COURSE STRUCTURE- 2015

5-Year Integrated Master in Technology (CS)

I-Semester			II-Semester		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
EN101	**English-I**	4-0-0	**EN151**	**English-II**	4-0-0
MM103	**Math-I**	4-0-0	**PY151**	**Waves & Oscillations, Sound & Light theory**	4-0-0
SB101	**Environmental Science**	3-0-0		**Math-II**	4-0-0
SB102	**Foundation Biology**	3-0-0	CA500	Mathematical Foundations of Computer Science	3-0-0
EC101	**Introduction to Economics**	4-0-0	**PY152**	***Waves, Oscillations & Optics Lab***	3
MM203	**Probability & Statistics**	3-0-0	CA521	Programming Methodology	3-0-0
	IT Workshop	2	CA510	*PM Lab*	2
		23			**23**
III-Semester			**IV-Semester**		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
MM202	**Math-III**	4-0-0	CA502	Computer Based Optimization Techniques	3-0-0
	Electronics Devices & Circuits (EDC)	4-0-0	CA522	Data & File Structures	3-1-0
CA501	Computer Based Numerical Methods	3-0-0	CA524	Object Oriented Design	3-0-0
CA520	Computer Organization & Architecture	4-0-0	CA528	Theory of Computation	3-0-0
CSxxx	*IT Lab(Numerical methods)*	2	CSxxx	Signals & Systems	3-0-0
	EDC Lab	2	CSxxx	*Java Lab*	2
	Information Security	3-0-0	CSxxx	*IT Lab(DFS)*	2
			CS711	*Communication Skills Lab*	2

		22			**22**
V-Semester			**VI-Semester**		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
CA523	Operating Systems	3-0-1	CA525	Software Engineering	3-0-0
CA545	Internet Technologies	3-0-0	CA527	Computer Networks	3-0-0
CA526	Data Base Management System	3-0-0	CA547	Computer Graphics & Visualization	3-0-1
CA529	Algorithms	3-1-0	CSxxx	Compiler Design	3-0-1
CSxxx	Principles of Programming Languages	3-0-0		Elective –I	3 or 4
CA512	*DBMS Lab*	2	CS712	*Software Engineering Lab*	2
CSxxx	*IT Lab (Internet Technologies)*	2	CSxxx	*IT Lab (CN)*	2
				Summer Internship	
		22			**21/22**

VII-Semester			**VIII-Semester**		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
CS786	Network Programming	3-0-0	CSxxx	Virtualization	3-0-0
AI720	Essentials of AI	3-0-0		Elective –V	3 or 4
CSxxx	Advanced Computer Architecture	3-0-0		Elective – VI	3 or 4
	Elective –II	3 or 4		Elective – VII	3 or 4
	Elective –III	3 or 4		Elective –VIII	3 or 4
IT711	*UNP Lab*	2	CSxxx	Virtualization *Lab*	2
CSxxx	*IT Lab(AI)*	2		Electuve - IX	3-0-0

CS710	*RMCS Lab*	2			
		21/23			20/24

IX-Semester			**X-Semester**		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
CS718	Project	9	CS719	Project	12

Total Credits: 196/203

1. Introduction

Credits: 4-0-0

2. Course Outline

UNIT - I: Matrices
Basic concepts of matrices, multiplication of matrices by scalars, addition and multiplication of matrices, transpose, trace, determinant of a matrix, rank and inverse of a matrix, special matrices such as Hermitian, unitary matrices, system of linear equations, solution by Crammer's rule, existence and general properties of solutions, eigenvalues, eigenvectors, diagonalization of matrices, functions of matrices and Caley-Hamilton theorem.

UNIT - II: Elementary functions
Defnition and examples of sequences and series. Using these, study Trigonometric functions, logarithmic, exponential function, hyperbolic trigonometric functions

UNIT – III: Analytical geometry in 3-D
Cartesian co-ordinates in 3-D, distance between two points, direction cosines, direction ratios and their properties, equation of plane using given data, equation of straight line in different forms, image of a point with respect to a plane, distance between a point and a plane along a straight line, equation of sphere, circle.

UNIT – IV: Complex numbers, vector algebra
Algebra of complex numbers, polar form, argand diagram, triangle inequality, curves and regions in complex plane. Addition of vectors, dot product, cross product and their geometric interpre-tation, triple product, area, volume given in terms of vector products**.**

Reference:

1. **Linear Algebra, Kenneth Hoffman and Ray Kunze,pearson,1971**
2. **Linear Algebra: A geometric approach, S Kumaresan,Pre…hall of.. 2004**
3. **Calculas and analytic geometry , George Thomas & Ross Finney..1995**

ENVIRONMENTAL STUDIES

1. Introduction

Credits: 4-0-0

2. Course Outline

UNIT - I: The multidisciplinary nature of environmental studies
Definition, Scope and importance Environmental Milestones

UNIT - II: **Ecosystems**
Concept of an ecosystem (Abiotic and biotic environment), Structure and function of an ecosystemProducers, Consumers and decomposers.Energy flow in the ecosystem, (Nutrient cycle in the ecosystem) Ecological succession Food chain, food webs and ecological pyramids. Introduction, types, characteristic features, structure and function of the following cosystems: Forest ecosystem,Grassl and ecosystem, Desert ecosystem aquatic ecosystems (ponds,streams,lakes,rivers,ocean estuaries).

UNIT – III: Biodiversity
Conservation: Introduction, Definition: genetic, species and ecosystem diversity, Hot spots of biodiversity, Threats to biodiversity: habitat loss, poaching of wildlife, man wildlife conflicts. Endangered and endemic species of India, Conservation of biodiversity: In- situ and Ex-situ conservation of biodiversity.

UNIT – IV:
Natural Resources and non-renewable resources An overview of natural resources and associated problems with references to a) Forest resources b) Water resources c) Mineral resources d) Food resources e)Energy resources f)Land resources.

UNIT – V : Environmental Pollution
Definition a) Air pollution b)Water pollution c) Soil pollution d)Marine pollution e) Noise pollution f) Thermal pollution g) Nuclear pollution Solid waste management: Causes, effects and control measures of urban and industrial wastes.Environmental impact assessment.

UNIT – VI:
Environment: Socio-Political Issues: Environmental Ethics: Science/Nature Debate) Global Environmental Issues a) Global Warming, Climate Change, Acid Rain, Ozone layer depletion, Pollution) Global Environmental Issues a) Deforestation, Loss of Biodiversity, Desertification, Land Degradation) Global Conventation on Environment: Indian environmental Laws.

UNIT – VII: Human –Environment Interaction
Modes of Resource Use, Development and Environment Population growth and Environment, Environment and Human Health. Unit 8: Project Work (Group of....) Select List of Themes: Study of simple ecosystem: pond/lake/rocks, iodiversity Register of Campus Pollution Energy sources Water Conservation Waste Disposal and Recycling GM crops.

FOUNDATION BIOLOGY

1. Introduction

Objective: A course to expose students to basics of biology

Credits: 3-0-0

2. Course Outline

UNIT - I:
Biological kingdoms, comparative account of prokaryotes and eukaryotes at cellular level;

UNIT - II:
Overview of interactions between prokaryotes and eukaryotes;

UNIT – III
Cells, organization, and functions of organelles; Principles of genetics or patterns of inheritance, theory of evolution;

UNIT – IV:
Biomolecules: water, sugars, amino acids, nucleotides, carbohydrates, lipids, proteins and nucleic acids;

UNIT – V :
Enzymes, chemical reactions, glycolysis, respiration, fermentation photosynthesis, nitrogen fixation, transpiration, osmosis Overview of development;

UNIT – VI:
Digestive system, circulatory system,nervous system, endocrine glands, reproductive system.

3. Reading Material

Reference Books:

1. Biological Science, Cambridge Low Price Edition, by N.P.O. Green, G.W. Stout, D.J. Taylor, Editor: R. Soper. XI and XII Biology CBSE Books

PROBABILITY AND STATISTICS

1. Introduction

Credits: 4-0-0

2. Course Outline

Random Experiments Sample spaces , Events, probability measures on events- definition, properties, examples. Conditional probability-de_nition, properties, examples, Bayes theorem, independent events.

Definition of random variables, standard discrete and continuous random variables-viz. Bernoulli, Binomial, Geometric, Poisson, Exponential, Gamma, Normal their probability distributions, . Expectation, variance,other properties. Definition of bivariate random variables, joint distributions, covariance and correlation between two random variables, independence, distributions of sums

Data collection methods, types of data, graphical summaries of data, numerical summaries of univariate data, bivariate summaries, measures of association. Introduction to statistical inference, population parameters, variable(s) of interest, statistic, estimators as random variables.

3. Reading Material

1. Ross, S. A First Course in Probability, sixth edition, Pearson Education, 2007.
2. Ramachandran, K.M. and Tsokos, C.P. Mathematical Statistics with applications, Academic Press, 2009.
3. Daniels, W.W. Biostatistics: a foundation for analysis in the health sciences, 9th edition, John Wiley & Sons, 2008
4. Moore, D.S. The Basic Practice of Statistics, W. H. Freeman, 2003.

IT WORKSHOP

1. **Introduction**
   Credits: 2

2. **Course Outline**

UNIT - I: Hardware
Identifying the various hardware components of a computer.

UNIT - II: Operating System Installation
Installation of a dual-boot operating system – Windows and Linux.

UNIT - III: Linux Commands
File manipulation commands (mkdir, rmdir, cp, mv, ls, rm, pwd, chmod, chown, umask),Process related commands (ps, top, kill, fg, bg), Environment (env, ˙bashrc,˙bash profile,export), Piping and redirection (|,>,>>,<,<<), Miscellaneous commands (sort, wc, alias, locate, whereis, which, pushd, popd, who, whoami, echo,cut, paste) Searching for files and contents of files (find, grep, awk, sed)

UNIT - IV: Document Preparation with Latex
Basic documents with sectioning, figures and tables, captioning, cross-referencing,multi-column and classes of documents, mathematical formulae, table of contents, Title page, Numbering, Page breaks, Boxes.

UNIT - V: Presentations with Beamer
Basic presentations using beamer, overlays, animation, using tables, figures and columns in presentations, boxes for definitions etc.

UNIT - VI: Data Analysis with Spreadsheets
Using basic sheet column definition, calculating simple statistics such as Average, Standard Deviation etc., setting up formulae for calculation of values of columns using regular formulae and/ or VLOOKUP, using pivot tables, drawing charts (bar, pie) and graphs using the data in the sheet.

