Courses (CSE)

Semester - I

CSE

	L	T	P	C
Introduction to Discrete Mathematics	3	1	0	4
Physics	3	1	2	5
Introduction to Programming	2	0	0	2
Introduction to Programming Lab	0	1	4	3
Digital Logic Design	3	0	4	5
Spoken and Written Communication	2	0	0	2
Total Credit				**21**

Semester - II

CSE

	L	T	P	C
Calculus	3	1	0	4
Data Structure	3	0	0	3
Data Structure Lab	0	1	4	3
Basic Electronics Circuits	3	1	2	5
Computer Organization	3	0	4	5
Introduction to Computer Science	1	0	2	2
Total Credit				**22**

Semester - III

CSE

	L	T	P	C
Object Oriented Design & Programming	3	0	0	3
Object Oriented Design & Programming LAB	0	1	4	3
Operating system	3	0	4	5
Systems and Signal Theory	3	0	2	4
Probability and Statistics	3	1	0	4
Economics	3	0	0	3
Total Credit				**22**

Semester - IV

CSE

	L	T	P	C
Database Management System	3	0	0	3
Database Management System LAB	0	1	4	3
Computer Networks	3	0	0	3
Computer Networks LAB	0	1	4	3
Design and Analysis of Algorithm	3	0	2	4
Science Technology and Society	3	0	0	3
Technical Writing	1	0	4	3
Total Credit				**22**

Summer-I

CSE

	L	T	P	C
Design Project*	0	0	6	3

*Pass/Fail

Semester - V

CSE

	L	T	P	C
Principles of Programming Language	3	0	0	3
Formal Language & Automata Theory	3	0	2	4
Software Engineering	3	0	0	3
Software Engineering LAB	0	1	4	3
Numerical Linear Algebra	3	0	0	3
Environmental Science	3	0	0	3
(SC) Elective - 1	3	0	0	3
Total Credit				**22**

Semester - VI

CSE

	L	T	P	C
Complier Design	3	0	0	3
Complier Design LAB	0	1	4	3
Introduction to Artificial Intelligence	3	0	2	4
Computer Architecture	3	0	2	4
Introduction to Cryptography & Security	3	0	0	3
(SC) Elective - 2	3	0	0	3
(HM) Elective - 1	3	0	0	3
Total Credit				**23**

Summer-II				
CSE				
	L	T	P	C
Research / Industrial Internship*	0	0	6	3

*Pass/Fail

Semester - VII				
CSE				
	L	T	P	C
Professional Ethics	2	0	0	2
Distributed Computing	3	0	2	4
(TE) Elective - 1	3	0	2	4
(TE) Elective - 2	3	0	2	4
(TE) Elective - 3	3	0	2	4
(TE) Elective - 4	3	0	0	3
Total Credit				**21**

Semester - VIII				
CSE				
	L	T	P	C
Project	0	0	36	18
Total Credit				**18**

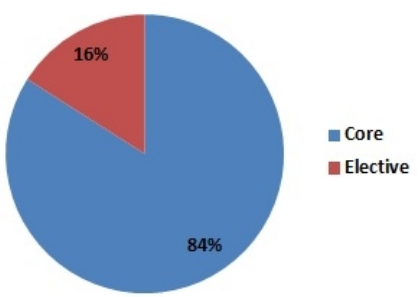
Possible Electives

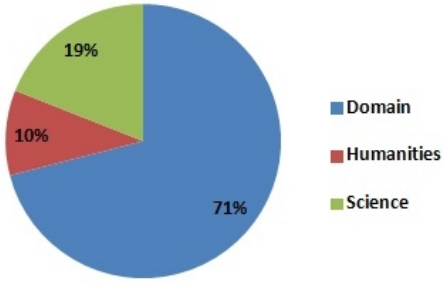
Category	Subjects
(HM) Elective - 1	Introduction to Business and Finance, Principles of management, Approaches to Indian society, Rural Development, Organizational behavior, Science Fiction, Film appreciation, Indian constitution
(SC) Elective - 1	Numerical methods, Optimization, Graph theory, Game theory, Algebraic structure, Bioinformatics, Coding Theory
(SC) Elective - 2	Dynamical systems, Modeling and simulations, Stochastic simulation, Nano science, Quantum computer, Logic for Computer Science
(TE) Elective -1	Model of Computation, Data analytics, Big Data, Computer vision, Speech processing, Wireless Sensor Networks
(TE) Elective -2	Robotics, Image processing, Management Information Systems, Digital rights management, Machine Learning,
(TE) Elective -3	Information Retrieval, Dynamics of Animation, Biotechnology, Service Oriented Computing
(TE) Elective -4	Pattern Recognition, soft Computing, Human computer Interaction, E-commerce, Information Security, web Technology

Structure of the Curriculum

- Total Credits requirement – 153 for graduation
- Credits requirement – 24 for Internship and Projects.
- Design (Summer) project after Semester IV (credit not counted for graduation requirement. It is a pass/Fail course). Students are expected to carry out a development (software/hardware) project preferably in a group of 2 – 4 students. Duration would be 6 – 8 weeks.
- Research or Industrial Internship in summer after Semester VI (credit not counted for graduation requirement. It is a Pass/Fail course). This is individual internship. Students can choose either research within/outside institute or can choose an industry internship. Typical duration would be 6-8 weeks.
- One full semester project (credit not counted for graduation requirement. This credit would be considered for computation of CPI). This is enabling students to take project in industry or any other research organization.

- 19% of the total credit is for science courses
- 10% of the total credit is for humanities courses
- Science elective 2 in numbers
- Humanities elective 1 in numbers





Semester - I

<u>Introduction to Discrete Mathematics (3-1-0-4)</u>

Course Contents:

FOUNDATION: Propositional and predicate logic, logical equivalences, predicates and quantifiers,
translation from language to logical expressions, nested quantifiers, set theory, set operations, set identities and functions, inverse and composition functions, graph of functions.

NUMBER THEORY: Division operator, prime factorization, properties of prime numbers, prime number theorem, GCD and LCM, modular arithmetic and applications, sequences and summations.

COUNTING: Permutation and combinations, pigeonhole principle, inclusion-exclusion principle, binomial theorem, Pascal identity and triangle.

MATHEMATICAL REASONING and INDUCTION: Rules of inference, direct proof, proof by contradiction, proof by contrapositive, mathematical induction and second law of mathematical induction.

RECURSION: Definition, recursive algorithm, recurrence relations, solving recurrence relations.

RELATIONS: Relations and their properties, applications and representations, equivalence relations, partial ordering, Hasse diagram.

GRAPHS: Introduction and terminology, representation, isomorphism, connectivity, Warshall‘s algorithm, Euler and Hamilton path, shortest path.

