooks:

Laws in Security

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Pre-requisites: Network Security, System Security

Suggested Textbooks:

Software Hardware Co-Design

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Computer Architecture

Suggested Textbooks:

Embedded System Design

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Operating Systems, Microprocessors and Assembly Language

Description

This course is about the design and analysis of embedded systems. The course will cover the following topics: Introduction to real-time and embedded systems; specification languages & model of computations (StateChart, SDL, KPN, SDF, etc.), embedded systems hardware (Sensors, ADC information processing elements, DAC, actuators), embedded systems software and real-time operating systems (RTOS), analysis and verification; worst-case estimation, power-aware computing, reliability-aware design, application mapping, testing.

Suggested Textbooks:

1. P. Marwedel, *Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems*, 2nd Edition, Springer, 2011.

2. E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems – A Cyber-Physical System Approach*, UC Berkeley, 2011.

Very Large Scale Integrated Circuits (VLSI)

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Digital Electronics

Description:

This course is about the design and analysis of very large scale integrated circuits. The course will cover the following concepts: Introduction to manufacturing process, combinational logic design, stick diagram and layout design rules, resistors & capacitors, delay, power, memory cell design, test. In this course, it is assumed that the students are familiar with the basic concepts of digital electronics.

Suggested Textbooks:

1. N. H. E. Weste and D. Harris. *CMOS VLSI Design, A Circuits and Systems Perspective*. Course Handouts, 3rd Edition, Addison-Wesley, 2005.
2. J. M. Rabaey, A. Chandrakasan, and B. Nikolić. *Digital Integrated Circuits, A Design Perspective*. Course Handouts, 2005.
3. Wayne Wolf. *Modern VLSI Design, System-on-Chip Design*. 3rd Edition, Course Handouts, 2004.

Architecture of Kernel Accelerators

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Advanced Programming, Computer Architecture

Suggested Textbooks:

Principles of Computational Intelligence

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Advanced Programming

Introducing intelligent systems: artificial neural networks, deep learning, evolutionary computation, fuzzy systems, swarm intelligence, and hybrid techniques.

Suggested Textbooks:

1. A. P. Engelbrecht, Computational Intelligence: An Introduction, John Wiley & Sons, 2007.
2. A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, Springer Verlag, 2003.
3. M. Dorigo and T. Stutzle, Ant Colony Optimization, MIT Press, 2004.
4. J. Kennedy, R. C. Eberhart, and Y. Shi, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
5. L. N. de Castro, “Fundamentals of Natural Computing: An Overview”, Physics of Life Reviews, Vol. 4, No. 1, pp. 1-36, 2007.
6. L. N. de Castro, Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications, Chapman & Hall/CRC Computer and Information Sciences, 2006.

Principles of Computer Vision

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Principles of Computational Intelligence

Description:

A broad introduction to the foundations, algorithms, and applications of computer vision. Topics include filtering, feature detection, stereo vision, structure from motion, motion estimation, and recognition.

Suggested Textbooks:

Principles of Natural Language Processing

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisites: Signals and Systems, (Engineering) Probability and Statistics

Description:

N-gram language models, part-of-speech tagging, statistical parsing, word sense disambiguation, discourse segmentation, information extraction, sentiment analysis, machine translation.

Suggested Textbooks:

1. Dan Jurafsky and James H. Martin. Speech and Language Processing
2. Manning and Schuetze, Foundations of Statistical Natural Language Processing
3. Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning

Principles of Robotics

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Signals and Systems

Description:

Foundations of reasoning for robotics. Topics include state estimation, robot motion, perception, localization and decision-making.

Suggested Textbooks:

1. Computational Principles of Mobile Robotics. Gregory Dudek and Michael Jenkin. 2nd ed. Cambridge University Press, 2010
2. Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004
3. Autonomous Robots: From Biological Inspiration to Implementation and Control G.A. Bekey, MIT Press, 2005

Human Computer Interaction

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Analysis of Design of Software Systems

Description:

Theories of human-computer interaction and analyzes human factors related to the design, development, and use of Information Systems.