UNIT - VII: Drawing Tools
Using tools such as OODRAW or Xfig or Calligra to draw figures to include in documents by exporting to .eps format.

3. **Reading Material**

Text Books

1. Stephen G. Cochan and Patrick Wood. Unix Shell Programming, 2003.
2. Brian W. Kernighan and Rob Pike. The Unix Programming Environment.
3. Leslie Lamport. A Document Preparation System: LATEX, User's Guide and Reference Manual, Pearson Education.
4. Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins. Linux in a Nut- shell, 6th edition, O'Reilly Publishers, 2009.
5. Xfig reference manual.

Suggested Assignments

1. List all directories in a directory that start with a particular letter.
2. Count the number of files in a directory.
3. Find all files with a given substring in the name.
4. Prepare a document about your favorite scientist with the title, author name (i.e., student name) and reg. no. in the first page which includes photos, text about the person referring to the photos.
5. Add tables to a document, formulae and refer to all these in the text of the document. Create a table of contents for the document from its sections and sub-sections limiting the TOC to two levels.
6. Using different schemes of Beamer, overlays, columns and boxes, create a presentation.
7. Give data sets such as marks of students and ask that the excel sheet be set up to calculate totals based on the columns, grade table set up using VLOOKUP.
8. Give data sets that let students generate pie charts, bar charts and other graphs.
9. Using Xfig or OODRAW, draw flowcharts, graphs, paths in graphs etc.

WAVES, OSCILLATIONS, SOUND & LIGHT

1. Introduction

Credits: 4-0-0

2. Course Outline

Simple harmonic motion, Angular simple harmonic oscillator, damped harmonic oscillator, Forced oscillations and Resonance; Simple coupled oscillators.Traveling waves, Superposition principle, Wave speed, Power and intensity in wave motion, Interference of sound waves, Stationary waves, Beats, Waves on strings and surfaces, Audible, ultrasonic and infrasonic waves, Propagation and speed of longitudinal waves, Vibrating systems and sources of sound, Musical instruments, The Doppler effect, Shock waves, Velocity of sound and its measurement, factors affecting the speed of sound Nature and propagation of light, Images, Defects of images, Spherical and Chromatic aberrations, Achromatism of two thin lenses separated by a distance, Optical instruments (Microscopes and Telescopes), Velocity of light and its measurement.

3. Reading Material

1. Fundamentals of Physics by Resnick, Halliday and Walker, 6th Wiley
2. University Physics by Sears and Zemansky, 10thedition, Addison –Wesley series
3. Fundamentals of Optics, Jenkins and White
4. Light by K. G. Mazumdar
5. Geometrical and Physical Optics by P. K. Chakraborty
6. Opticsby B. K. Mathur.

MATH-II

1. Introduction

Credits: 3-0-0

2. Course Outline

UNIT - I: Multivariable calculus

Brief introduction to co-ordinate systems- spherical and cylindrical systems- double integral over a rectangle, double integral over a region, change of order of integration, triple integrals, change of variables and Jacobian. Vector fields, gradient, divergence, curl, vector calculus identities, parametric curves, line integrals, path dependence, fundamental theorems of line integrals, conservative fields, application of Greens theorem in 2-D, parametric surfaces, surface of revolution, surface integrals, applications of Stokes theorem and Gauss divergence theorem, Green's identities, statement of integration by parts.

UNIT - II: Mathematical Analysis

Rational numbers, sequences, subsequences, monotonicity, boundedness, convergence, limit of a sequence, Cauchy criteria, series, di_erent tests of convergence, power series, radius of convergence

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

1. Introduction

Objective: The purpose of this course is to understand basic concepts of mathematics (definitions, proofs, sets, functions, and relations) and use (abstract) discrete structures(integers, bits, strings, trees, and graphs) that are backbones of computer science.

Credits: 3 -0-0

2. Course Outline

UNIT - I: Logic
Propositional logic, logical equivalence, predicates & quantifiers, and logical reasoning. Boolean Algebra

UNIT - II: Sets, Relations &Functions
Basics, set operations, Properties, Combining relations, Closures, Equivalence, partial ordering, One-to-one, onto, inverse, composition, graphs

UNIT - III: Algorithms on Integers
Greatest common divisor, Euclidean algorithm.

UNIT - IV: Mathematical reasoning

Proof strategies, Mathematical Induction, Recursive definitions, Structural Induction

UNIT - V: Counting
Basic rules, Pigeonhole principle, Permutations and combinations, Binomial coefficients and Pascal triangle

UNIT - VI: Probability
Discrete probability. Expected values and variance

UNIT - VII: Graph Theory
Paths, degree sequences, trees, minimum spanning trees, shortest path, bipartite matching, Tutte's theorem, connectivity, flows, graph coloring

3. Reading Material

Text Books

1. Discrete mathematics and its application: K H Rosen

References

1. Discrete and Combinatorial mathematics : Grimaldi
2. Concrete mathematics: Donald Knuth
3. Discrete mathematics with applications: Sussana Epp

WAVES, OSCILLATIONS, SOUND & LIGHT LAB

1. Introduction

Credits: 3

2. Course Outline

Lab Exercise:

1. Coupled Oscillator- Measurement of Normal Mode Frequencies
2. KUNDT'S TUBE Determination of Velocity of Sound in Air
3. SONOMETER –Resonance Modes of a Stretched String & Velocity of Wave Propagation
4. BREWSTER ANGLE METHOD –Measurement of Refractive Index of Dielectric Material
5. FRESNEL BIPRISM –Determination of Wavelength of Light by Interference
6. NEWTON'S RINGS –Determination of Radius of Curvature of a Lens
7. FABRY ─PEROT INTERFEROMETER –Measurement of AirgapThickness
8. DIFFRACTION GRATING –Determination of Wavelengths of mercury vapor lamp.

PROGRAMMING METHODOLOGY

1. Introduction

Objective: This course teaches the basics of solving problems by writing computer programs. At the end of the course, a student will be able to think programmatically and write programs in a procedural/ imperative language. The primary language used for programming is 'C.'

Credits: 3

2. **Course Outline**

UNIT – I
Introduction to problem solving; Problems and problem instances; Informal approach to program design: generalisation, special cases, algorithms, breaking down a problem into functions, input and output.

UNIT – II
Introduction to the 'C' programming language; program structure; main() function; unnamed and named blocks; basic data types, variables, declaration and definition; initialisation and assignment; arithmetic operators and precedence; implicit and explicit type conversions; arrays; boolean variables and logical operators.

UNIT – III
Control structures: branching and iteration; functions and parameters; break(), return() and exit() functions; local and global variables; function prototypes.

UNIT – IV
Pointer variables and dynamic structures; static and dynamic (run-time) memory structures; static variables; breaking a program across multiple files; creating and linking libraries.

UNIT – V
Detecting and correcting common errors; debugging and debuggers; documenting programs; good programming practices; programming exercise (writing a program of at least 200 lines split across multiple files).

3. Reading Material

Text Books

1. Brian W. Kernighan, Dennis M. Ritchie. "The C Programming Language, 2nd Edition," Prentice-Hall India.

Additional Readings

1. G. Michael Schneider. "Introductionto Programmingand Problem SolvingwithPASCAL," JohnWiley and Sons.
2. Brian W. Kernighan and R. Pike. "The Unix Programming Environment," Prentice-Hall India.
3. Chakravarthy Bhagvati. "How to Program (An Informal Guide)," http://dcis.uohyd.ernet.in/ ~chakcs/howtoprogram.pdf

Suggested Assignments

1. Write a program for *Collatz Conjecture*.
   Input: An integer x($< 10^7$) **Problem:** If x is *even*, x = x/2 If x is *odd*, then x =3x +1 Repeat the above step untilx =1. **Output:** The number of times the above step is repeated until x =1.

2. Write a program to verify if an input string is a valid vehicle number.
   Note that a vehicle number is of the form: CCDDCDDDD or CCDDCCDDDD where 'C' stands for a single letter and 'D' stands for a digit.

3. Read "AdditionalReading3" carefully; follow the steps given there and writea program to produce a 4 × 4 Magic Square.

MATH- III

1. Introduction

Credits: 4 -0-0

2. Course Outline

UNIT - I: Ordinary Differential Equations
Order and degree of a di_erential equation, _rst order equations: variables separable method, homogeneous equations of degree zero, nonhomogeneous equations, exact equations, integrating factor, linear equations, Bernoulli's equation. Higher order homogeneous linear equations with constant coefficients, second order homogeneous linear equation with variable coefficients, variation of parameters, 2 x2 autonomous system of equations, power series solution, meaning of existence and uniqueness of a solution and some counter examples.

UNIT - II: Laplace Transforms
Definition, L.T. of some elementary function, e_ect of L.T. on translation, scaling, convolution. Inverse Laplace transform, applications of L.T. to ODE.

UNIT - III: Fourier series
Fourier series of a periodic function, half range Fourier series.

UNIT - IV: Sets, relations and functions
Sets, relations, equivalence, partial ordered relations, mathematical induction, elements of

ELECTRONICS DEVICES & CIRCUITS

1. Introduction

Credits: 4-0-0

2. **Course Outline**

UNIT – I: ElectronicDevices and Circuits
Voltage and current sources, Kirchhoff's voltage and current law, Superposition theorem, Thevnin's theorem, Norton's theorem, Maximim power transfer theorem.

UNIT – II: Diode
Semiconductors, p-n junction Diode, Working principle and characteristics of p-n junction diode, Zener diode, LED, Photo-diode, Half-Wave, Full-Wave and Bridge rectifiers, efficiency, ripple factor, voltage regulation. Clipping, clamping, voltage doublers and multipliers.

UNIT – III: Transistors
pnp and npn structures; CE,CB and CC configuration, input and output characteristics, α β and γ, cut off, active and saturation regions, biasing and bias stability, load line and Qpoint, Transistor as an amplifier (CE), Transistor as a switch, Types of FET, construction of junction FET, output characteristics, biasing, operating region, pinchoff voltage, MOSFET: enhancement and depletion type, construction, principle of operation and characteristics, elementary idea on CMOS, MOS inverter.

UNIT – IV : Amplifiers
Voltage and current amplifiers, principle of feedback, positive and negative feedback, advantages of negative feedback, multistage amplifier, RC coupled amplifier: frequency response, gain and band- width, class A, class B, Class AB and class C amplifiers, Introduction to Op-Amp.