TEXT BOOK:
Discrete Mathematics and its Application, 7th Ed, K. Rosen, Tata McGraw Hill, 2011.

REFERENCE BOOK:
1) Discrete Mathematical Structure, 4th Ed, B. Kolman, R.C. Busby and S. C. Ross, PHI, 2000.
2) Discrete Mathematics, Richard Johnsonbaugh, Prentice Hall, 2007.
3) Mathematics: A Discrete Introduction, 3rd Ed., Edward R. Scheinerman, Cengage Learning, 2006.
4) Mathematical Structure for Computer Science, 6th Ed, J. Gersting, Freeman, 2006.

Physics (3-1-2-5)

Course Contents:

COORDINATE SYSTEM: Cartesian, cylindrical and spherical coordinates; unit vectors and their time derivatives.

REVIEW OF PARTICLE DYNAMICS: Inertial and non-inertial frames of reference, centrifugal and coriolis forces; conservative force, work-energy theorem; centre of mass, conservation of momentum; collision in one and two dimensions. small oscillations, free, forced and damped oscillations.

ATOMIC PHYSICS: Rutherford and Bohr's atomic model, quantum numbers, atomic spectra, energy levels.

ELEMENTARY PARTICLES: Nuclear model, protons and neutrons, nuclear force, introduction of elementary particles.

CONCEPTUAL FOUNDATION of MODERN PHYSICS: Electromagnetic waves, blackbody radiation, Planck’s law of radiation, photoelectric effect, wave-particle duality, Compton wavelength, de-Broglie wave, Heisenberg’s uncertainty principle, contribution of Dirac, Pauli, Schrodinger and Born in foundation of quantum mechanics, topics in Quantum Mechanics.

INTRODUCTION TO APPLIED PHYSICS: A non-mathematical exposure to applied physics such as:
pendulum, heat engine, transformer, optical microscope, electron microscope, scanning tunneling microscope, laser diode, photo detector, solar cells, transistors.

TEXT BOOK:
Concepts of Modern Physics. A.Beiser, Tata McGraw-Hill, New Delhi, 1995.

REFERENCE BOOK:
1) Quantum Physics of Atoms, Molecules, Solids, Nuclei and Particles, 2ndEd, R. Eisberg and R. Resnick , John-Wiley, 1985.
2) Quantum Mechanics: Theory and Applications 5thEd, AjoyGhatak, Macmillan, 2004.

Introduction to Programming (2-0-0-2)

Course contents:

INTRODUCTION TO PROGRAMMING: Programming methods, paradigms, problem solving techniques, algorithm development, flow charts.

LINUX ENVIRONMENT: Editor, compiler, debugger.

BASICS OF PROCEDURAL PROGRAMMING: Constants, variables, expressions, operators, assignment, basic input and output, built-in functions, program debugging.

VARIABLES AND OPERATORS: Basic data types, precedence and order of evaluation, pointers, memory allocation of variables.

CONTROL STRUCTURES: Selection statements, iteration statements.

FUNCTIONS AND PROGRAM STRUCTURE: Return values, actual and formal parameters, parameter passing: call by value versus call by reference, external variables, scope rules, header files, and recursion.

ARRAYS: Character arrays, one and two dimensional arrays; pointer arrays, command-line arguments.

I/O: ASCII data files, file pointers, end-of-file.

BASIC DATA STRUCTURES: Structures, defining new types, enumerations, dynamic memory allocation, dynamic arrays, linked lists and other pointer-based structures.

TEXT BOOK:
C How to Program, 6thEd, P Deitel and H Deitel, Prentice Hall of India, 2010.

REFERENCE BOOK:
1) C programming language, 2ndEd, Kernighan, Brian W. & Ritchie, Dennis M, New Delhi. Prentice Hall of India, 1998.
2) A Book on C, 4thEd, Kelley, A.L. and Pohl Ira, Pearson India, 2002
3) A Structured Programming Approach Using C, 1stEd.,Forouzan, Behrouz, Course Technology, 2012.
4) Practical C Programming, 3rdEd, Oualline, Steve, Shroff Publishers, 2000.
5) C programming: The essentials for engineering and scientists, Brooks, David R. New York. Springer, 1999.

Introduction to Programming Lab (0-1-4-3)

Course contents:
Lab and take home assignments based on the course “Introduction to Programming”.

Digital Logic Design (3-0-4-5)

Course Contents:

NUMBER SYSTEMS: Representations, signed, 1's complement, 2's complement, saturation and overflow in fixed point arithmetic.

BOOLEAN ALGEBRA: Axioms and theorems, DeMorgan’s law, universal gate, duality, expression manipulation using axioms and theorems.

COMBINATIONAL LOGIC: Introduction to switching algebra, canonical forms, two-level simplification, boolean cube, logic minimization using K-map method, QuineMcCluskey tabular method, minimization for product-of-sum form, minimization for sum-of-product form, multiplexers, demultiplexers, decoders, encoders, hazard free synthesis, Arithmetic circuits, adders, half adder, full adder, BCD adder, ripple carry adder, carry-lookahead adder, combinational multiplier.

SEQUENTIAL LOGIC: Simple circuits with feedback, basic latches, clocks, R-S latch, master-slave latch, J-K flip flop, T flip-flop, D flip-flop, storage registers, shift register, ripple counter, synchronous counters, Finite State Machine (Moore/Mealy Machines), FSM with single/multiple inputs and single/multiple outputs etc.

HARDWARE DESCRIPTION LANGUAGE: Programming and simulation, structural specification,

behavioral specification, dataflow modelling, testbench, testing using test vectors, testing using waveforms, design of basic blocks to build larger circuits, case studies, adder, ALU, counters, shift registers, register bank, FSM design example etc.

TEXT BOOK:
Digital Fundamentals,10th Ed, Floyd T L, Prentice Hall, 2009.

REFERENCE BOOK:
1) Digital Design-Principles and Practices, 4th Ed, J F Wakerly, Prentice Hall, 2006.
2)Digital Design, Morris Mano, Prentice Hall, 2002
3) Digital Systems: Principles and Applications, Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, Pearson Education, Limited, 2011.
4) Fundamentals of Digital Logic with Verilog Design, 2nd Ed, S. Brown and Z. Vrsaniec, McGraw Hill, 2007

Spoken and Written Communication (2-0-0-2)

Course Contents:

Unit-I:
Course Instructor should make an optimal use of cinema for increasing the students' familiarity with English. Testing be done on the basis of the student's comprehension of the plot and the ability of describe scenes from the film. Class room exercise of asking students to comment on the plot or scenes of a given film – not in writing but by standing before the entire class and speaking in English — be frequently carried out. The aim of this unit is to make the student feel confident about her/his ability to form sentence in English for discursive communication.