Suggested Textbooks:

Software Testing

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Pre-requisites: Analysis of Design of Software Systems

Description:

Software testing concepts including unit tests, test oracles, automated testing, test coverage, integration testing.

Suggested Textbooks:

1. P. Ammann and J. Offutt. *Introduction to Software Testing*. Cambridge University Press, 2017.

Formal Methods in Software Engineering

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Pre-requisites: Analysis of Design of Software Systems

Description:

Concepts in formal specification and verification of systems: linear time logic, Hoare logic, concrete and symbolic model checking.

Suggested Textbooks:

Object-Oriented Design

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Pre-requisites: Advanced Programming

Suggested Textbooks:

1. J. Arlow and I. Neustadt. *UML 2 and the Unified Process*. 2nd Edition, Addison-Wesley, 2005.
2. H. Gomaa. *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*. Cambridge University Press, 2011.
3. G. Booch, R. A. Maksimchuk, M. W. Engel, B. J. Young, J. Conallen, and K. A. Houston. *Object-Oriented Analysis and Design with Applications*. 3rd Edition, Addison-Wesley, 2007.
4. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
5. C. Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd Edition, Prentice-Hall, 2004.

Database lab

Credit Hours: 1.0
Lecture Contact Hours: 0   Lab Contact Hours: 1
Prerequisite:

Description:
It will be presented according to the syllabus of Principles of Database design course.

Suggested Textbooks:

Foundations of Operations Research

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: -

Description:

Operations research helps in solving problems in different environments that needs decisions. The module convert topics that include: linear programming, Transportation, Assignment, and Critical Path Method (CPM). Analytic techniques and computer packages will be used to solve problems facing business managers in decision environments. This module aims to introduce students to use quantities methods and techniques for effective decisions–making; model formulation and applications that are used in solving business decision problems.

Course Components • Introduction to Operations Research (OR) • Introduction to Foundation mathematics and statistics • Linear Programming (LP), LP and allocation of resources, LP definition, Linearity requirement • Graphical LP methode • Simplex method definition, formulating the Simplex model • Sensitivity Analysis: Changes in Objective Function, Changes in RHS. •Duality in linear programming. Economic Interpretation of duality. • The Transportation Model • Basic Assumptions. • Solution Methods: Feasible Solution: The Northwest Method, The Lowest Cost Method; Vogel method • The Assignment Model: Basic Assumptions • Solution Methods: Short-Cut Method (Hungarian Method).

Suggested Textbooks:

1. Hamdy Taha, Operations Research An Introduction.
2. Hillier, Liberman, Introduction to Operation research.
3. Bradly, Hax, Magnenti, Applied Mathematical Programming.
4. M. Bazaraa, Linear Programming and Network Flow.

Mobile Programming

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Advanced Programming (Java)

Description:

This is an applied yield-level course that educates students to develop mobile applications mainly for devices running Android OS (as the main focus, 80% of the course). We further introduce cross platform application development based on web platform (PhoneGap, Cordova, PWA) and other custom platforms such as Xamarin, React Native and Flutter (as the remaining 20% of the course). The later is mostly based on students activity and presentations. The curriculum is organized as follows:

1. Introduction to Android OS, its layers and design philosophy
    1. Linux kernel, HAL, Core native libraries, Dalvik VM, SDK libraries, Apps
    2. Component Oriented App programming
2. Introduction to Android Application Programming
    1. The four building block components (view components: Activities keeping View objects, background components: Services, event broadcasting components: BroadcastReceivers , data access components: ContentProviders)
    2. Intents as a massega packet and a glue to inter connect different app components
    3. UI/UX, using Activities, Views, ViewGroups and Layouts
    4. Intra-Component event handling using different EventListeners
    5. Separation of concerns in View layout (XML resource files) and View actions (Java code)
    6. Capturing data from motion and environment sensors (accelerometer, gyroscope, temperature, humidity, compass, light, linear acceleration, gravity, ...)
    7. Geolocation sensors (GPS/GNSS and network aided location)
    8. Camera and microphone
    9. Programming for SMS and telephony
    10. Event driven programming based on BroadcastReceiver components
    11. Data access using Android Sqlite db adapter
    12. Database abstraction layer (DBAL) and ORMs in Android
3. Introduction to Cross Platform Mobile Programming Foundations
    1. Introduction to web-based cross platform foundations such as Proggressive Web Apps (PWA), PhoneGap, Cordova, Ionic, Framework7 (Android, IOS, Windows, Blackberry, Web, ... platforms)
    2. Student presentations:
        1. Introduction to React Native (Android, IOS)
        2. Introduction to Xamarin (Android, IOS)
        3. Introduction to Flutter (Android, IOS)

Suggested Textbooks:

1. *Android SDK Documentations*, url: https://developers.android.com
2. R.Meier, I. Lake, *Professional Android*, 4th Ed.

GPU Programming

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Computer Architecture, Operating Systems

Description:

The multicore revolution, parallel architectures, pipelined execution, superscalar architectures, multicore architectures, concurrency programming principles, Pthreads, GPGPU architectures, CUDA programming concepts, profiling CUDA programs, Memory Management on Modern GPUs, Unified Memory Concepts, Streams and Concurrency on GPUs, OpenACC programming concepts, Optimizing OpenACC programming, multi GPU architecture, multi GPU programming, CUDA-Aware MPI Programming.

Suggested Textbooks:

1. G. Barlas, Multicore and GPU Programming: an Integrated Approach, 2015, Morgan Kaufmann, 2015.
2. John Cheng, Max Grossman and Ty McKercher. *Professional CUDA® C Programming*. Wiley, 2014.
3. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional; 1st edition (July 19 2010)

User interface design lab

Credit Hours: 1.0
Lecture Contact Hours: 0   Lab Contact Hours: 1
Prerequisite: -

Description:

- Introduction to UI and UX
- Design Process
- Usable UX Design
- Color Philosophy
- UI Consistency
- UX Writing
- Visual Hierarchy
- Alignment and White Space
- Interactive Prototype Design using Figma
- Illustration and Icon Design using Adobe Illustrator

Suggested Textbooks:

1. Don Norman, *Universal Principles of design*
2. Steve Krug, *Don't Make Me Think,*

Linear Algebra

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite: -

Description:

It is a theoretical and also application based course considering capabilities of representing, modeling and solving affine and quadratic problems by addressing following concepts:

- Vectors, matrices and applications
- Operators and properties
- Linear dependency, vector space and matrix ranks
- Mapping, Inverse, determinant and matrix decomposition
- Forward problems, Linear models and applications
- Inverse problems, Linear Equation Systems and applications
- Projection and Least Square Error
- Quadratic forms, Eigen vector decomposition and applications

Suggested Textbooks:

1. Harville, David A. "Matrix algebra from a statistician's perspective." Taylor & Francis, 1998.
2. Boyd, Stephen, and Lieven Vandenberghe. “Introduction to applied linear algebra: vectors, matrices, and least squares.” Cambridge university press, 2018.

Green Computing

Credit Hours: 3.0

Lecture Contact Hours: 3   Lab Contact Hours: 0

Prerequisite: The course is open to all graduate students, and advanced undergraduates, with a technical background.