UNIT – V: Digital Electronics
Decimal, binary and hexadecimal numbers, binary arithmetic, Boolean algebra, logic gates: OR, AND, NOT, NAND, NOR and exclusive¬OR, universal gate, de¬Morgan's theorems, 1's and 2's complement, Boolean simplifications, sum¬ of¬ product and product¬ of¬ sum form, Karnaugh map

3. Reading Material

Text Books

1. Integrated Electronics – J. Millman and C. Halkias (Mc Graw Hill).
2. Microelectronics – J.Millman ((Mc Graw Hill).
3. Electronics Fundamentals and Applications – J. D. Ryder (PHI Pvt. Ltd).
4. Electronic Device and Circuit Theory – R. Boylestad and L. Nashelsky (Prentice –Hall).
5. Operational Amplifier & Linear IC – Gayakwad (TMH)
6. Digital Principles & Applications – Malvino and Leach (TMH)
7. Digital Design by Moris Mano
8. Electronics Principles – A.P.Malvino (TMH).

COMPUTER BASED NUMERICAL METHODS

1. **Introduction**

Objective: To understand Computation and to make use the computers effectively for solving problems. Designing Flowcharts, algorithms for providing apt computation. Students are expected to have functions to provide solutions to numerical problems, which will be useful in their other courses as well as project works

Credits: 3-0-0

Upon successful completion of this course, the student will be able to:

- Quantify absolute and relative errors.
- Distinguish between round-off and truncation errors.
- Interconvert binary and base-10 number representations.
- Define and use floating-point representations.
- Quantify how errors propagate through arithmetic operations.
- Implement the bisection method for solving equations.
- Implement both Newton-Raphson and secant methods.
- Perform basic matrix operations.
- Define and perform Gaussian elimination to solve a linear system.
- Identify pitfalls of Gaussian elimination.
- Define and perform Gauss-Seidel method for solving a linear system.
- Define and identify special types of matrices.
- Use LU decomposition to find the inverse of a matrix.
- Define and perform singular value decomposition; explain the significance of singular value decomposition.
- Define interpolation.
- Define and use direct interpolation to approximate data and find derivatives.
- Define and use Newton's divided difference method of interpolation.
- Define and use Lagrange and spline interpolation.
- Derive and apply the trapezoidal rule and Simpson's rule of integration.
- Distinguish Simpson's method from the trapezoidal rule.
- Estimate errors in trapezoidal and Simpson integration.
- Derive and apply Romberg and Gaussian quadrature for integration.
- Investigate how step size affects accuracy in Euler's method.
- Implement and use the Runge-Kutta 2nd order method for solving ordinary differential equations.

2. **Prerequisites**

Finite representation of numbers, Flowchart, exposure to algorithms, Real Analysis, one of the programing Languages.

3. **Course Outline**

UNIT - I: Transcendental and Polynomial equations
UNIT - II: System of Linear Algebraic equations and eigenvalue problems
UNIT - III: Interpolation and Approximation
UNIT - IV: Differentiation and Integration
UNIT - V: Ordinary differential equations

4. **Reading Material**

Text Books

1. M.K. Jain, S.R.K. Iyengar and R.K. Jain (1993) "Numerical Methods for Scientific and Engineering Computation" Wiley Eastern Limited

References

1. Erwin Kreyszig (1993) "Advanced Engineering Mathematics" Wiley
2. Press, Teukolsky, Vellerling and Flannery (1992) "Numerical Recipes in C - the Art of Scientific Computing" Cambridge
3. Abromowitx and Stegum (1972) "Handbook of Mathematical Functions with formulas, Graphs and Methematical Tables" National Bureau of Standards Applied Mathematics Series . 55 Siperintendent of Documents U.S. Government Printing Office Washington, DC 20402
4. Jaan Kiusalaas (2009) " Numerical Methods in Engineering with MATLAB" Cambridge University Press (http://mhbb2012.persiangig.com/.uxdhABZETd/Numerical%20Methods%20in%20Engineering%20with%20MATLAB,%202nd%20Edition%20Sep%202009.pdf)

Note: Recommended to use MATLAB for Demos explaining/visualizing the concepts

COMPUTER ORGANIZATION AND ARCHITECTURE

1. Introduction

Objective: To build up foundation for the study of hardware organization of digital computers. To understand the structure, function and characteristics of computer systems including various building blocks of computers, without being specific to any particular architecture. Understand importance of interaction between hardware and system software

Credits: 4-0-0

2. Course Outline

UNIT - I: Computer Evolution & Arithmetic

A Brief History of computers, Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating point numbers, Signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm, Hardware Implementation, Division, Restoring and Non Restoring algorithms, Floating point representations, IEEE standards, Floating point arithmetic
.

UNIT - II: The Central Processing Unit

Machine Instruction characteristics, types of operands, types of operations, Addressing modes, Instruction formats, Instruction types, Processor organization, Processor as running example, Programmers model of , max/min mode, Register Organization, Instruction cycles, Read Write cycles, assembly instruction examples to explain addressing modes

UNIT - III: The Control Unit

Single Bus Organization, Control Unit Operations: Instruction sequencing, Micro operations and Register Transfer. Hardwired Control: Design methods – State table and classical method, Design Examples - Multiplier CU. Micro-programmed Control: Basic concepts, Microinstructions and micro- program sequencing

UNIT - IV: Memory Organization

Characteristics of memory systems, Internal and External Memory, Types of memories: ROM: PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM
High-Speed Memories: Cache Memory, Organization and Mapping Techniques, Replacement Algorithms, Cache Coherence, Virtual Memory: Main Memory allocation, Segmentation, Paging, Address Translation Virtual to Physical.
Secondary Storage: Magnetic Disk, Tape, DAT, RAID, Optical memory, CDROM, DVD

UNIT - V: I/O Organization

Input/Output Systems(features and principles), Programmed I/O, Interrupt Driven I/O, Interrupt structure, Direct Memory Access (DMA), features Buses and standard Interfaces: Synchronous, Asynchronous, Parallel I/O features, Serial I/O features, PCI, SCSI, USB Ports Working mechanisms of Peripherals: Keyboard, Mouse, Scanners, Video Displays, Touch Screen panel, Dot Matrix, Desk-jet and Laser Printers.

UNIT - VI: Case Studies

Concepts RISC: Instruction execution characteristics,, RISC architecture and pipelining. RISC Vs CISC. ARM and Embedded Systems PowerPC, Intel X86 Evolution from 32bit to 64bit architectures. AMD Opteron

3. Reading Material

Text Books

1. Pattersen D.A. & Hennesy J.L., Computer Organisation & Design: The Hardware/Software Interface

Reference Books

1. Computer Organization and Architecture, 10/E William Stallings ISBN-10: 0134101618 • ISBN-13: 9780134101613 - See more at:
http://www.pearsonhighered.com/pearsonhigheredus/educator/product/products_detail.page?isbn=9780134101613&forced_logout=forced_logged_out#sthash.WVVJbZUb.dpuf

IT LAB (NUMERICAL METHODS)

1. Introduction

Credits: 2

2. Course Outline

UNIT - I: Transcendental and Polynomial equations

Practical problems:

(i) Bisection Method
(ii) Newton's Method
(iii) Iteration Methods

.

UNIT – II: System of Linear Algebraic equations and eigenvalue problems

Practical problems:

(iv) L U decomposition
(v) Gauss Elimination Method
(vi) Computation of eigenvalues and eigenvectors

UNIT - III: Interpolation and Approximation

Practical problems:

(vii) Lagrange Interpolation
(viii) Hermite Interpolation
(ix) Least Square Approximation
(x) Rational Approximation

UNIT - IV: Differentiation and Integration

Practical problems:

(xi) Numerical Integration based on Interpolation
(xii) Romberg Integration
(xiii) Quadrature method

UNIT – V: Ordinary differential equations

Practical problems:

(xiv) Single step and multiple step method
(xv) Predictor-Corrector Methods

Credits: 2

Outline:

1 . P-N junction diode

a. I~V characterisitics,b) half wave rectifier, c) full wave rectifier d) clipper

2. Bipolar Junction Transistor

a. Transistor characteristics , Transistor application as an amplifier (Common Emitter configuration)

3. Truth table verification – Universal gate realization

4. Adder/substractor

5. seven segment display interface to adder/subtractor.

(Circuit level simulation of above experiments using circuit simulator)

INFORMATION SECURITY

1. Introduction

Objective: This course aims at imparting good understanding of information security, workstation & network security and introduction to various security and cryptography technologies

Credits: 3-0-0

2. Course Outline

UNIT - I: Introduction to Computer Security & Hackers .

UNIT - II: Security Concepts Confidentiality, Integrity, Authority, Privilege and Basics of Defense

UNIT - III: Network Technology OSI layer model, TCP model, Overview of networked systems, Network design and topology**.**

UNIT - IV: Legal Issues

UNIT - V: Basics of Hacking

UNIT - VI: Security Policies

UNIT - VII: Access Control, Permissions & Passwords

UNIT - VIII: Risk and Risk Analysis

UNIT - IX: E-mail & Phishing

UNIT - X: Viruses, Spyware, and Malware

UNIT - XI: Incident Response

UNIT - XII: Computer Forensics

UNIT - XIII: The Human Element

UNIT - XIV: Cryptography

3. Reading Material

Text Books

1. Principles of Computer Security: Conklin, White, Cothren, Williams, and Davis. McGraw Hill.

Reference Books

1. Cryptography and Network Security: W Stallings. PHI.
2. Cryptography Theory and Practice: D R Stinson. CRC.

Suggested Assignments

1. Installing windows and linux
2. Network ports and communication
3. Hacking Practices
4. Cracking Passwords
5. E-Mail Phishing
6. Basic Computer Forensics Lab

COMPUTER BASED OPTIMIZATION TECHNIQUES

1. Introduction

Objective: To become familiar with various optimization techniques

Credits: 3-0-0

2. Course Outline

UNIT - I: Linear Programming

Mathematical formulation of Linear Programming problem, Canonical and standard forms of Linear Programming problem, Solution by Graphical and Simplex methods, Revised Simplex method, Two phase & Big M – method, Duality, Primal – Dual relationship, Dual Simplex method

.