Unit-II:
Course Instructor should use audio tapes, Ted Lectures, radio news broadcast or celebrated speeches, etc for exposing the students' to a real time' and good spoken English. Class room tests be set to check the students' ability to respond to their listening experience in writing. This will help the Course Instructor to continually assess the requirements of the students and provide corrective advise. Testing the writing skills of students will require setting several questions of very short composition tasks, from 50 words to 150 words. The topics chosen for the composition tasks should be selected from the topics covered in the classroom discussions or from the life on the campus.

Unit-III:
Students should be provided four to five extended samples of written English such as short stories or newspaper editorials for them to mark their difficulties – words, idioms, sentence structures, etc. This will help the students in improving their ability to do focused reading of serious written literature. Testing of the reading comprehension skills be tested by giving them in advance of the test several passages for reading. The Course Instructor may select one or more of those seen passages' for the examination purpose.

TEXT BOOK:
Prism: Spoken and Written Communication, Prose & Poetry published by Orient Longman, 2008.

Reading materials:
The Bet – Anton Chekov
Socrates and the Schoolmaster – F. L. Brayne
An Astrologer's Day – R. K. Narayan
The Gift of the Magi – O' Henry
With the Photographer – Stephen Leacock
Speech on Indian Independence – Jawaharlal Nehru

Semester II

Calculus (3-1-0-4)

Course Contents:

FUNDAMENTALS: Limits, continuity, differentiability, mean value theorems, and Taylor‘s theorem; fundamental theorem of integral calculus, definite integrals, trapezoidal and Simpson’s rule; sequences and series, tests for convergence: absolute and conditional convergence; power series and radius of convergence.

FUNCTIONS OF SEVERAL VARIABLES: Partial derivatives, chain rule, gradient and directional derivative; Taylor’s theorem for functions of several variables; maxima, minima and saddle points.

VECTOR CALCULUS: Gradient, divergence and curl. double, triple, line and surface integrals; theorems of Green, Gauss, Stokes and their applications.

INTRODUCTION TO COMPLEX VARIABLES: Complex numbers and the complex plane, derivative and analytic functions.

DIFFERENTIAL EQUATIONS: First order equations, second linear differential equations, partial differential equations ⟆ basic concepts and important examples, Laplace and Fourier transforms.

TEXT BOOK:
Calculus and Analytical Geometry, 9th Ed, G B Thomas and R L Finney, Addison-Wesley, 1999.

REFERENCE BOOK:
1) Differential and Integral Calculus, 3th Ed, Schaum’s Outline Series, McGraw Hill, 1992.
2) Advanced Engineering Mathematics, 8th Ed, R. Kreyszig, John Wiley, 1999.

Data Structures (3-0-0-3)

Course Contents:

INTRODUCTION: Representation of data on a computer, data types & array and linked list representations ways of representing programs and associated data on computers

ANALYSIS TOOLS: Notion of the running time of an algorithm, recurrences, parameters of performance.

DICTIONARY OPERATIONS: Find, max, min, successor, predecessor (query operations); insert, delete (modify operations)

LIST DATA: Stacks, queues, variants implementation using arrays and linked lists
SORTING: Comparison based sorting algorithms, other sorting algorithms, lower bounds for comparison-based sorting algorithms best-case, worst-case and average-case running times; quicksort, heap Sort, insertion sort, bubble sort etc.

ORDER STATISTICS: Maximum and minimum elements of a set, Finding median, searching for an element of a given rank, finding the rank of a given element, ranks of a subset of elements, maintaining rank information for a dynamic set.

TREES: heaps, Binary search trees (BST), heights of BST

BALANCED BSTs: Red Black trees, AVL Trees, 2,3,4-trees, B Trees

GRAPHS: Representation using adjacency matrices and adjacency lists, Graph searching algorithms BFS and DFS.

TEXT BOOK:
Data Structures and Algorithms, Aho, Hopcroft and Ullman, Addison-Wesley, 1999.

REFERENCE BOOK:
Introduction to Algorithms, 3th Ed, Cormen, Lieserson and Rivest, PHI, 2011.

Data Structures Lab (0-1-4-3)

Lab and take home assignments based on the course “Data Structures”. It is essential for the instructor to use the tutorial hours of this course to give hands on of any object oriented programming language so that students can code the assignments given.

Basic Electronic Circuits (3-1-2-5)

Course Contents:

ANALOG CIRCUIT ELEMENTS: Resistor, Capacitor, Inductor, Concepts of LLFPB, Non-linear circuit elements, Incremental equivalent of nonlinear elements, Voltage and Current sources, Controlled sources, Active circuits, Practical circuit elements of different types.

ANALYSIS OF LINEAR CIRCUITS: Kirchhoff‘s laws, D-C analysis of resistive circuits, Time-domain analysis of a-c circuits ,Sinusoidal steady state analysis of a-c circuits – notions of phasors, impedance, transfer function and frequency response, Frequency response vs transient response , Superposition theorem, Thevenin’s and Norton’s theorems, Two-port parameters, Analysis of circuits having controlled sources.

AMPLIFIERS: Diodes, BJT, FET, Amplifier parameters, Controlled source models, Active devices as controlled sources, Different amplifier configuration using the OPAMP, Frequency response of OPAMP and OPAMP-based amplifiers, Power amplifiers using OPAMP and transistors.

OSCILLATORS: Amplifier with feedback, Condition of harmonic oscillation, RC oscillator circuits.
WAVEFORM GENERATORS: OPAMP as a comparator, Regenerative comparator, Timer, Relaxation oscillator, Non-sinusoidal waveform generator using comparator.

D-C POWER SUPPLY: Half-wave and full-wave rectifiers, Shunt capacitor filter, Ripple and voltage regulation, Voltage regulator using zener diode, Active voltage regulator.

TEXT BOOK:
1) Electronic Principles, 7th Ed, Albert Malvino, and David Bates, Tata McGraw-Hill, 2006.
2) Microelectronic circuits, 5th Ed, A Sedra , K Smith, A N Chandorkar, Oxford University Press,2009.

REFERENCE BOOK:
1) Network Analysis, 3th Ed, Van Valkenburg, PHI, 2000.
2) Introduction to electric circuits, 8th Ed, R C. Dorf and J A Svoboda John Wiley, 2000.
3) Engineering Circuit Analysis, 6th Ed, William H.Hayt, Jack Kemmerly , Steven Durbin, Tata McGraw-Hill, 2002.
4) Electric circuit fundamentals, Sergio Franco, Oxford University Press, 1995.
5) Foundations of Analog and Digital Electronic Circuits, Anant Agarwal and Jeffrey Lang, Morgan Kaufman, 2005.