Description:

The purpose of this course is to review and introduce concepts of green computing methods. Green computing includes the development of environmentally sustainable manufacturing, carbon-efficient hardware and software, as well as the use of computing for more energy-efficient infrastructure. This course also covers related topics such as green energy, as well as smart grids and their use in computing. A collection of papers describing the state-of-the-art in the field will be made available to students. Students will be required to present and review papers, participate in class discussions, and complete a project on a relevant topic of interest. This course empowers students to learn about the life cycle costs of IT systems and classified them based on environmental competitive advantages. Students also learn to design IT systems with less energy consumption, waste, and other environmental impacts. At the end of this course, students will face real problems and be able to challenge IT systems based on up-to-date criteria.

Suggested Textbooks:

1. Bud E. Smith, *Green Computing Tools and Techniques for Saving Energy, Money and Resources,* 2014
2. Jason Harris, *Green Computing and Green IT Best Practice*, 2013.
3. Alvin Galea, Michael Schaefer, Mike Ebbers, *Green Data Center: Steps for the Journey,* 2011
4. Toby Velte, Anthony Velte, Robert Elsenpeter, *Green IT*, 2008

Randomized Algorithms

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Configurable Architectures

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Evolutionary Computing

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Description:

Suggested Textbooks:

Statistical Pattern Detection

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Description:

Suggested Textbooks:

Fuzzy Systems

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Description:

Suggested Textbooks:

Advanced Database Systems

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Description:

Suggested Textbooks:

Optimization Algorithms

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Web Programming

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite: Fundamentals of Computer Programming

Description:

This is an applied yield-level course that teaches students both client and server side web programming considering emerging languages and technologies. The goal is to educate full-stack web programmers and the curriculum contains following concepts

1. Introduction to HTTP and www
    1. Web protocol, web server, web client, web request and response
2. Client Side Web Programming
    1. HTML 4,5, page layout
    2. CSS 2,3, theme design and styling
    3. JS, dynamic client side Web-Apps
3. Introducing Client Side libraries/frameworks
    1. Bootstrap, JQuery and their philosophy
    2. Student presentations: VueJS, AngularJS, Quasar, Ionic
4. Server Side Web Programming
    1. PHP / Python for server side pages
    2. Web service / Web API
    3. DB Connectivity
5. Emerging Concepts/Technologies:
    1. Web II:
        1. PWA (Progressive Web App)
        2. LocalCache, LocalStorage, IndexedDB, Local FileSystem, WebSQL,
        3. Web Sockets, WebNotification, MediaStream, WebRTC
    2. Web III:
        1. Immutability, digital/virtual properties
        2. Blockchain in web (PulkaDot, ...)
        3. Metaverse

Suggested Textbooks:

1. W3Schools Tutorials, url: https://www.w3schools.com
2. R. Ahmad, Full Stack Web Development For Beginners: Learn Ecommerce Web Development Using HTML5, CSS3, Bootstrap, JavaScript, MySQL, and PHP

Parallel Algorithms

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Data Mining

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0

Prerequisite: Graduate standing or consent of instructor

Description:

This is an applied course that teaches students how to solve real-world data mining problems based on already available and existing solutions and/or algorithms. In the course, students learn how to use various data mining and machine learning techniques to solve problems without inventing a new algorithm or technique. The course covers basics of data analysis and statistical techniques. In addition to pre-processing, visualization techniques, and feature selection, basic and advanced topics on clustering, classification, association rules, frequent item set mining, and assessment techniques are covered. No prior exposure to machine learning and Artificial Intelligence is assumed. Theoretical foundations relating to the course are covered but no algorithm novelty is expected from students. Students are obliged to do a real-world problem as a partial fulfilments for course.

Suggested Textbooks:

1. Han, Jiawei, Micheline Kamber, and Data Mining. "Concepts and techniques." Morgan Kaufmann 340 (2006): 94104-3205.
2. Zaki, Mohammed J., and Wagner Meira Jr. Data mining and machine learning: Fundamental concepts and algorithms. Cambridge University Press, 2020.