UNIT – II: Special Types of Linear Programming Problem

Transportation problem, Northwest corner method, Stepping stone method, Unbalanced transportation problem, Assignment problem, The Hungarian method

UNIT - III: Integer Programming

Integer Linear Programming problem, Mixed Integer Linear Programming problem, Cutting Plane method, Branch and Bound Technique

UNIT - IV: Dynamic Programming

Bellman's Principle of optimality, General theory of solving multistage decision problems using Dynamic Programming, Application of General Theory to specific problems such as the Traveling Salesman problem.

3. Reading Material

Text Books

1. F S Hillier and G J Lieberman, Introduction to Operations Research, 7th edition, McGraw Hill, 2000
2. H A Taha, Operations Research – An Introduction, 8th Edition, Pearson Prentice Hall, 2007

Reference Books

1. G Hadley, Linear Programming, Narosa Publishing
2. Harvir Singh Kasana and K D Kumar, Introductory Operations Research: Theory and Applications, Springer Science & Business Media, 2004

Suggested Assignments

Computer implementation of

1. Simplex method and its various other forms
2. Northwest corner method, Stepping stone method
3. Hungarian method
4. Cutting plane method
5. Recursive solution of traveling salesman problem using Dynamic Programming

DATA AND FILE STRUCTURES

1. Introduction

Objective: The course would equips the students with knowledge about various basic data structures and their appropriate use in specific applications.

Credits: 3-1-0

2. Course Outline

UNIT - I: Introduction

Introduction to data structures and data types, Tracing of the algorithms, Time complexity analysis of algorithms with respect to data structure operations

UNIT – II: Linear data structures

Arrays, Row-major and Column major order, Polynomial Manipulations Stacks, Recursion with examples from mathematics, graphics, and languages, Infix to Postfix conversion Queues, Priority queues , Linked Lists and their variations (Circular linked list, Double linked list)

UNIT - III: Non-linear Data Structures

Trees, Traversals, Binary Search Trees, Heaps, Splay trees, Graphs, Representations of the graphs, Graph Traversals

UNIT - IV: Search Mechanisms

Linear Search, Binary Search, Hash Tables

UNIT - V: Sorting

Bubble sort, Selection sort, Merge Sort, Heap sort

UNIT - VI: File structures

Indexing, B-trees, B^+ trees

3. Reading Material

Reference Books

1. An introduction to Data Structures with Applications (Second edition), Tremblay and Sorenson, Tata Mecgrahill
2. Data Structures and Algorithms in Java (4th edition), Michael T. Goodrich and Roberto Tamassia, John Wiley and sons
3. A Practical Introduction to Data Structures and Algorithm Analysis, Clifford A Schaffer, Prentice Hall

Suggested Assignments

Constuction of a city database using a linked list and binary search tree and the appropriateness of these structures under various demands for the data. Some problems of similar nature.

OBJECT ORIENTED DESIGN

1. Introduction

Credits: 3-0-0

2. Course Outline

UNIT - I: Object Oriented Fundamentals and Modeling
Need for OOP paradigm, What is object orientation and OO Development, Modelling, Abstraction, Three models of OOD, Object and class concepts, Links and Association Concepts, Generalization and Inheritance, N-ary associations, Aggregation, Abstract classes, multiple inheritance, metadata, Reification, Constraints, Derived data, packages.

UNIT - II: Java Basics
History of Java, java data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring string class.

Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT - III: State Modelling and Interaction Modelling
Events, states, Transitions and Conditions, State Diagram, Nested state diagram, Concurrency

Use-Case model, Sequence model, Activity model, procedural sequence model, Relation between class, state model and interaction model

UNIT - IV:
Hierarchical abstractions, Generalization and Aggregation, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class
Exception handling in Java:
Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling, Exploring java.util

UNIT - V: System Design, Class Design and Implementation Modelling
Overview of system design, performance estimation, reuse plan, Subsystems, Management of data storage and global resources, software control strategy and boundary conditions
Overview of class design, Realizing use-cases, designing algorithms, refactoring and design optimization, Overview of implementation, fine tuning of classes, generalization, and realizing associations

3. Reading Material

Text Books

1. Java: The complete reference, Herbert Schildt, McGrawhill
2. Understanding OOP with Java, T. Budd, Pearson Education
3. Object Oriented Modelling and Design with UML, 2nd edition, Michael Blaha and James Rambaugh, Eastern Economy Edition

THEORY OF COMPUTATION

1. Introduction

Objective: This course aims to give an understanding of the theoretical foundations of computer science. Study of simple models of computation leading to the model for the modern world digital computer, the Turing machine, is one of the primary goals along with understanding limits of computability.

Credits: 3-0-0

2. Course Outline

UNIT - I: Preliminaries on Alphabets and Languages
Finite and infinite sets, fundamental proof techniques.

UNIT - II: Regular Languages (RL)
Definition and examples; Regular expressions (RE), Non-deterministic Finite automata (NFA), ε -NFA and De- terministic Finite automata (DFA); Equivalence of RE, NFA and DFA; Conversions from RE to (ε -) NFA to DFA to RE; Minimal DFA; Moore machine, Melay machine; Closure properties of RL; Pumping lemma for RL.

UNIT - III: Context-Free Languages (CFL)
Definition and examples; Grammar formalism for regular languages, Context free grammar (CFG), Derivation trees, Ambiguity, Normal forms; Push down automata(PDA) (deterministic and non-deterministic); Equivalence of CFG and PDA; Closure properties of CFL; Pumping lemma for CFL.

UNIT - IV: Turing Machine (TM)
Definition of TM; Examples; Variants of TM: Multi-tape and other versions of TM; Nondeterministic TM; Equivalence; Church-Turing Thesis; Universal Turing Machine

UNIT - V: Decidability and Undecidability
Definition of decidability; decidable problems concerning RL, CFL; Recursive and recursively enumerable (re); Undecidability; The Halting problem; Cantor's diagonalization argument; Examples of undecidable problems: Post's correspondence problem.

3. Reading Material

Text Books

1. Introduction to Automata Theory, Languages and Computation, by John. E. Hopcroft, Rajeev Motwani, J. D. Ullman, published by Pearson Education Asia, 2006.
2. Introduction to Languages and the Theory of Computation, by John C. Martin, published by Tata McGraw Hill, 2003

Reference Books:

1. Elements of the Theory of Computation, by H.R. Lewis and C.H.Papadimitrou, published by Prentice Hall Inc, 1981.
2. Introduction to Computer Theory by Daniel I.A. Cohen, published by Wiley India, 1991.

Suggested Assignments:

Construction of DFA, Regular expression, CFG, PDA, TM for a specified language.

Sample assignments:

1. Write a regular expression and construct DFA for the set of all strings $\{0,1\}^+$ in which each block of _ve consecutive symbols contains at least two 0's. Minimize your DFA.
2. Devise a Context free grammar that recognizes the language $\{w \varepsilon \{a, b, c\}^+ : n_a(w) < n_b(w)\}$ where $n_x(w)$ denotes the number of occurrences of the symbol x in w explaining each rule. Construct a push down automaton that recognizes this language.
3. Construct a Turing machine that recognizes the language $\{w \varepsilon \{a, b, c\}^+ : n_a(w) < n_b(w) < n_c(w)\}$. First write the idea behind the machine and then show the δ-function

SIGNALS AND SYSTEMS

1. Introduction

Objective: The aim of the course is to introduce students to mathematical foundations of signals, systems and algorithms to process signals. At the end of the course students will be able to use MATLAB to generate signals and process the same using various types of filters and systems. The practical aspects will be explained through a series of lab assignments on MATLAB

Credits: 3-0-0

2. Course Outline

UNIT - I: Introduction to signals
Brief introduction to signals and their applications. Analog and digital signals.
Continuous and discrete signals. Types of signals: Constant, Step, Ramp, Impulse, Dirac, periodic, exponential. Operations on Signals: Magnitude Scaling, Time shifting, Sampling.

UNIT - II: Introduction to systems
Properties: Memory less, stability, Linearity, Causality, Time invariance. Linear Time invariant systems, Causality and Causal Systems, system stability Integration of systems: Cascading, parallel, feedback.

UNIT - III: Mathematical Concepts
Review of probability, statistics and differential equations. Introduction to Random Variables, Probability and Cumulative distribution curves, Correlation, Covariance, Convolution of signals

UNIT - IV: Signal Processing Techniques
Fourier Transforms, Fourier Series, Z-Transforms and Laplace Transforms and their applications

UNIT - V: Digital signal Processing
Types of Filters, Signal responses to filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). Nyquist criterion, Fast Fourier Transform, Discrete Fourier Transforms, decimation-in-time (DIT) FFT, Inverse Fourier transforms, Types of window functions and frequency responses.

3. Reading Material

Text Books

1. Signals and Systems – by Simon Haykin (Author), Barry Van Veen (Author), 820 pages,
2. Fundamentals of Signals and Systems Using the Web and Matlab – by Edward W.Kamen (Author), Bonnie S Heck (Author), 632 pages, Publisher: Pearson Education (2007)

Reference Books:

1. **Communication Systems–** by Simon Haykin (Author), Michael Moher (Author), 436 pages Publisher: Wiley; Fifth edition (25 June 2009
2. **Signals and Systems** – by Oppenheim Alan V. (Author), Willsky Alan S. (Author), Nawab S. Hamid (Author), 992 pages, PHI; 2 edition (1997)

Suggested Assignments
Weekly assignments based on MATLAB programming may be provided to help students to better understand design aspects and visualize results. Assignments may be designed around concepts like generation of different types of signals, processing them using functions provided in MATLAB, A brief introduction to Simulink Tool boxes (in MATLAB) may be provided. Assignments may be designed to augment topics taught in classroom.

JAVA LAB

1. Introduction

Objective: To master the skills of programming in Core Java. The course emphasis on implementation of object oriented principles using Java language along with getting exposure to Java language features and packages.