Computer Organization (3-0-4-5)

Course Contents:
von NEUMANN MACHINE: Functional units, stored program concept, ALU, data paths, registers, status flags; instruction cycle.

DATA REPRESENTATION: Integer data; fixed and floating point systems; representation of non-numeric data (characters, strings, records, and arrays).

ASSEMBLY/MACHINE LEVEL: Instruction sets and types (arithmetic, data movement, and control); instruction formats and addressing modes, subroutine call and return mechanisms; representations of fundamental high-level programming constructs at the assembly language
level; Heap vs. Stack vs. Static vs. Code segments.

MEMORY SYSTEM: Principles of temporal and spatial locality; cache memories (address mapping, block size, replacement and store policy); virtual memory (page table, TLB); disk organization and data access from disk drive.

I/O COMMUNICATION: Handshaking, buffering, programmed I/O, interrupt-driven I/O, bus protocols.

TEXT BOOK:

1) Introduction to Computing Systems: From Bits and Bytes to C and Beyond, 2th Ed, Yale Patt and Sanjay Patel, Tata McGraw-Hill, 2001.
2) Computer Systems: A Programmer's Perspective, 1st Ed Bryant and O'Hallaron, Pearson, 2002.

REFERENCE BOOK:
1) The Essentials of Computer Architecture and Organization, 3rd Ed, Null and Lobur, Jones & Bartlett/Viva Books, 2011.
2) Structured Computer Organization, 6th Ed, Tanenbaum and T Austin, Pearson, 2012.
3) Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.
4) Computer System Organization, N. Jotwani, Tata McGraw Hill, 2009.

Introduction to Computer Science/
Introduction to Information Technology (2-0-0-2)

Course contents

Students of BTech CS and BTech IT, will take this course separately. These courses aim at introducing the broad perspective of computer science and information technology to the respective students. Students get to understand the breadth of the subject area they would be exploring in the coming years. It is expected that more than one faculty instructor would deliver the lectures of these courses.

Introduction to computer science (1-0-2-2)

Course Contents:

Module – 1: Models of Computations and Complexity
Module – 2: Algorithms
Module – 3: Computer Networks
Module – 4: Cryptology and Security

Each module will have 3- 4 lectures. There is no text book specified as instructor may choose to introduce only glimpses of each module.

Semester - III

Object Oriented Design and Programming (3-0-0-3)

Course Contents: INTRODUCTION: Principles of OOD; programming Paradigms; benefits of OOD&P, applications of OOD; Classes and objects; access qualifiers; instance creation; constructors, parameterized constructors, overloaded constructors, constructors with default arguments, copy constructors, static class members and static objects.

FUNCTIONS an OPERATORS: Function prototyping, function components, passing parameters, inline functions, default arguments, overloaded function; array of objects, pointers to objects, dynamic allocation operators, dynamic objects; Operator overloading, overloading unary and binary operator, overloading the operator using friend function, stream operator overloading, data conversion.

INHERITANCE: Defining derived classes, single inheritance, protected data with private inheritance, multiple inheritance, multi level inheritance, hierarchical inheritance, hybrid inheritance, multipath inheritance, constructors in derived and base class, abstract classes, virtual function and dynamic polymorphism, virtual destructor.

EXCEPTION HANDLING: Principle of exception handling, exception handling mechanism, multiple catch, nested try, re/throwing the exception.

OBJECT ORIENTED DESIGN: Requirements modeling, business modeling, component based development; Rational Unified Process (RUP), process overview, phases and iterations, static structure of the process, core workflows; UML history, building blocks of UML, structural modeling, behavioral modeling; Use Case Diagrams, Modeling Ordered Interactions: Sequence Diagrams; case studies.

TEXT BOOK:

1) Introduction to object-oriented programming, B. Timothy, Pearson, 2001.

2) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, C. Larman , Prentice Hall, 2004.

REFERENCE BOOK:

1) Object Oriented Design and Patterns, C. Horstmann, John Wiley & Sons, 2005.

2) Unified Modeling Language User Guide, G. Booch, J. Rumbaugh, I. Jacobson, Pearson Education, 2001.

3) Object-Oriented Systems Analysis and Design using UML, Bennett, McRobb, Farmer, McGraw-Hill, 2002

Object Oriented Design and Programming Lab (0-1-4-3)

Course Contents: Lab and take home assignments based on the course “OOD & P”. Emphasis on following topics:

- Eclipse (or NetBeans) IDE introduction
- Compiling & running programs on IDE
- Object oriented coding conventions
- Simple example of object-oriented design and message passing
- Problems on object based iteration
- Problems on object based arrays, matrices, and strings
- Design oriented problems on object polymorphism
- Design oriented problems on object inheritance & overriding
- Object-oriented designing of advanced data structures (linked list, trees, graphs, tables)
- Problems on object based linked lists
- Problems on object based trees
- Problems on object based graphs
- Mini Projects

Operating Systems (3-0-2-4)

Course Contents:

Introduction: overview and history, multiprogramming, functions of an OS, device drivers, I/O interrupts, and system call interface.

Process management: the process abstraction, process states, implementing processes (PCB), CPU scheduling algorithms, real-time scheduling, inter-process communication, threads.

Resource allocation: classical synchronization problems; synchronization primitives; semaphores; monitors; Deadlocks, deadlock avoidance.

Virtual memory management: segmentation, demand paging, hardware support, page fault handling; page replacement algorithms; shared memory.

Files systems: logical disks, file organization, buffer cache, file descriptor, directory structure, blocks and fragments, directory tree, inodes, implementation of file system, file protection and access control.

Text Books:

William Stallings, Operating Systems: internals and design principles, 6th edition PHI.

Reference Books:

Silberschatz, Abraham, Galvin, Peter B. & Gagne, Greg: Operating system concepts, 8th ed.. New Delhi. Wiley India, 2009.
Andrew Tanenbaum, Modern Operating Systems.
Uresh Vahalia, UNIX Internals: The New Frontiers, Pearson Education Asia, 2002.

Systems and Signal Theory (3-0-2-4)

Course Contents

The course first builds up fundamentals and introduction of signals and systems ranging from their types, properties and different examples. The course then discusses extremely useful class of systems, *viz.*, linear time-invariant (LTI) systems and their representation with the help of convolution sum (which can used to model many physical processes). Fourier analysis ranging from Fourier series, Fourier transform, time and frequency characterization is discussed in a greater depth followed by study on Shannon's sampling theory. The course then shifts to Z-transform and Laplace transform with detailed discussion on properties and their applications. The course also discusses in detail design of digital resonators and their applications in addition to discussion on application of signals and systems methods to control systems and communication systems.