Modeling of Biological Systems

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Intro to Computational Biology and Bioinformatics

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Text Mining

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Information Networks

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Advanced Engineering Mathematics

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Business Intelligence

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

E-Commerce Security

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Knowledge Management Systems

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Scientific computations and data analysis

Credit Hours: 3.0
Lecture Contact Hours: 3   Lab Contact Hours: 0
Prerequisite:

Suggested Textbooks:

Computer Graphics

Credit Hours: 3.0
Lecture Contact Hours: 3 Lab Contact Hours: 0
Prerequisite:

Description:
Introduction to graphical systems: raster scan, random scan, color and DVTS systems. Graphical standards: GKS, Phigs, Phigs+. Base outputs: vector, circle, oval, curve types, characters and texts. Different algorithms about base outputs: filling surfaces, scan line boundary, Nicholl-Lee-Nicholl, Cohen & Sutherland, Liang & Barsky. Optical pen, mouse, Bezier, Octree, CSG, fractal, transform, rotation, reflection, 3D view, surfaces deletion algorithms with Z-buffer method, surface shading, Guraud and Phong algorithms, introduction to Ray Tracing.

Suggested Textbooks:

Information Technology Project Management

Credit Hours: 3.0

Lecture Contact Hours: 3 Lab Contact Hours: 0

Prerequisite: Graduate standing or consent of instructor

Description:

This is a practical course that introduces students to the role of the project manager and the fundamental concepts and competencies necessary to lead projects. A high-level overview of the project management processes and 9 knowledge areas (Project Integration and Scope Management, Project Schedule and Cost Management, Quality and Risk Management, Project Team Leadership, Communication, and Resource Management) with an emphasis on project influences and project manager sphere of influence will provide in this course. The course also covers Project Life Cycle (PLC), System Development Life Cycle (SDLC) and System Engineering (SE) concepts. Students are obliged to manage a real-world project using related software (MSP) as a partial fulfilments for the course.

Suggested Textbooks:

1. Schwalbe, Kathy. Information technology project management. Cengage Learning, 2015.
2. Guide, A. "Project management body of knowledge (pmbok® guide)." Project Management Institute. Vol. 11. 2001.
3. Walden, David D., Garry J. Roedler, and Kevin Forsberg. "INCOSE systems engineering handbook version 4: Updating the reference for practitioners." INCOSE International Symposium. Vol. 25. No. 1. 2015.

Special Topics in Software Engineering (Multi-Core and GPU Programming)

Credit Hours: 3.0

Lecture Contact Hours: 3 Lab Contact Hours: 0

Prerequisite: Computer Architecture, Operating Systems

Description:

- The Multicore Revolution (Third software crisis and its origin and roadmap to its solutions)
- Different Parallel Architectures & Multicore Architectures (From pipeline to Homogeneous & Heterogeneous Multi-core architectures)
- Memory Hierarchies & Shared Memory Issues (Different memory levels, Cache coherency problem & Concurrency problem and their solutions)
- Parallel Programming Concepts with emphasis on Shared Memory programming
- Shared Memory Programming Optimization Techniques
- Creating parallelism (Design Patterns in Multi-core Programming)
- Open-MP Programming
- Introduction to GPGPU Architecture
- GPGPU programming (CUDA)
- CUDA Programming Concepts

Suggested Textbooks:

1. G. Barlas, Multicore and GPU Programming: an Integrated Approach, 2015, Morgan Kaufmann, 2015.
2. Quinn M.J, Parallel Programming In C With Mpi And Openmp , Mcgrawhill, 2003.
3. John Cheng, Max Grossman and Ty McKercher. *Professional CUDA® C Programming*. Wiley, 2014.
4. Jason Sanders and Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional; 1st edition (July 19 2010)

Quick Programming Challenges

Credit Hours: 3.0

Lecture Contact Hours: 3   Lab Contact Hours: 0

Prerequisite: Advanced Programming, Design and Analysis of Algorithms (or Concepts of Programming for the elite students with the approval of the teacher)

Description:

The purpose of this course is to give a thorough introduction to competitive programming. It is assumed that the students know the basics of programming, but no previous background in competitive programming is needed. The course is especially intended for students who want to learn algorithms and possibly participate in contests such as the International Collegiate Programming Contest (ICPC). Of course, the course is also suitable for anybody else interested in problem solving or competitive programming.