Credits: 2

2. Course Outline

UNIT - I: To understand the basic features of Java programming language. (2 Lab Sessions)

Topics:

- Java Development Environment
- Java Fundamentals
    - Class with main function
    - Primitive Variable Types, Operators (Arithmetic, Logical, Relational)
    - Control Structures, Loops
    - Arrays

UNIT - II: Writing Classes and working with objects. (2 Lab Sessions)

Topics:

- Class, Methods, Access Modifiers, Object Creation
- Constructor, Constructor Overloading, Copy Constructor
- finalize method and Garbage Collector
- Working with frequently used Java Classes like java.lang.String, java.lang.Math

UNIT - III: Using Inheritance and Polymorphism in Java (4 Lab Sessions)

Topics:

- Writing a Sub class
    - Super key word, Constructors in Sub Class
    - Method Overriding
    - Overriding Object class methods
- Understanding Class Hierarchy: Packages
- Polymorphism
- Abstract Classes and Inheritance
- Interfaces and Inheritance
- Case Study depicting OO application design using polymorphism and inheritance
    - (for example) Developing a Solitaire Application (Chapter 8 of "Introduction to Object Oriented Programming by Timothy Budd")

UNIT - IV: Exception Handling (2 Lab Session)

Topics:

- Exception Class Hierarchy
- Exception Handling Mechanism
    - Catching an Exception, Multiple Catch Blocks, finally block
    - Throwing an exception
- Writing own exception class

UNIT - V: Exploring Java IO Package (2 to 3 Lab Sessions)

Topics:

- IO Package Class Overview
- Input/Output Stream Classes
- Readers and Writers
- File I/O
- Serialization
    - Object Input and Output Streams
- Utility of Scanner class

Unit VI: Programming Applets (2 Lab Sessions)

Topics:

- Life Cycle of Applet
- Passing Parameters to Applet
- Event Handling in Applet

Unit VII: Miscellaneous Topics

Topics:

- Generics
- Exploring java.util package
- Collections Framework

3. Reading Material

Text Books

1. Java: A Beginner's Guide, by Herbert Schildt, McGraw Hill Education (India) Private Limited.
2. Thinking in Java by Bruce Eckle, Prentice Hall.

Reference Books:

1. Java: The Complete Reference by Herbert Schildt, McGraw Hill Education.
2. Effective Java by Joshua Bloch, Createspace Independent Pub. Head First Java by Kathy Siera, O'Reilly Media

OPERATING SYSTEMS

1. Introduction

Objective: In this course, a detailed view of the kernel is given. This includes a detailed description of the way system calls work, the extended process state diagram in process management, detailed file system management description including the
data structures such as *inodes* used to maintain metadata and introduction to device drivers as part of the I/O subsystem.

This course excludes user space application development programming such as using IPC or thread-based programming. This is included in the *Network Programming* **course**.

Credits: 3-0-1

2. Prerequisites

C programming, Data and File Structures.

3. Course Outline

UNIT - I: Introduction and Operating System Structures
Operating Systems Functionality, Computer Organization and Architecture, OS Operations, Kernel Data Structures, OS Services, User interfaces to OS, Programmer interfaces to OS, OS Structure, System Boot.

UNIT - II: Process and Thread Management
Process Concept, Process operations, Process Scheduling, Extended Process State Diagram (To be done from Stallings, Operating Systems: Internals and Design Principles), Process Context Switch in detail including System Call inter- face and implementation (To be done from Crowley: Operating Systems: A design-oriented approach), Interprocess Communication: Pipes, Named Pipes, Shared Memory, Process Synchronization: Signals, Mutexes, Semaphores, Monitors (To be done from Silberschatz et al. and Stevens), Thread Management: thread creation, thread scheduling, thread synchronization, Deadlocks: Resource Allocation Graphs, deadlock detection, prevention and avoidance, recovery from deadlock**.**

UNIT - III: Memory Management
Memory allocation techniques: paging and segmentation, Swapping, structure of the page table, Virtual memory: demand paging, copy-on-write, Page replacement, allocation of frames, kernel memory allocation, thrashing, memory-mapped files, Translation-Lookaside Buffer (TLB), multiprocessor concerns.

UNIT - IV: File System Management
Disk management: formatting, boot block, swap-space management, RAID structure, Disk scheduling algorithms: elevator, C-SCAN, File concept, Access methods,Directory structure, File system mount and unmount operations, file sharing, protection, file system structure, file system implementation: file system metadata storage structures such as inode (To be done from Bach: The Design of the Unix OS), allocation methods, free space management, efficiency and performance including disk cache and recovery from failures.

UNIT - V: I/O Management

I/O devices: polling, interrupt-driven, DMA, Application I/O interface: character and block devices, network devices, clocks and timers, nonblocking and asynchronous I/O, vectored I/O, Kernel I/O interface: I/O scheduling, Buffering, Caching, Spooling and device reservation, error handling, I/O protection, Kernel data structures Transforming I/O requests to hardware operations, Performance.

UNIT - VI: Case Studies1
Linux, Unix: Solaris/AIX, Windows 7.

4. Reading Material

Text Books
Abraham Silberschatz, Peter Baer Galvin and Greg Gagne. Operating System Concepts , 9th edition, Wiley

Reference Books

1. Charles Crowley. Operating Systems: A Design-Oriented Approach, Prentice-Hall India.
2. W. Richard Stevens, . Advanced Programming in Unix Environment, Pearson Education.
3. W. Richard Stevens. Unix Network Programming, vol. 2, Pearson Education.
4. William Stallings. Operating Systems: Internals and Design Principles, Pearson Education.
5. Maurice J. Bach. The Design of the Unix Operating System, Prentice-Hall India.
6. Robert Love. Linux Kernel Development, Pearson Education.
7. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice, 2nd edition, Recursive Books.

Suggested Assignments

1. Modify the kernel to include the statement “Hello, World”, compile the kernel and modify the boot loader to add the new version.
2. Compare the popular file systems: ext3, NTFS and XFS.
3. Implement small modifications of Producer-Consumer problem.
4. Implement a user level ps command by walking through the /proc directory.
5. Implement page tables and virtual-physical address mapping using the page tables. In this, the input can consist of a file containing the processor arch. (16/32/64-bit), page size, available RAM. For each process the file will contain the size of the executable and/or page no. and permissions on it and the frame start address. Then a set of virtual addresses with the operation (r/w/x) are input to the program for which the corresponding physical address must be returned if it is a valid virtual address and also whether the operation attempted is legal as per the permissions on the page.
6. Go through the /etc/fstab and manipulate it to mount partitions with different settings and/or use mount/umount commands to mount and unmount partitions and understand the concept of logical volumes.
7. Implement a simple file system such as FAT using FUSE API.
8. Implement a simple software device driver.

INTERNET TECHNOLOGIES

1. Introduction

Credits: 3-0-0

2. Course Outline

UNIT - I: Application Layer Protocols
HTTP, Proxy Servers, SMTP, POP, IMAP, SSH, FTP, Peer-to-Peer protocols such as BitTorrent, Distributed Hash Tables..

UNIT - II: Client-Side Technologies
HTML, CSS, PHP, JavaScript, XML, Document Object Model (DOM), Dynamic Content, Cookies.

UNIT - III: Connection to Server and Server-Side Technologies
Databases (MySQL/MongoDB) and JDBC, Servlets, JSP, NodeJS..

UNIT - IV: Advanced Client-Side Technologies
Asynchronous JavaScript and XML (AJAX), JQuery, JavaScript Object Notation (JSON), Google Web Toolkit, [Ruby on Rails1], GoJS, Firebug.

UNIT - V: Advanced Web Applications
Search Engines and their algorithms, Google Maps and building your own Google Maps, Keyhole Markup Language (KML) on Google Earth.

3. Reading Material

Text Books

1. Core Servlets and Java Server Pages (JSP), by Marty Hall, Prentice Hall, 2nd edition (2003).
2. Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX by Elliotte Rusty Harold, Addison-Wesley Pub Co; 1st edition, 2002. (: Available online at http://cafeconleche.org/books/xmljava/

References

1. http://www.andrew.cmu.edu/user/mm6/95-733/schedule.html
2. http://cs.nyu.edu/courses/spring05/V22.0480-001/
3. http://web.stanford.edu/class/cs193c/lectures.html
4. http://www.sei.cmu.edu/library/assets/presentations/nelson-saturn2013.pdf
5. http://getfirebug.com/
6. http://gojs.net/latest/index.html

DATABASE MANAGEMENT SYSTEMS

1. Introduction

Objective: The course mainly covers database design and the use of databases in applications, with a short introduction to back-end structure of database systems. It includes extensive coverage of the relational model, relational algebra, and SQL. The course also features database design, and relational design principles based on dependencies and normal forms. Many additional key database topics from the design and application-building perspective are also covered: indexes, views, transactions, authorization, integrity constraints, triggers, Emerging NoSQL systems like Cassandra and MapReduce will also be covered

Credits: 3-0-0

2. Course Outline

UNIT - I: Introduction
Characteristics of the Database Approach , Advantages of Using the DBMS Approach, A Brief History of Database Applications, Data Models, Schemas, Instances, Three-Schema Architecture and Data Independence , Database Languages and Interfaces, The Database System Environment, Centralized and Client/Server Architectures for DBMS, Classification of Database Management Systems

UNIT - II: Data Modeling using the ER/EER model (If needed)
Entity Types, Entity Sets, Attributes, and Keys , Relationship Types, Relationship Sets, Roles, Structural Constraints, Weak Entity Types, ER Diagrams, Naming Conventions, and Design Issues , Relationship Types of Degree Higher than Two, Subclasses, Superclasses, and Inheritance , Specialization and Generalization, Constraints and Characteristics of Specialization and Generalization Hierarchies, Modeling of UNION Types Using Categories.

UNIT - III: Relational model (ER/EER to relational mapping)
The relational model, Relational Model Constraints, Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations, Querying relational databases, Relational Database Design Using ER-to-Relational Mapping, Mapping EER Model Constructs to Relations.

UNIT - IV: Relational Algebra
Relational Algebra operations from set theory, Binary Relational Operations: join and division, Additional Relational Operations, Examples of Queries in Relational Algebra, The Tuple Relational Calculus, he Domain Relational Calculus(Select, project, join) (Set operators, renaming, notation).

UNIT - V: Functional Dependencies & Normalisation
Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form , Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal form, Inference Rules, Equivalence, and Minimal Cover Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, About Nulls, Dangling Tuples, and Alternative Relational Designs Further Discussion of Multivalued Dependencies and 4NF, Other Dependencies and Normal Forms.

UNIT - VI: SQL
Introduction to SQL, Basic SELECT statement, Table variables and set operators, Subqueries in WHERE clause, Subequeries in FROM and SELECT, The JOIN family of operators, Aggregation, NULL values, Data modification statements, Constraints and Triggers, Constraints of several types, Referential integrity, Views, Defining and using views, View modifications – introduction, View modifications using triggers, Automatic view modifications, Materialized views.