Text books

1. A. V. Oppenheim, A. S. Wilsky and S. H. Nawab, "Signals and Systems," *2nd* edition, Prentice-Hall of India, 1999 .
2. B. P. Lathi, "Linear Systems and Signals," Berkeley-Cambridge Press, 1992.
3. Simon Haykin, "Signals and Systems,"

Reference Books:

1. A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.
2. G. Strang, *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, Publishers, San Diego, 1988.
3. R. N. Bacewell, "The Fourier Transform and Its Applications," 2nd edition, McGraw-Hill, New York, 1986.

Probability and Statistics (3-1-0-4)

Course Contents:

INTRODUCTION: Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Bayes' theorem and independence.RANDOM VARIABLES: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, probability and moment generating function, median and quantiles, Markov inequality, Chebyshev's inequality.SPECIAL DISTRIBUTIONS: Discrete uniform, binomial, geometric, negative binomial, hypergeometric, Poisson, continuous uniform, exponential, gamma, beta, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability and hazard rate, reliability of series and parallel systems.JOINT DISTRIBUTIONS: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution.TRANSFORMATIONS: functions of random vectors, distributions of order statistics, distributions of sums of random variables.SAMPLING DISTRIBUTIONS: Mean, median, variance, standard deviation, The Central Limit Theorem, distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions. ESTIMATION: Unbiasedness, consistency, the method of moments and the method of maximum likelihood estimation, confidence intervals for parameters in one sample and two sample problems of normal populations, confidence intervals for proportions.TESTING OF HYPOTHESES: Null and alternative hypotheses, the critical and acceptance regions, two types of error, power of the test, the most powerful test and Neyman-Pearson Fundamental Lemma, tests for one sample and two sample problems for normal populations, tests for proportions, Chi square goodness of fit test and its applications.

TEXT BOOK:

Introduction to Probability and Statistics for Engineers and Scientists, S. M. Ross, Academic Press, 2009.

REFERENCE BOOK:

1) Introduction to Probability and Statistics, J.S. Milton & J. C. Arnold, Cengage Learning, 2008
2) Introduction to Probability Theory and Statistical Inference, H.J. Larson, Wiley, 1982.
3) A First Course in Probability, S.M. Ross, Prentice Hall, 2001.

Economics (3-0-0-3)

Course Contents

The Problems of Economic Organisation; Demand and Supply; Price Determination; Elasticity of Demand and Supply; Theory of Production; Production function; Law of diminishing returns; Analysis of Cost; Fixed and variable costs; Marginal cost; Market Structure and Various Types of Markets; Perfectly Competitive Market; Monopolistic Markets; Aggregate Demand and Aggregate Supply; Determination Of National Income and criticisms; Consumption, Saving and Investment; Business Cycle and remedies; International Trade; Balance of Payment; Case for and against free trade; Economics of banking; Interest rates and demand for money; Role of Central Bank; Inflation: measurement, causes and index numbers.

TEXT BOOK:

1) Economics, P. Samuelson & Nordhaus, Tata-McGraw Hill.

2) Indian Economy, Ruddar Datt & Sundaram, S. Chand & Co.

Semster - IV

Database Management System (3-0-0-3)

Course Contents:

INTRODUCTION AND CONCEPTUAL MODELING: Databases and database users; database system concepts and architecture; data modeling using the entity relationship (ER) model; enhanced entity relationship.

DATA STORAGE AND INDEXING: Introduction, record storage, and primary file organization index structures for files; single level indexing; multilevel indexing.

RELATIONAL MODEL: The relational data model; relational database constraints; relational algebra; relational calculus; relational database design by ER and EER; relational mapping; SQL; the relational database standard; examples of relational database management systems; Oracle.

DATABASE DESIGN THEORY AND METHODOLOGY: Functional dependencies and normalization for relational databases, relational database design algorithms and further dependencies.

SYSTEM IMPLEMENTATION TECHNIQUES: Query processing and optimization, transaction processing concepts, concurrency control techniques, database recovery techniques .

OBJECT AND OBJECT RELATIONAL DATABASES: Object database concepts, the ODMG standard for object databases, object relational systems and SQL.

EMERGING APPLICATIONS: Distributed databases and client/server models, XML Database (DTD, XML Schema), Query for XML Database, NoSQL.

TEXT BOOK:

Fundamentals of Database Systems, R. Elmasri, S. B. Navathe, Prentice Hall, New Delhi, 2007.

REFERENCE BOOK:

1) Database System Concepts, A. Silberschatz, H. F. Korth, S. Sudharshan,Tata McGraw Hill, New Delhi, 2005.

2) Introduction to Database Systems, C. J. Date, Prentice Hall, New Delhi, 2004.

Database Management System Lab (0-1-4-3)

Course Contents:

Lab and take home assignments based on the course “DBMS”. Emphasis on following topics:

ER MODELING TOOL (ERWin): Introduction to ERWin; Adding Entity types & relations; Forward generation.

ABSTRACT QUERY LANGUAGE INTERPRETER (JCup & JFlex): Relational Algebra (syntax, RA interpreter); Domain Relational Calculus (syntax, DRC interpreter); Datalog (syntax, Datalog interpreter).

RELATIONAL DATABASE MANAGEMENT SYSTEM (Oracle): SQL* Plus Utility; SQL* Loader Utility; Programming with Oracle using JDBC API.

RELATIONAL DATABASE MANAGEMENT SYSTEM (MySQL): MySQL Utility; Bulk loading of data; MySQL and PHP programming; Making an online Address Book.

DATABASE DESIGN TOOLKIT (DBD): Coding Relational Schemas & Functional Dependencies; Invoking SWI-Prolog Interpreter; DBD system predicates (xplus, finfplus, fplus, implies, equiv, superkey, candkey, mincover.

OBJECT-ORIENTED DATABASE MANAGEMENT SYSTEM (db4o): db4o Installation & Introduction; Simple database creation exercise; Database updates & deletes; Database Querying (queryByExample, Native Queries, SODA Queries); Company database application exercise; Web application exercise (client-server configuration).

XML DATABASE: XML basics; Creating a company database in XML; XML Editor (EditiX); XPath; XQuery; FLWOR expressions; XML Schema

Reference:

Fundamentals of Database Systems: Laboratory Manual, R. Sunderraman (http://tinman.cs.gsu.edu/~raj/elna-lab-2010/lab-manual.pdf)

Computer Networks (3-0-0-3)

Course Contents:

Introduction: Overview of an internet, internet as a service, internet architecture, circuit switching, packet switching, network performance metrics (delay, packet loss and throughput), layered approach (TCP/IP and OSI models)

Link layer: multiple access protocols (channel partitioning protocols, random access protocols and CSMA protocols), Ethernet – IEEE 802.3, Token ring – IEEE 802.5, WiFi – IEEE 802.11, reliable link layer protocols (stop and wait, sliding window protocols), switches and bridges.

