DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Scheme of Instructions and Syllabus for Under Graduate Studies

B.Tech.
in Computer Science & Engineering
2017 - 2018

Visvesvaraya National Institute of Technology, Nagpur

February 14, 2017

MISSION AND VISION
OF
VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY, NAGPUR

MISSION

The Mission of VNIT is to achieve high standards of excellence in generating and propagating knowledge in engineering and allied disciplines. VNIT is committed to providing an education that combines rigorous academics with joy of discovery. The Institute encourages its community to engage in a dialogue with society to be able to effectively contribute for the betterment of humankind.

VISION

To contribute effectively to the national endeavor of producing quality human resource of world class standard by developing a sustainable technical education system to meet the changing technological needs of the Country, incorporating relevant social concerns and to build an environment to create and propagate innovative technologies for the economic development of the Nation.

MISSION AND VISION OF

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, VNIT, NAGPUR

MISSION

To produce highly qualified and motivated graduates through a rigorous curriculum of theory and application that develops the ability to solve problems, individually and in teams.

Creating knowledge of fundamental principles and innovative technologies through research within the core areas of computer science and also in inter-disciplinary topics.

Serving the communities to which we belong at local and national levels, combined with a deep awareness of our ethical responsibilities to our profession and to society.

VISION

To contribute effectively to the important national endeavour to produce quality human resource in the information technology and related areas for sustainable development of the country's IT industry needs.

To advance the state of the art in computer science and engineering by working on cutting edge research topics, publishing quality research papers and filing enduring patents.

To serve the local and the national community by creating awareness about IT related products and to impress upon then the importance of knowledge management.

Department of Computer Science & Engineering came into being in 1987. It offers under-graduate and post-graduate programs. The department has well qualified and motivated faculty members and support staff. The laboratories are adequately equipped with state-of-the-art facilities. The department is actively involved in R&D as well as consultancy projects and has collaborations with several industries, academic institutes and R&D organizations in the country.

B.Tech. Computer Science Engineering is a eight semester program, wherein student has to complete certain number of credits as indicated in Table 1. Each subject (or course) has certain number of credits. There are two types of subjects: Core and Elective. Core courses are compulsory and some courses from electives are to be taken to complete the required credits.

TABLE 1 : CREDIT REQUIREMENTS FOR UNDER GRADUATE STUDEIS

Program Core (PC)		**Program Elective (PE)**	
Category	**Credit**	**Category**	**Credit**
Basic Science (BS)	43	Departmental Electives (DE)	41-47
Engineering Science (ES)		Humanities & Management (HM)	0-3
Humanities (HU)	-	Open Courses (OC)	0-3
Departmental Core (DC)	80		
Total	**123**	**Total**	**47**
Grand Total PC+PE			**170**

The number of credits of a subject depends on number of classes in a week. For example a subject with 3-1-0 (L-T-P) means it has 3 Lectures, 1 Tutorial and 0 Practical in a week. This subject will have four credits (3x1+1x1+0x1 = 4). If a student is declared pass in a subject, then he / she gets the credits associated with that subject. Depending on marks scored in a subject, student is given a Grade. Each grade has got certain grade points as follows :

Grades	AA	AB	BB	BC	CC	CD	DD	FF
Grade Points	10	09	08	07	06	05	04	Fail

The performance of a student will be evaluated in terms of two indices, viz. the Semester Grade Point Average (SGPA) which is the Grade Point Average for a semester and Cumulative Grade Point Average (CGPA) which is the Grade Point Average for all the completed semesters at any point in time. SGPA & CGPA are :

$$SGPA = \frac{\sum_{semester}(\text{Course credits x Grade points}) \text{ for all courses except audit}}{\sum_{semester}(\text{Course credits}) \text{ for all courses except audit}}$$

Students can Audit a few subjects i.e. they can attend the classes and do home work and given exam also, but they will not get any credit for that subject. Audit subjects are for self enhancement of students.

Programme : B.Tech. in Computer Science Engineering

Programme Educational Objectives of B.Tech. in Computer Science Engineering

1. Achieve the understanding of the basics and emerging techniques of a broad range of computer science and engineering concepts. Gain the ability to analyze and solve computer science and engineering problems through application of fundamental knowledge of maths, science, and engineering.

2. Learn to apply modern skills, techniques, and engineering tools to create computational systems. Understand the state of the art in the recent areas of research in computer science and engineering and to formulate problems from them and perform original work to contribute in the advancement of the state of the art.

3. To be able to adapt to the evolving technical challenges and changing career opportunities. Learn to effectively communicate ideas in oral, written, or graphical form and to promote collaboration with other members of engineering teams.