UNIT - VII: Concurrency and Parallelism
Transactions, properties of transactions, scheduling definitions, Locking, Lock based concurrency control, using strict 2hase locking and serialisability, Deadlocks, dealock prevention/detection, multiple granularity locking.

UNIT - VIII: Indexing
Index motivation, what is stored in an index, operations on an index, datafile with several index files, high level of index structures, B-trees, Hash tables, B+trees, clustered vs unclustered index, sorting.

UNIT - IX: NoSQL, NewSQL, and Not-yet-SQL
Distributed storage using DHT's (Routing, joining, replication), NoSql overview (ACID vs BASE, Datamodel, Key value stores, Document stores, Extensible Record stores, Scalable relational systems, sort case studies-Dynamo, Cassandra)

OLAP and Cubes and Graph Processing are other topics which could be addressed if we take out ER/EER module.

3. Reading Material

Text Books
Fundamentals of Database Systems (6th edition) by Elmasri and Navathe

Reference Books
1. A First Course in Database Systems (3rd edition) by Ullman and Widom (same material and sections as Database Systems: The Complete Book (2nd edition) by Garcia-Molina, Ullman, and Widom)
2. Database Management Systems (3rd edition) by Ramakrishnan and Gehrke
3. Database System Concepts (6th edition) by Silberschatz, Korth, and Sudarshan

ALGORITHMS

1. Introduction

Objective: Learn classic algorithms designed for classic problems, analyzing an Algorithm, learn techniques to design algorithms, computational tractability (NP-completeness)

Credits: 3-0-0

2. Prerequisite

Data Structures, Discrete mathematics

3. Course Outline

Introduction to algorithms, Asymptotical Notations, Solving Recurrences, Master Theorem

Divide-and-Conquer: Strassen's Algorithm, Quicksort, Randomized Quicksort, Linear time sorting, Medians and Order Statistics

Greedy Algorithms: Activity-Selection, Kruskal , Prim

Dynamic Programming: Shortest Paths in DAG, Matrix-Chain Multiplication, Bellman-Ford, Floyd-Warshall, Dijkstra Network Flow, Ford-Fulkerson Method, Max-Flow-Min-Cut Week , Maximum Matching,

NP-Completeness, Reductions

Linear Programming, Duality

Randomized Algorithms
Approximation Algorithms

4. Reading Material

Text Books

1. Introduction to Algorithms" by Cormen et al, Third edition

References

1. " Algorithm Design" by Kleinberg and Tardos
2. Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne
3. "The Art of computer programming" by Donald Knuth

PRINCIPLES OF PROGRAMMING LANGUAGES

1. Introduction

Objective: The aim of this course is to provide students an understanding of the concepts underlying design and implementation of programming languages. The evolution of these concepts will be presented through the study of languages from FORTRAN to Java. The course will make the student understand the underlying differences and common features of procedural, functional, logic, object-oriented and scripting languages.

Credits: 3-0-0

2. Course Outline

UNIT – I
Basics of programming languages; language design and implementation issues; impact of machine architectures; what makes a good (or successful) language.

UNIT – II
Common features of programming; elementary data types; encapsulation; inheritance; sequence control.

UNIT – III
Subprogram control; storage management and run-time structures; distributed processing and network programming.

UNIT – IV
Summaries of popular proceduraland object-oriented languages;FORTRAN,C; Smalltalk,C++, Java;

UNIT – V
Summaries of popular logic and functional languages; LISP, ML; Prolog; specialised languages such asPostscript,PHP.

3. Reading Material

Text Books
1.T.W. Pratt andM.V. Zelkowitz. “Programming Languages: Design and Implementation,” 4th Edition, Prentice-Hall India (2001).

Additional Readings
1. Robert W. Sebesta. “Concepts of Programming Languages,” 10th Edition, Pearson Publishing (2009).

DBMS LAB

1. **Introduction**

Credits: 2

2. **Course Outline**

UNIT - I: ER-Modelling tools

Starting with Erwin, Adding Entity Types, Adding relationships, forward engineering, supertype/subtype example. ERWin, a popular data modeling software that allows database designers to represent Entity-Relationship diagrams and automatically generate relational SQL code to create the database in one of several commercial relational database management systems such as Oracle or Microsoft SQLServer.

UNIT - II: Abstract query languages

Creating the database, relational algebra interpreter, Relational Algebra Syntax, Naming of Intermediate Relations and Attributes, relational algebraic operator supported by the RA-Interpreter, Examples, Domain Relational Calculus Syntax, Safe DRC Queries, Datalog interpreter, datalog query examples. The three interpreters that can be used to execute queries in Relational Algebra, Domain Relational Calculus, and Datalog. These interpreters are part of a Java package that includes a rudimentary database engine capable of storing relations and able to perform basic relational algebraic operations on these relations. It is hoped that these interpreters will allow the student to get a better understanding of abstract query languages.

UNIT - III: Relational Database management System- MYSQL

Company Database, MySql utility, MySql and PHP programming, Online address book. Programming Web applications in PhP that accesses MySQL databases is introduced with a complete database browser application for the COMPANY database as well as a complete Online Address Book application.

UNIT - IV: Database design (DBD) toolkit

The toolkit allows the student to work with numerous concepts and algorithms that deal with functional dependency theory and data normalization. The student may use DBD to verify answers to many questions related to functional dependency theory and normalization algorithms.

UNIT - V: Object-oriented database management Systems: DB4O

A popular open source Object-Oriented Database Management system. Creating and populating objects in db4o is covered as well various methods to query and retrieve data from the object-oriented database is introduced.

UNIT - VI: XML

XML and its related technologies will be taught. Query languages XPath and XQuery are covered as well as schema specification language XML Schema.

Projects

Projects may be implemented in Java, PhP or any other favorite programming language and may access Oracle, MySQL or any other relational database management system.

SOFTWARE ENGINEERING

1. Introduction

Objective: Intends to introduce software engineering principles and techniques required for automation and quality production of software systems

Credits: 3-0-0

2. Course Outline

UNIT - I: Introduction
Software Development - an engineering process, Short History of Software Engineering, Software Development Life Cycle, Engineers in Software Development Cycle, Systems Engineering, Software Engineering and other branches of study, Identifying Issues in Software Engineering.
Defining Software: Its Nature and Qualities: Representative Qualities, Types of Systems and Quality Requirements, Software Engineering and maintenance of qualities.
Software Engineering Principles: Rigor and Formality, Separation of Concerns, Modularity, Abstraction, Change Management, Incrementality, Generality

UNIT - II: Requirement Engineering
Software requirements, Requirement Engineering Process, Software Requirements Specification (SRS), Techniques for Requirements Elicitation, Specifying Software Requirements, Formal and Rigour based Specification: Use cases, Activity Diagrams, FSM, Petri Nets, Algebraic And Logic Based Specifications.

UNIT - III: UML-based Design
UML Based Object-Oriented Design: Class, Sequence, State, Architectural and Deployment Diagrams.
Design with Reuse: Introducing Design Patterns
User-Interface Design: Aesthetic and Technical Issues.
Concerns Real-Time System Design
Modularization
Software Architectural Design, Software Design Document

UNIT - IV: Verification
Requirement Verification: Completeness and Correctness of Software Requirements. Introducing Model Checking and Symbolic Execution, Software Test Planning
Software Testing: Static and Dynamic Testing Techniques, Levels of Testing.
Testing a Process: Introducing Tools to Automate Testing

UNIT - V: Project Management
Software Production Process, Overview of Software Process Models,
Cost Estimation, Project Management, Risk Management, Quality Management:
CMM: Capability Maturity Model, Agile Method. Software Project Management Plan (SPMP)

UNIT - VI: Tools
Introducing Software Engineering Tools, Diagramming Tool, Analysis Tool, Code Generation Tool, Testing Tool, Management Tool, Process Support Tool, General Idea on Tool Development Technique

UNIT - VII: Evolution
Legacy Systems, Software Reverse Engineering, Re-engineering, Configuration Management

3. Reading Material

Text Books

1. Fundamentals of Software Engineering; Carlo Ghezzi, Mehdi Jazayeri, Dine Madrioli, Pearson Education Asia, Low Priced Edition.

Reference Books

1. Software Engineering, Ian Sommerville; Pearson Education Asia, Low Priced Edition.
2. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson; Pearson Education Asia, Low Priced Edition.
3. Fundamentals of Software Testing, Aditya P Mathur; Pearson.

Suggested Assignments

1. List the qualities software should have and explain the usefulness of each clearly.
2. Discuss on requirement Engineering Process
3. List the techniques for requirement elicitation.
4. Define Module, relations. And show use of relations uses, isComposedOf in giving modular architecture of a software system. (use TDN and graphic design notations)
5. Justify use of models: FSM and PetriNet With examples illustrate your view point.
6. Write the steps to be followed for writing algebraic formal specification.
7. Write the steps to be followed for writing logic-based formal specification.
8. Present a system architecture for a tool to automate Usecase based requirement engineering.
9. "Traceability is a quality maintainable by engineering principle." Justify the statement.
10. Derive test cases from a state diagram.

1. **Introduction**
   Credits: 3-0-0
2. **Prerequisite**
   C programming, Operating Systems.
3. **Course Outline**

UNIT - I: Physical Layer
Modulation Techniques: Amplitude, Frequency and Phase, ADSL, Multiplexing Techniques: Frequency division multiplexing, Time division multiplexing, wave length division multiplexing, Differential PCM, Switching Techniques: Circuit, message and packet switching.

UNIT - II: Data Link Layer
PPP, PPPoE, MAC Layer: Ethernet (incl. manchester encoding), Switched Ethernet, VLANs, Spanning Tree Protocol.

UNIT - III: Network Layer: Data Plane
Internet Protocol Addressing: CIDR, Internet Protocol Datagram (including fragmentation and reassembly, routing options), IP Forwarding Algorithm, ARP, ICMP (including ICMP Redirect, ICMP Path MTU discovery, ICMP Destination Unreachable options).

UNIT - IV: Transport Layer
UDP, TCP sliding window protocol, TCP connection establishment, TCP reliability including cumulative and delayed acknowledgements, Nagle algorithm, Karn's algorithm for RTT and RTO estimation, TCP AIMD Congestion Control Algorithm, TCP half-close connections including TCP keepalive timer and probe timer, TCP Fast Retransmit and Fast Recovery.