Network layer: IP addressing: IPv4, IPv6 and ICMP header formats; intradomain routing: distance vector and link state routing protocols; interdomain routing: BGP; routing for multicasting and broadcasting.

Transport layer: principles of reliable data transfer; connection oriented transport: TCP connection establishment, TCP timeout estimation, TCP RTT estimation, TCP congestion control; connectionless transport: UDP.

Application layer: network applications, hypertext transfer protocol, domain name system, simple mail transfer protocol, socket interface, client-server programming.

Textbooks:

1. Computer Networks: A Systems Approach (Fifth edition) by L. L. Peterson and B. S. Davie, publisher: Morgan Kaufmann
2. Computer Networking: A Top-Down Approach (Fifth edition) by J. F. Kurose and K. W. Ross, publisher: Pearson

References:

TCP/IP Illustrated Volume 1: The Protocols, Second edition, K. R. Fall and W. Richard Stevens, publisher: Pearson.

Computer Networks LAB (0-1-4-3)

Lab assignments based on the course “Computer Networks”.

Design and Analysis of Algorithms (3-0-2-4)

Course Description:
This course intends to provide a rigorous introduction to fundamental techniques in the design and analysis of algorithms. The course can be divided into five parts namely (1) Foundations, (2) Sorting and Order Statistics, (3) Advanced Design and Analysis Techniques, (4) Graph Algorithms and (5) Special Topics. In the Foundations part, we will overview asymptotic notation, divide and conquer techniques, solving recurrences, probabilistic analysis and randomized algorithms. In the Sorting and Order Statistics part, we will cover heapsort, quicksort, sorting in linear time and median order statistics. In the Advanced Design and Analysis Techniques part, we will cover dynamic programming, greedy algorithms and amortized analysis. In the Graph Algorithms part, we will cover depth first search, breadth-first search, bi-connectivity and strong connectivity, topological sort, minimum spanning trees, shortest paths and maximum flow. If time permits, we will cover Rabin-Karp and Knuth-Morris-Pratt’s string matching algorithms as a part of Special Topics. This course is open to third and final year B.Tech students..

Topics

Foundations: Asymptotic notation, Divide and Conquer Techniques, Solving Recurrences, Probabilistic Analysis and Randomized Algorithms

Sorting and Order Statistics: Heapsort, Quicksort, Sorting in Linear Time, Median Order Statistics.

Advanced Design and Analysis Techniques: Dynamic Programming, Greedy Algorithms and Amortized Analysis

Graph Algorithms: Depth First search, Breadth First Search, Bi-connectivity and Strong Connectivity, Topological Sort, Minimum Spanning Trees, Shortest Paths and Maximum Flow

String Matching: Rabin-Karp and Knuth-Morris-Pratt's string matching algorithms

Text Book: Introduction to Algorithms by Cormen, Lieserson, Rivest and Stein, 2nd or 3rd edition, Eastern Economy Edition,, MIT Press.

Science, Technology, Society (3-0-0-3)

Course Contents:

Module 1:

- Introduction to STS as a field of study and research in the twentieth century
- Philosophical, Historical and Sociological Approaches to Science and Technology and Society
- The growth and identity of Modern Science and Technology in India

Module 2: Science Communication- Institutions, ideologies, practices

- The diversity of science communication in colonial India
- Science communication and the Nehruvian Agenda
- The ideology and image of developmental science
- The agenda of People's Science
- Liberalization and the commoditization of science and technology

TEXT BOOK:
1) Science, Technology and Medicine in Colonial India – David Arnold (Cambridge,2004)
2) Western Science in Modern India, Metropolitan Methods, Colonial Practices – Pratik Chakrabarti, (Permanent Black, 2004)

REFERENCE BOOK:
A Concise History of Science in India – D. M. Bose, S. N. Sen, and B.V. Subarayappa (Universities Press, 2009)

Technical Writing (2-0-2-3)

Course Contents:

Structure of sentences, paragraphs, and documents. using stress for emphasis, and sequencing topics to create forward flow, writing for the reader; Formats of technical documents; the experimental report; the technical report, the proposal; workshop on published documents; Discussion and workshop on term paper proposals; Graphics; emphasis without distortion; visual illusions; a minimalist approach to data representation; univariate and multivariate displays; Discussion and workshop on term papers; elements of oral presentations; oral presentations.

TEXT BOOK:

1) The Elements of Style, W. Strunk, E B White, New York: Macmillan, 1972.

2) The Mayfield Handbook of Technical and Scientific Writing, L. Perelman, Mayfield Publishing Company, 1998.

3) The Science of Scientific Writing, G. D. Gopen, J. A. Swan, American Scientist, 78(6):550-558, Nov-Dec 1990.

Semester V

Principles of Programming Language (3-0-2-4)

Course Description: Introduces theoretical concepts in programming languages and familiarizes students with different paradigms including functional, object-oriented and logic programming paradigms. This course emphasizes the design and implementation of programming languages, including data representation and types, functions, sequence control, environments, block structure, subroutines, storage management.

Course Goals/Objectives

- Be able to compare and evaluate different programming languages and implement different programming constructs and features (e.g. variables, loops, procedures, dynamic memory);
- Be able to
  - Formally specify the syntax and semantics of programming languages.
  - Write a lexical analyzer, parser and a translator to convert from one language to another;
  - Describe the semantics of and implement an interpreter and compiler for a simple programming language;
  - Understand the major programming paradigms and be able to use at least one language from each paradigm.

Required Text: Kenneth C. Louden & Kenneth A. Lambert, *Programming Languages, Principles and Practice, 3rd Edition*, Cengage Learning, 2012

Reference Text: Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd *Edition*, Pearson Education, 2006
Michael L. Scott, *Programming Language Pragmatics, 3rd Edition*, Morgan Kaufmann Publishers, 2009

Formal Languages and Automata Theory (3-0-2-4)

Course Description: This course introduces Introduce students to (i) the various types of regular languages, their equivalences to finite automata, various techniques in analyzing the closure and algorithmic properties of regular languages; (ii) context sensitive languages, their grammars; and (iii) Turing machines and Undecdability.

Course Goals/Objectives: Introduce students to (i) the various types of regular languages, their equivalences to finite automata; (ii) context sensitive languages, their grammars; and (iii) Turing machines and Undecdability.