Programme Outcomes of B.Tech. in Computer Science Engineering

1) To obtain sound knowledge in the theory, principles and applications of computer systems.
2) Apply knowledge of mathematics, science, and engineering in the design and development of software systems.
3) Configure recent software tools, apply test conditions, and deploy and manage them on computer systems.
4) Perform experiments on different software packages either obtain from external parties or developed by themselves and analyse the experimental results.
5) Design and develop software projects given their specifications and within performance and cost constraints.
6) Identify, formulate and solve software engineering problems and understand the software project management principles.
7) Ability to understand the computing needs of inter-disciplinary scientific and engineering disciplines and design and develop algorithms and techniques for achieving these.
8) Acquire and understand new knowledge, use them to develop software products, and to understand the importance of lifelong learning.
9) Ability to extend the state of art in some of the areas of interest and create new knowledge.
10) Communicate effectively in oral, written and graphical form.
11) Work cooperatively, responsibly, creatively, and respectfully in teams.
12) Understand professional and ethical responsibilities and analyze the impact of computing on individuals, organizations, and the society.

i) General information

Department of Computer Science & Engineering came into being in 1994. It offers under-graduate and post-graduate programs. The department has well qualified and motivated faculty members and support staff. The laboratories are adequately equipped with state-of-the-art facilities. The department is actively involved in R&D as well as consultancy projects and has collaborations with several industries, academic institutes and R&D organizations in the country.

ii) List of faculty members

Name of the faculty	Qualification	Designation	Area of Research
N.S. Chaudhari	Ph.D.	Director	Game AI,Novel Neural Network Models like Binary nets and Bidirectional nets,Context free grammer parsing and Graph Isomorphism Problem.
O.G. Kakde	Ph.D.	Professor	Language Processor,Computer Programming Languages,Advanced Compiler Construction
S. R. Sathe	Ph.D.	Professor	Theoretical Computer Science
P.S. Deshpande	Ph.D.	Associate Professor,	Database Management Systems, Data warehousing & Mining, Pattern Recognition
U. A. Deshpande	Ph.D.	Associate Professor	Multi-agent systems, Distributed Systems, Soft Computing
A.S. Mokhade	Ph.D.	Associate Professor	Software Engineering, Software Architecture
R. B. Keskar	Ph.D	Associate Professor	Telecommunication software, Distributed Systems, Formal Verification, Compiler Optimization
M. P. Kurhekar	Ph.D..	Associate Professor	Theoretical Computer Science, BioInformatics
M. M. Dhabu	Ph.D.	Assistant Professor	Soft Computing, Network Security

Ashish Tiwari	Ph.D.	Assistant Professor	Mobile Communication, Information Security, Operating Systems
S. A. Raut	Ph.D.	Assistant Professor	Data Mining and Warehousing, Business Information Systems
Deepti Shrimankar	Ph.D.	Assistant Professor	Parallel and Distributed Systems, Embedded Systems, Computer Networks
M.A. Radke	M.S.(CSE)	Assistant Professor	Information Retrieval, Semantic Web
P.A. Sharma	Ph.D.	Assistant Professor	Image Processing, Biomatrics, Neural Networks
Praveen Kumar	Ph.D.	Assistant Professor	Image Processing

iii) Scheme of Examination / Instruction – B.Tech. Computer Science Engineering Credit Requirements (2015 admitted batch) :

Details of Credits :

III Semester				IV Semester			
Code	**Course**	**L-T-P**	**Cr**	**Code**	**Course**	**L-T-P**	**Cr**
Core				**Core**			
MAL 208	Probability Theory& Statistical Methods	3-1-0	4	CSL 204	Concepts in Programming Languages	3-0-2	4
CSL 202	Discrete Mathematics and Graph Theory	3-1-0	4	CSL 214	Data Structures and Program Design II	3-1-2	5
CSL 226	Digital Circuits and Microprocessor	3-1-2	5	CSL 222	Computer Organization	4-0-0	4
CSL 213	Data Structures and Program Design I	3-1-2	5	MAL 206	Linear Algebra and Applications	3-1-0	4
CSP 201	Software Lab – I	1-0-2	2	CSP 202	Software Lab – II	1-0-2	2
Elective (Any one)				**Elective (Any one)**			
CSL 224	Introduction to Web Programming	3-0-2	4	CSL 225	Advanced Web Programming	2-0-2	3
ECL 208	Analog Circuits	3-0-2	4	EEL 211	Control Systems	3-0-0	3
				ECL 215	Signals and Systems	3-0-0	3
				MAL203	Numerical Methods and Computation	3-1-0	4
Total No. of Credits			**24**	**Total No. of Credits**			**22-23**