At the end of the course, the students will be capable of using most techniques that are required for participating in programming contests and they should see their improvement in solving various algorithmic problems.

Although the students will see some parts of the course materials in the courses “Data Structures and Algorithms” and also “Design and Analysis of Algorithms”, the students learn many applications and also advanced topics related to the classical data structures and algorithms in this course. Additionally, in order to choose the best implementation of an algorithm during the contests, various implementation issues related to the studied algorithms will be mentioned and analysed during the course.

The course has many problem solving home works and exams so that the students become master in using the algorithms and also in preparing their own code books.

Some main topics of the course are listed below:

- Simple overview of the time complexity
- Simple data structures and their related libraries in the programming languages
- Advanced data structures
  - Binary indexed tree
  - Segment tree
- Complete Search
  - Generating all subsets or permutations

- Backtracking and pruning
  - Binary search on the solution
- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
  - From backtracking to memorization and then dynamic programming
  - Bit masking
- Graph Algorithms
  - BFS and its applications
  - DFS and its applications
    - § Topological sort
    - § Eulerian Path/Cycle
    - § Finding connected components, strongly connected components, bridges, and articulations points
  - Single source and all shortest path algorithms
  - Finding maximum unweighted and weighted matching of bipartite and also general graphs and their applications
  - Maximum flow of a graph and many of its applications
- Game Theory
  - Sprague–Grundy theorem
- String algorithms

Suggested Textbooks:

1. Laaksonen, Antti . *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. 2nd. Springer, 2020.

2. Erickson, Jeff . *Algorithms*. 2019.

3. Cormen, Thomas H., et al. *Introduction to Algorithms*. 4th. MIT Press, 2022.

4. Halim, Steven and Felix Halim. *Competitive Programming 3: The New Lower Bound of Programming Contests*. 3rd. Lulu.com, 2013.

Information Storage and Retrieval

Credit Hours: 3.0

Lecture Contact Hours: 3 Lab Contact Hours: 0

Prerequisite: Data Structures and Algorithms

Description:

The objective of the data structures course was to teach ways of efficiently organizing and manipulating data in main memory. In this course, you will learn equivalent techniques for organization and manipulation of data in secondary storage. In the first part of the course, you will learn about "low level" aspects of file manipulation (basic file operations, secondary storage devices and system software). In the second part of the course, you will learn the most important high-level file structures tools (indexing, co-sequential processing, B trees, Hashing, etc). You will apply these concepts in the design of C++ programs for solving various file management problems.

Course Outline:

1. Introduction to file management. Fundamental file processing operations.
2. Managing files of records. Sequential and direct access.
3. Secondary storage, physical storage devices: disks, tapes and CD-ROM.
4. System software: I/O system, file system, buffering.
5. File compression: Huffman and Lempel-Ziv codes.
6. Reclaiming space in files. Internal sorting, binary searching, keysorting.
7. Indexing.
8. Co-sequential processing and external sorting
9. Hashing
10. Extendible hashing
11. Multilevel indexing and B trees
12. Indexed sequential files and B+ trees
13. Introduction to File Systems, Storage devices, Files and Directories and Reliable Storage

Reference: https://www.site.uottawa.ca/~lucia/courses/2131-05/

Suggested Textbooks:

1. Folk, Michael J., Bill Zoellick and Greg Riccardi. *File Structures: An Object-oriented Approach with C++*. 3rd. Addison-Wesley, 1998.

2. Anderson, Thomas and Michael Dahlin. *Operating Systems: Principles and Practice*. 2nd. Recursive Books, 2014.