Course Schedule and Tentative List of Topics: This course introduces Regular Languages: various types of finite automata and their equivalences thereof, minimization of finite automata, Myhill-Nerode theorem, regular expressions, regular grammars, closure properties of regular languages, pumping lemma, algorithmic properties of regular languages;

Context-free languages: context-free grammars, derivation trees, ambiguous grammars, Chomsky and Greibach normal form, nondeterministic and deterministic pushdown automata, pumping lemma and Ogden's lemma, closure and algorithmic properties of context-free languages, Top-down and Bottom-up parsing;

Context sensitive languages: context sensitive grammars, linear bounded automata; Turing machines: recursively enumerable languages, unrestricted grammars, variants of Turing machines and equivalence thereof. Undecidability.

Required Text:

J. E. Hopcroft, R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson, 2001.

Reference Text: H. R. Lewis and C. H. Papadimitriou, Elements of the Theory of Computation, Prentice Hall, 1997/Pearson 1998.

Software Engineering (3-0-0-3)

Course Contents:

INTRODUCTION: The evolving role of software; software characteristics; software process - software process models; linear sequential model; prototyping model; The RAD model; evolutionary software process models; The incremental model; The spiral model.

SYSTEM ENGINEERING: Requirements analysis and negotiation; Requirements validation; Requirements management.

ANALYSIS MODELING: Data modeling; data objects, attributes and relationships; cardinality and modality; entity-relationship diagram; data flow diagrams; data dictionary.

DESIGN CONCEPTS AND PRINCIPLES: Software architecture; control hierarchy; structural partitioning; functional Independence; cohesion, coupling; design documentation; architectural design; transform centered architecture; transaction centered architecture; user Interface design models, user interface design process.

TESTING TECHNIQUES: Software testing fundamentals; test case design; White box testing; basis path testing; control structure testing; Black box testing, testing for specialized environments, testing strategies; verification and validation - unit testing, integration testing, validation testing, system testing, debugging.

SOFTWARE QUALITY ASSURANCE: Quality concepts; cost of quality, Software Quality Assurance (SQA) Group - roles and responsibilities, formal technical reviews, quality standards.

TEXT BOOK:

Software Engineering – A Practitioner's Approach, R. S. Pressman, McGraw Hill International Edition, Singapore, 2006.

REFERENCE BOOK:

1) Software Engineering, I. Sommerville, Pearson Education, New Delhi, 2001.
2) An Integrated Approach to Software Engineering, P. Jalote, Narosa Publishers, New Delhi, 2005.

Software Engineering LAB (0-1-4-3)

Lab assignments based on the course "Software Engineering".

Numerical Linear Algebra (3-0-0-3)

Course Contents:

Vector sub-spaces, dimension, Linear transformations and their representation by matrices, rank of a matrix Triangular form, Matrix norms, Conditioning of linear systems, Direct methods (Gauss, Cholesky, Householder), Iterative methods (Jacobi, Gauss-Seidel,Relaxation) for solving linear systems, Computing of eigen values & eigen-vectors (Jacobi, Givens-Householder, Q-R, Inverse methods), Conjugate gradient method & its preconditioning.

Text Book
Applied Numerical Linear Algebra, James W. Demmel, SIAM, 1997.
Numerical Linear Algebra, Lloyd V. Trefethen, David Bau, SIAM, 1997.

References:
1. S Kumaresan, Linear Algebra, A Geometric approach, Prentice Hall of India, 2000.

Environmental Sciences (3-0-0-3)

Course Contents: TBD [Taking clue from the curriculum of DA-IICT, Gandhinagar]

Sciences Elective – 1 (3-0-0-3)

Semester VI

Compiler Design (3-0-0-3)

Course Description: This course introduces concepts relevant to practical compiler construction. Topics include formal programming language translation, program syntax, semantics, Finite state machines, regular expressions, context-free parsing techniques such as LL(k) and LR(k), code generation, simple optimizations.

List of Topics (Topics generally conform to the order in the primary required text)**:** Lexical analysis (scanning, scanner generation), Parsing (recursive descent, LL(1), LR(1)), Context-sensitive analysis (

ad hoc techniques and attribute grammars or syntax-directed translation), Intermediate representations, The procedure abstraction and how to implement it, Heap management, Simple code generation, Instruction selection (better code generation), Register allocation (better code generation), Code improvement techniques (data-ow analysis, dependence analysis, simple transformations).

Required Text: A. Aho, R. Sethi, J. Ullman, M. Lam, Compilers: Principles, Techniques and Tools
Addison-Wesley Publishing Company, New York, New York, 2006

Compiler Design LAB (0-1-4-3)

Lab assignments based on the course “Compiler Design”.

Introduction to Artificial Intelligence (3-0-2-4)

Course Contents:

INTRODUCTION: The state of art; intelligent agents; structure; environment.
SEARCH STRATEGIES: Breadth-first search; uniform cost search; depth-first search; depth-limited search; iterative, deepening search; bi-directional search; heuristic search techniques; comparing search strategies.
KNOWLEDGE AND REASONING: Prepositional logic; predicate logic; rules; forward and backward chaining; strong and weak slot fillers.
PLANNING: Overview, example domain, components; goal stack planning; non-linear planning; hierarchical planning; reactive systems; uncertainty: non-monotonic reasoning; logics; implementation; probability and Bayes theorem; certainty factors; Bayesian networks; Dempster Shafer theory.
INTRODUCTION TO EXPERT SYSTEMS: What is an Expert System; advantages of Expert System; general concepts of Expert system, characteristics of Expert System; Expert System application and domain.
THE REPRESENTATION OF KNOWLEDGE: The meaning of knowledge, production, semantic nets, schemata, frames; prepositional logic; The first Order Predicate Logic; The Universal Quantifier; The Existential Quantifier.
DESIGN OF EXPERT SYSTEMS: Introduction, rule-based system architecture, nonproduction system architecture, dealing with uncertainty; knowledge acquisition and validation; knowledge system building tools; selecting the appropriate problem; stages in the development of Expert system; errors in development stages; software engineering and expert systems.

TEXT BOOK:
1) Artificial Intelligence – A Modern Approach, S. Russell, P. Norvig, Pearson Education, New Delhi, 2002.
2) Artificial Intelligence, E. Rich, K. Knight, Tata McGraw Hill, New Delhi, 1991.

REFERENCE BOOK:
1) Artificial Intelligence, P. H. Winston, Pearson Education, New Delhi, 1992.
2) Introduction to AI and Expert Systems, D. W Patterson, Prentice Hall, New Delhi, 1990.