UNIT - V: Network Layer: Control Plane
Distance Vector Algorithm and Routing Information Protocols V1 and V2, Link State Algorithm and Open Shortest Path First Protocol (OSPF).

UNIT - VI: Application Layer
Domain Naming System (DNS) and Dynamic Host Configuration Protocol (DHCP), Network Management using SNMP.

4. **Reading Material**

Text Books

1. James F. Kurose and Keith W. Ross. Computer Networking: A top-down approach, 6th edition, Pearson Education..

Reference Books

1. Douglas Comer. Computer Networks And Internets Sixth Edition, 2014. ISBN 0133587932/9780133587937, Pearson Education.
2. Douglas Comer. Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 6th edition, 2013. ISBN-10: 0-13-608530-X ISBN-13: 9780136085300, Pearson Education.
3. Kevin R. Fall and W.Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols, 2/E, 2012, ISBN-10: 0321336313 ISBN-13: 9780321336316, Pearson Education.
4. Radia Perlman. Interconnections: Bridges, Routers, Switches, and Internetworking Protocols, 2/E, 2000, ISBN-10: 0201634481 ISBN-13: 9780201634488.Pearson Education.

Suggested Assignments

1. Implement the IP fragmentation and reassembly algorithm.
2. Implement the IP forwarding algorithm.
3. Implement the simplest sliding window protocol of TCP.
4. Connect two systems using a switch and configure private IP addresses to the systems and ping them from each other. Using Wireshark, capture packets and analyze all the header information in the packets captured.
5. Convert a system with two network interface cards (NICs) into a router by configuring each NIC in a different LAN and enabling forwarding. Use two switches to connect one NIC each of the router to these two switches. Connect two other systems, one each to each switch. Now, we have two VLANs. Ping from one system to the other through the router after configuring the required default routes in the hosts and static routes in the router.
6. Install Telnet Server on one of the systems connected by a switch and telnet to it from the other system. Using Wireshark, capture the packets and analyze the TCP 3-way Handshake for connection establishment and tear down.
7. Use Mininet to create multiple compute nodes, multiple VMs in each compute node, an OVS to connect VMs in a compute node and one OVS that connects the nodes. Configure IP addresses to these VMs in different VLANs, configure static routes in the VMs and ping them from one another.

8. Once the initial Mininet assignment is successful, try commands like traceroute to trace the route from one VM to another.

COMPUTER GRAPHICS

1. Introduction

Objective: This being the first course in computer graphics introduces the basics of the subject.

Credits: 3-0-1

2. Course Outline

UNIT - I: Introduction
History, Advantages, Applications, Graphics I/O Devices, Raster Graphics, Graphics Packages and Libraries, Line and Circle Drawing Algorithms, Scan Conversion, Polygon Filling.

UNIT - II: Geometric Transformations and Clipping
2D Transformations, Homogeneous Coordinate System, 3D Transformations, Plane Geometric Projections, Viewing Transformations, Line and Polygon Clipping..

UNIT - III: Curves and Surfaces
Parametric Representation of Curves, Cubic Splines, Bezier Curves, B-Splines, Parametric Surfaces, Surfaces of Revolution, Sweep Surfaces, Quadric Surfaces, Fractal Curves and Surfaces.

UNIT - IV: Realism in 3D Graphics
Hidden Line and Hidden Surface Removal Algorithms, Illumination Models, Phong and Gouraud Shading.

3. Reading Material

Text Books

1. D. F. Rogers: Procedural Elements for Computer Graphics, Tata McGraw Hill
2. D. F. Rogers and J.A. Adams: Mathematical Elements for Computer Graphics, Tata McGraw Hill
3. J. D. Foley, A. VanDam, S.K. Feiner. and J.F. Hughes: Computer Graphics: Principle and Practice, Pearson Education.
4. Z. Xiang and R. Plastock: Computer Graphics, Tata McGraw Hilledition, Pearson Education.

Reference Books

1. E. Angel: OpenGL - A Primer, Pearson Education
2. D. Shreiner, G. Sellers,J. Kessenich, B. Licea-Kane: OpenGL Programming Guide, Pearson Education.
3. T. McReynolds and D. Blythe: Advanced Graphics Programming Using OpenGL, Elsevier
4. Nehe OpenGL Tutorials @ http://nehe.gamedev.net..

Suggested Assignments

Programming Assignments Based on Line and Circle Drawing, Polygon Filling Algorithms, 2D, 3D & Viewing Transformations, Fractal Curves, Hidden Line/Hidden Surface

COMPILER DESIGN

1. Introduction

Objective: To teach basics of compiler principles and design.

Credits: 3-0-1

2. Course Outline

UNIT - I: Compiler Structure
Analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

UNIT - II: Lexical Analysis
Interface with input program, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, implementation, regular definition, transition diagrams.

UNIT - III: Syntax Analysis
CFGs, ambiguity, error detection and recovery, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing, LR parsers (SLR, LALR, LR).

UNIT - IV: Syntax Directed Translation
Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions

UNIT - V : Type Checking
Type systems, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

UNIT - VI : Run time Environments
Source language issues, storage organization, activation tree, activation record, stack allocation of activation records, parameter passing mechanisms, symbol tables, dynamic storage allocation techniques.

UNIT - VII : Intermediate Code Generation
Intermediate representations, translation of declarations, assignments, control flow, boolean expressions and procedure calls. Implementation issues.

UNIT - VIII : Code Generation and Instruction Selection
Issues, basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from DAGs, peep hole optimization.

UNIT - IX : Code Optimizations
Principal Sources of Optimization, Optimization of basic blocks, Introduction to Dataflow Analysis (Reaching Definitions and Live Variable Analysis).

3. Reading Material

Text Books

1. AV Aho, MS Lam, R Sethi, JD Ullman: Compiler Design: Principles, Techniques and Tools, Pearson Education

Reference Books

1. AW Appel, M Ginsburg: Modern Compiler Implementation in C, Cambridge University Press
2. K Cooper, L Torczon: Engineering a Compiler, Morgan Kaufmann
3. J.P. Tremblay, P.G. Sorenson: Theory and Practice of Compiler Writing, McGraw Hill

Suggested Assignments

Programming assignments based on lexical analysis, construction of predictive/operator precedence/SLR parsing table and parser, symbol table, dynamic storage allocation strategies, syntax directed translation, data flow analysis

1. Introduction

Objective: Develop an Application using tool.
The Lab includes a project component wherein the students have to put into practice the software engineering methods learned to develop a software solution to a given problem by performing all the SDLC activities viz. requirements analysis & design by using appropriate CASE tools, coding, testing, project management and configuration management along with doing the necessary technical documentation based on standard templates.

Credits: 2

2. Course Outline

UNIT - I: SRS (deliverable: SRS document)
UNIT - II: High Level Design (deliverable: HLD document)
UNIT - III: Detailed Design (deliverable: DD document)
UNIT – IV: Unit test case and integrated test case design
(Deliverables: UTP and ITP documents)
UNIT - V Implementation (deliverable: working demo of the code)

3. Reading Material

Suggested Projects

- Online Event Management System
- Online Health care System
- Vehicle Tracking System
- E-Governance application

References

1. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=278253
2. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5981339
3. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4578383
4. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5167255
5. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1488512
6. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=741937

NETWORK PROGRAMMING

1. **Introduction**

Objective: Sockets provide the application layer level interface for two systems to communicate over Network. The primary objective of the course is to learn basics and advanced techniques of socket based client server programming. Berkeley Socket API based network programming using C language in Unix environment is taught in the course.

Credits: 3-0-0

2. **Prerequisite**

Unix System Programming (Covered as part of OS Course), C Programming.

3. **Course Outline**

UNIT – I (Ten Hours):

To review Unix System Programming and OS concepts relevant for Network Programming.

Topics:

- Process Control (fork, vfork, wait, exec system calls, user ids and related system calls),
- Reliable Signal Implementation (signal, sigaction, sigprocmask, sigsuspend, sigpending system calls, handling SIGCHLD)
- Pthreads

UNIT – II (Two Hours):

Overview of TCP/IP Protocol Suite and Internet Applications

Topics:

- Overview of TCP/IP Protocols
  - TCP (State Transitions)
  - UDP
  - IP
- Popular Internet Applications and their protocol usage

UNIT – III (Three Hours):

For introducing fundamentals of sockets and common functions used in any Client/Server application.

Topics:

- Socket
  - Introduction, role as an application programming interface
  - Address Structure and address management functions for IPv4, IPv6
- Common functions
  - Byte Ordering, byte manipulation functions
  - readn, writen, readline functions

UNIT – IV (Five Hours):

Basics of TCP Client/Server application development

Topics:

- Client/Server Paradigm
- Iterative TCP Server
  - Socket, Bind, Listen, Accept, read, write, close system calls
- TCP Client
  - Socket, Bind, Connect, read, write, close system calls
- Concurrent TCP Server
  - Using fork
  - Using pthreads
- Example of TCP Client/Server Application

UNIT - V (Three Hours) :

Basics of UDP Client/Server application development

Topics:

- UDP Server and UDP Client

    - recvfrom, sendto system calls
- connect system call in UDP Client and asynchronous errors

UNIT - VI (8 Hours) :
Miscellaneous topics for building more sophisticated client/server applications
Topics:

- Data representation issues
- I/O Multiplexing
    - select or poll system call
    - Design of TCP, UDP Servers using select or poll system call
- Socket Options
    - getsockopt, setsockopt system calls
    - Few examples of SOCKET, TCP, UDP, IP options and their role in client/server applications
    - Using fcntl
- DNS related functions

UNIT - VII (4 Hours):
Understanding the mechanisms for logging messages, making a server as a daemon server and developing server programs to be used by inetd super server.
Topics:

- syslogd server and syslog system call
- Daemon server using daemon_init function
- inetd super server

UNIT - VIII (4 Hours):
Debugging client server applications and understanding an implementation of application protocol
Topics:

- Tools for debugging TCP, UDP applications
    - tcpdump, tcpflow, netstat, ethereal
- Detailed analysis of implementation of an application layer protocol's client and server programs (like HTTP Server, HTTP Client)

.

UNIT - IX :
Advanced Network Programming topics (optional)
Topics;

- Broadcasting
- Multicasting
- Routing Sockets
- Raw Sockets
- Data link access and design of packet filters.

4. Reading Material

Text Books

1. Richard Stevens: "Advanced Programming in Unix Environement", Pearson Education Asia. (For review topics)
2. Richard Stevens: "Unix Network Programming Volume I (Networking APIs: Sockets and XTI)", Pearson Education Asia.

Reference Books

1. Douglas E. Comer and David L. Stevens: "Internetworking with TCP/IP Volume III: Client-Server Programming and Applications, Linux/POSIX Sockets Version", Prentice Hall.

ESSENTIALS OF AI

1. Introduction

Objective: To have an appreciation for and understanding of both the achievements of AI and the theory underlying those achievements. To have an appreciation for the engineering issues underlying the design of AI systems. To have a basic proficiency in a traditional AI language including an ability to write simple to intermediate programs and an ability to understand code written in that language. To have an understanding of the basic issues of knowledge representation and blind and heuristic search, as well as an understanding of other topics such as minimax, resolution, etc. that play an important role in AI programs.

Course Description: This course will introduce the basic principles in artificial intelligence research. It will cover simple representation schemes, problem solving paradigms, constraint propagation, and search strategies. Areas of application such as knowledge representation, natural language processing, expert systems, vision and robotics will be explored.

Credits: 3-0-0

2. Course Outline

UNIT - I: Introduction
What is AI, Foundations of AI, History of AI.

UNIT - II: Intelligent Agents
Agents and Environments, Structure of Agents.

UNIT - III: Problem Solving by Searching
Problem Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-First Search, Depth-First Search, Depth-limited Search, Iterative Deepening Depth-first Search, Comparison of Uninformed Search Strategies.

UNIT - IV: Informed Search and Exploration
Informed (Heuristic) Search Strategies: Greedy Best-first Search, A* Search, Heuristic Functions, Local Search Algorithms and Optimization Problems.

UNIT - V : Constraint Satisfaction Problems
Backtracking Search for CSPs, Local Search for CSPs.

UNIT - VI : Adversarial Search
Games, Minimax Algorithm, Alpha-Beta Pruning

UNIT - VII : Reasoning and Knowledge Representation
Introduction to Reasoning and Knowledge Representation, Propositional Logic, First-order Logic, Semantic Nets, Other Knowledge Representation Schemes.

UNIT - VIII : Automated Planning
Planning with state-space search, partial-order planning, planning graphs, planning iwth propositional logic

UNIT - IX : Reasoning with Uncertainty & Probabilistic Reasoning
Acting Under Uncertainty, Bayes' Rule, Representing Knowledge in an Uncertain Domain, Bayesian Networks.

UNIT - X : Making simple & complex decisions
Hidden Markov Models, Utility and decision theory, Markov Decision Processes (MDP), Partially Observable Markov Decision Processes (POMDP)

3. Reading Material

Text Books

1. Artificial Intelligence: A Modern Approach, 2nd edition, by Russell and Norvig, Prentice Hall

References

1. What is AI? (by John McCarthy)
2. History and Promise of AI (by David Waltz)
3. The Human Intelligence Enterprise. Why I am Optimist (by Patrick Winston)

ADVANCED COMPUTER ARCHITECTURE

1. Introduction

Credits: 3-0-0

2. Prerequisite

Computer Organization, Basics of Computer Architecture, Operating Systems.

3. Course Outline

UNIT - I: Fundamentals of Quantitative Design and Analysis

Classes of computers, Trends in technology, Measuring and Reporting Performance, Power consumption and efficiency as the metric, Dependability, Quantitative Principles of Computer Design.

UNIT - II: Instruction Set Principles

Classifying Instruction Set Architectures, Memory Addressing, Addressing modes, Operations in the instruction set, Instructions for control flow, encoding an instruction set, Role of compilers.

UNIT - III: Memory Hierarchy Design

Cache performance review, Four basic memory hierarchy questions, Six basic optimizations of cache performance, Ten advanced optimizations of cache performance, Protection: virtual memory and virtual machines, memory technology and optimizations: SRAM, DRAM, Flash memory, Graphics Data RAMs.

UNIT - IV: Instruction-Level Parallelism and its Dynamic Exploitation

Instruction level parallelism: concepts and challenges, basics of pipelining, data hazards, structural hazards, control hazards, minimizing data hazards through forwarding, overcoming branch penalties by delayed branches, static and dynamic branch prediction, dealing with exceptions in pipelining, dynamic scheduling, Tomasulo's algorithm, speculative processors, high performance instruction delivery, VLIW approach, static and dynamic superscalar processors.

UNIT - V : Multiprocessors and Thread-Level Parallelism

Multithreading: exploiting thread-level parallelism within a processor, Symmetric shared-memory architectures and their performance, Distributed shared-memory architectures and their performance, synchronization, models of memory consistency.

4. Reading Material

Text Books

1. John L. Hennessey and David A. Patterson. Computer Architecture: A Quantitative Approach, 5E, Morgan-Kaufmann, 2012. ISBN-13: 978-0123838728.

Suggested Assignments

Assignments can primarily be exercises from Hennessy and Patterson with modifi-
cations to them.

1. Problems involving use of Amdahl's law.
2. Problems that use the timing diagram of pipelining to estimate the number of cycles taken for execution of a set of instructions.
3. Problems involving branch prediction and branch target buffers and estimating cycles taken for instruction with and without these optimizations.
4. Problems involving memory operations and improvement in performance through use of different cache optimizations.
5. Problems involving use of dynamic scheduling and speculation to compare the performance of the system with these optimizations versus regular pipelining.
6. Problems involving the centralized and distributed shared-memory architectures and operations involved in accessing different memory locations present in multiple processors.

VIRTUALIZATION

1. Introduction

Objective: This is an advanced course and suggested reading is de_ned in the form of de_nitive research papers for every unit. Virtualization technologies allow decoupling of the user-perceived experience of hardware/software systems from a physical implementation. Presently every aspect of computing resources is being seen in with view of abstraction. Techniques exist to virtualize processing, networking, and storage. These are becoming pervasive in industry and are being used at a foundation level. The combination of virtualization technologies and ubiquitous network connectivity allows for the creation of virtual computers where processing, data and communication are distributed and decou-pled from physical resources. This course is aimed to cover the basic mechanisms and techniques involved in resource virtualization from individual machines to vir tualized networked infrastructures. This course aims to give students an exposure of the theory and implementation aspects of virtualization.

Credits: 3-0-0

2. Prerequisite

Computer Architecture, Operating Systems, Computer Networks

3. Course Outline

UNIT - I: Introduction

Why Virtualization? Trends in Virtualization. Formal Requirements for Virtualizable Third Generation Architectures (Popek and Goldberg, CACM, 1974).

UNIT - II: VMWare Workstation Design

1. Bringing Virtualization to the x86 Architecture with the Original VMware Workstation (Bugnion, Devine, Rosenblum, Sugerman, and Wang, TOCS, 2012
2. Disco: Running Commodity Operating Systems on Scalable Multiprocessors (Bugnion, Devine, and Rosenblum, SOSP, 1997).[Optional Reading]
3. The Evolution of an x86 Virtual Machine Monitor (Agesen, Garthwaite, Sheldon, and Subrahmanyan, OSR, 2010) [Optional Reading]

UNIT - III: x86 Virtualization Hardware

1. A comparison of software and hardware techniques for x86 virtualization (Adamsand Agesen, ASPLOS, 2006).

UNIT - IV: Binary Translation

1. Fast Dynamic Binary Translation for the Kernel (Kedia and Bansal, SOSP 2013)
2. Comprehensive Kernel Instrumentation via Dynamic Binary Translation (Feiner, Brown, and Goel, ASPLOS'12)
3. QEMU, a Fast and Portable Dynamic Translator (Fabrice Bellard, USENIX Annual Technical Conference 2005.)

UNIT - V: Paravirtualization, Virtualizing the VMM, KVM

1. Xen and Art of Virtualization (Barham, Dragovic, Fraser, Hand, Harris, Ho, Neugebaur, Pratt and War_eld, SOSP 2003).
2. The Turtles Project: Design and Implementation of Nested Virtualization (Ben-Yehuda, Day, Dubitzky, Factor, Har'El, Gordon, Liguori, Wasserman, and Yassour, OSDI 2010)
3. kvm: the Linux Virual Machine Monitor (Kivity, Kamay, Laor, Lublin, and Liguori, OLS 2007)

UNIT - VI: Memory and I/O Virtualization

1. AMD-V Nested Paging (AMD White Paper, July 2008)
2. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor (Sugerman et al. USENIX 2001)
3. virtio: Towards a De-Facto Standard For Virtual I/O Devices (Russell, OLS 2008)
4. Safe device assignment with VFIO (Corbet, LWN 2012)
5. ELI: Bare-Metal Performance for I/O Virtualization (Gordon, Amit, Har'El, Ben-Yehuda, Landau, Schuster, and Tsafrir, ASPLOS 2012)
6. vIOMMU: E_cient IOMMU Emulation (Amit, Ben-Yehuda, Tsafrir, and Schuster, USENIX 2011) [Optional Reading]

UNIT - VII: Virtualization-Aware File Systems and Virtual Disks

1. Virtualization Aware File Systems: Getting Beyond the Limitations of Virtual Disks (Pfa_, Gar_nkel, and Rosenblum, NSDI 2006)
2. Parallax: Virtual Disks for Virtual Machines (Meyer, Aggarwal, Cully, Lefeb-vre, Feeley, Hutchinson, and War_eld, EuroSys 2008)

3. IOFlow: A Software-De_ned Storage Architecture (Thereska, Ballani, O'Shea, Karagiannis, Rowstron, Talpey, Black, and Zhu, SOSP 2013) [Optional Reading]

4. Reading Material

Text Books

Virtualization Essentials, Matthew Portnoy, Copyright 2012 by John Wiley Sons, Inc., Indianapolis, Indiana, ISBN: 978-1-118-17671-9

References

1. Virtual Machines, Versatile Platforms for Systems and Processes James E. Smith and Ravi Nair, ISBN: 978-1-55860-910-5, 2005 Morgan Kaufman
2. Don Porter's Course on Virtualization at Stonybrook University http://www.cs.stonybrook.edu/ porter/courses/cse591/s14/
3. Topics in Virtualization and Cloud Computing at IIT Bombay http://www.cse.iitb.ac.in/ puru/courses/autumn12/cs695/lectureplan.html
4. Virtualization and Cloud Computing: I and II at IIT Delhi http://www.cse.iitd.ernet.in/ sbansal/csl862-virt/readings.html http://www.cse.iitd.ernet.in/ sbansal/csl862-virt/2010/