Computer Architecture (3-0-2-4)

Course contents

Introduction**,** Logic Design Convention**,** Building a Data path**,** An overview of Pipelining**,** Pipelined Data path and Control**,** Data dependency and hazard**,** Control hazard and Structural Hazard**,** Exceptions
Defining Computer Architecture**,** Trends in technology**,** Measuring, reporting and summarizing performance
Instruction Level Parallelism (ILP) Concepts and Challenges, Basic compiler techniques for exposing ILP, Reducing Branch costs with advanced branch prediction techniques, Overcoming Data hazards with dynamic scheduling, Dynamic scheduling: examples, Hardware based speculation, Exploiting ILP using multiple Issue and static scheduling (Super-scalar), ILP using dynamic scheduling, multiple issue and speculation, Limitations of ILP
Data-level parallelism in Vector, SIMD and GPU architecture, Introduction to Vector, SIMD and GPU architecture
Review of Memory Hierarchy - Introduction, Cache performance, Six basic cache optimization, Virtual Memory
Thread-Level Parallelism TLP / Issues in Multicore processors - Introduction, Centralized shared-memory architectures, Performance of Symmetric shared memory multiprocessors, Models of memory consistency, Distributed Shared memory and directory based coherence, Synchronization: basics

Textbook

1. Computer Organization and Design: The hardware/software Interface (ARM edition) by John L Hennessy & David A Patterson (This book will be referred to as COD)
2. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy & David A Patterson (you may use earlier editions. It may have different Chapter numbers) (CAQA)

Introduction to Cryptography and Security (3-0-0-3)

Course Contents:

Secret Key Cryptography: Substitution-Permutation Network; Feistel structure; block ciphers - DES, AES, IDEA; stream ciphers – LFSR, RC4, eStream ciphers.
Modes of operation: ECB, CBC, CFB, CTR, OFB.
Data Integrity: Hash functions – MD5, SHA; Message Authentication Codes.
Public Key Cryptography: Integer Factorization Problems - RSA, Rabin's scheme; Discrete Logarithm Problems - Diffie-Hellman, ElGamal, DSA; CRT; Elliptic Curves arithmetic - ECDSA, identity-based crypto; PKI.
Security Models: CPA, CCA/CCA2.
Kerberos; E-Mail Security; SSL/TLS; Web security; Systems security; Access controls; Malwares; Firewalls; Intruders.

Text Book:
1) Introduction to Cryptography with Coding Theory -- Washington & Trappe, [Pearson Education], 2009.
2) Introduction to Modern Cryptography -- Katz & Lindell, [CRC press], 2007.
3) Computer Security: Art and Science -- M. Bishop, [Pearson Education], 2004.

Reference:
1) Applied Cryptography: Protocols, Algorithms, and Source Code in C -- Bruce Schneier, [John Wiley & Sons].
2) Network Security -- Kaufman, Perlman and Speciner, [Pearson Education], 2002.
3) Cryptography and Network Security -- W. Stallings, [Prentice Hall], 2010.
2) Handbook of Applied Cryptography -- A. Menezes, P. van Oorschot and S. Vanstone [CRC press].

Science Elective – 2 (3-0-0-3)

Humanities Elective – 1 (3-0-0-3)

Semester VII

Professional Ethics (3-0-0-3)

Course Contents:

HUMAN VALUES: Morals, values and ethics, integrity; service learning; civic virtue; respect for others; sharing; honesty; courage; valuing time; cooperation; commitment; empathy; self-confidence; spirituality.
ENGINEERING ETHICS: Senses of Engineering Ethics; moral dilemmas, moral autonomy; Kohlberg's theory; Gilligan's theory; consensus and controversy; models of professional roles; theories about right action; self-interest; customs and religion; uses of ethical theories.
ENGINEERING AS SOCIAL EXPERIMENTATION: Engineers as responsible experimenters; codes of ethics; case study.
SAFETY, RESPONSIBILITIES AND RIGHTS: Safety and risk; assessment of safety and risk; risk benefit analysis and reducing risk; the three mile island and chernobyl case studies.
COLLEGIALITY AND LOYALTY: Respect for authority; collective bargaining; confidentiality; conflicts of interest; occupational crime; professional rights; employee rights; Intellectual Property Rights (IPR); discrimination.
GLOBAL ISSUES: Multinational corporations; environmental ethics; computer ethics; weapons development; engineers as managers; consulting engineers; engineers as expert witnesses and advisors; moral leadership; sample code of ethics like ASME, ASCE, IEEE, IE (India), IETE (India).

TEXT BOOK:

Engineering Ethics, C. D. Fleddermann, Pearson Education/ Prentice Hall, New Jersey, 2004.

REFERENCE:

1) Ethics in engineering, M. Martin, R. Schinzinger, McGraw-Hill, New York1996.

2) Engineering Ethics – Concepts and Cases, C. E. Harris, M. S. Pritchard, M. J Rabins, Thomson Learning, United States, 2000.

Distributed Computing (3-0-2-4)

Course Description:

This course covers the foundations of distributed systems including models of computing, different types of communication (Layered Protocols, Remote Procedure Calls, Remote Objects, messages, streams), process models (threads, client/server, code migration and software agents), naming of entities, logical clocks and synchronization. We will review some of the popular applications of distributed computing including distributed file systems and web services. The course will include two programming assignments, one group project and one project presentation. The programming assignments will provide hands on experience in understanding different types of communications and process models. The project will help develop a reasonably sized distributed application based on research papers that will be discussed during the semester. The expectation is that students have mastered one or more programming languages. Specific language mastery is not important, though knowing one of C, C++, or Java will be helpful.

Course content

Overview of C, UNIX and UNIX system calls.
INTRODUCTION: Definition of a distributed system, goals, hardware concepts, software concepts, the client-server model.
COMMUNICATION: Layered Protocols, Remote Procedure Call, Remote Object Invocation, Message-Oriented and Stream-Oriented Communication.
PROCESSES: Threads, Servers, Code Migration and Software Agents.
NAMING: Naming Entities, Locating Mobile Entities, Removing Unreferenced Entities.
SYNCHRONIZATION: Clock Synchronization, Logical Clocks and Election Algorithms.

Text Book
A. Tanenbaum, M. V. Steen: Distributed Systems principles and paradigms. Prentice Hall ISBN 0-13-121786-0

Reference Book

G. Coulouris, J. Dollimore, T. Kindberg, Distributed Systems: Concepts and Design. Prentice Hall ISBN 0-201-61918-0.

Technical Elective – 1 (3-0-2-4)

Technical Elective – 2 (3-0-2-4)

Technical Elective – 3 (3-0-2-4)

Technical Elective – 4 (3-0-0-3)

Semester VIII

Project (0-0-36-18)

Final semester project is a guided project. Students can take individual project or group project. In case it is a group project the size of the group would be restricted to not more than two students. Students would be allowed to undertake their final semester project outside the Institute. Students can undertake such projects in any other educational institute or in a research lab. Students would also be allowed to join industry for this final semester project in case the nature of the project is based on some research.