V Semester				VI Semester			
Code	**Course**	**L-T-P**	**Cr**	**Code**	**Course**	**L-T-P**	**Cr**
Core				**Core**			
CSL 307	Theory of Computation	3-1-0	4	CSL308	Software Engineering	3-0-0	3
CSL 313	Design & Analysis of Algorithms	3-1-0	4	CSL315	Database Management Systems	3-0-2	4
CSL 312	Operating Systems	3-0-2	4	CSL316	Language Processors	3-0-2	4
CSP 300	Software Lab – III	1-0-2	2	CSL317	Computer Networks	3-0-2	4
				CSP302	Software Lab – IV	1-0-2	2
Elective (Any Two)				**Elective/HM (Any one)**			
CSL 303	Introduction to OO Methodology	3-0-2	4	MAL304	Financial Mathematics	3-0-0	3
CSL 304	Neurofuzzy Techniques	3-0-2	4	CSL 305	Computer Graphics	3-0-2	4
CSL 305	Computer Graphics	3-0-2	4	HUL301	Technical Communication	3-0-0	3
CSL 318	Business Information Systems (BIS)	3-0-2	4				
Total No. of Credits			**22**	**Total No. of Credits**			**20**

VII Semester				VIII Semester			
Code	**Course**	**L-T-P**	**Cr**	**Code**	**Course**	**L-T-P**	**Cr**
Core				**Core**			
CSL443	System & Network Security	3-0-2	4				
CSD 401	Project Phase I	0-0-4	2	CSD 402	Project Phase – II	0-0-8	4
Elective + OC (Min. credits to be earned 14)				**Elective + OC (Min. credits to be earned 15)**			
CSL 408	Introduction to Embedded Systems	3-0-2	4	CSL 407	Data Mining & Data Warehousing	3-0-2	4
CSL 412	Artificial Intelligence	3-0-2	4	CSL 409	Introduction to Distributed Systems	3-0-2	4
CSL 427	Spatial Databases	3-0-0	3	CSL 410	Topics in Graph Theory	3-0-0	3
CSL436	Information Retrieval	3-0-2	4	CSL 411	Software Project Management	3-0-0	3
CSL 441	Paradigms in Programming	3-0-0	3	CSL 430	Business Intelligence	3-0-0	3
CSL 517	Pattern Recognition	3-0-2	4	CSL 431	Introduction to Cloud Computing	3-0-0	3
CSL 522	Advances in Compilers	3-0-2	4	CSL 437	Enterprise Resource Planning	3-0-0	3
CSL 523	Advanced Computer Architecture	3-0-0	3	CSL 439	Human Computer Interface	3-0-0	3
CSL 530	Top. In Bioinformatics	3-0-0	3	CSL 508	Machine Learning	3-0-0	3
CSL 539	Formal Methods in Program Design	3-0-0	3	CSL 521	Software Architecture	3-0-0	3
ECL 445	Digital Signal Processing	3-0-0	3	PHY4xx	Quantum Computation and Quantum Information	3-0-0	3
MAL 407	Statistics and Optimization Techniques	3-0-0	3	CSL447	Introduction to Parallel Computing	3-0-2	4
MAL403	Numerical Linear Algebra	3-0-0	3	CSL442	Image & Video Processing	3-0-2	4
CSL443	System and Network Security	3-0-2	4	CSL519	Distributed Systems	3-0-2	4
	OPEN COURSE	**3-0-0**	**3**		**OPEN COURSE**	**3-0-0**	**3**
Total No. of Credits			**20**	**Total No. of Credits**			**19**

MAL208: Probability & Statistics for Engineers (DC) (L-T-P-C: 3-1-0-4)
Pre-requisite: None

Syllabus

- Numerical Analysis: Solutions of algebraic and transcendental equations by Iteration method, method of false position, Newton-Raphson method and their convergence.
- Solutions of system of linear equations by Gauss elimination method, Gauss Seidal method, LU decomposition method. Newton-Raphson method for system of nonlinear equations.
- Eigen values and eigen vectors : Power and Jacobi methods.
- Numerical solution of ordinary differential equations: Taylor's series method, Euler's modified method, Runge-Kutta method, Adam's Bashforth and Adam's Moulton, Milne's predictor corrector method.
- Boundary value problems: Shooting method, finite difference methods.
- **Probability theory:**
- Random variables, discrete and continuous random variable, probability density function; probability distribution function for discrete and continuous random variable joint distributions.
- Definition of mathematical expectation, functions of random variables, The variance and standard deviations, moment generating function other measures of central tendency and dispersion, Skewness and Kurtosis.
- Binomial, Geometric distribution, Poisson distribution, Relation between Binomial and Poisson's distribution, Normal distribution, Relation between Binomial and Normal distribution.
- Random processes, continuous and discrete, determinism, stationarity, ergodicity etc. correlation functions, autocorrelation and cross-correlation, properties and applications of correlation functions.

Text Books:

- Jain, Iyengar and Jain : Numerical Methods for Engineers and Scientists, Wiley Eastern
- V.K. Rohatgi and A.K.M. Ehsanes Sateh: An Introduction to Probabability and Statistics, John Wiley & Sons.

Reference Books

- S. D. **Cante** and **C**. **de Boor**, Elementary Numerical Analysis, an algorithmic approach, McGraw-Hill.
- Gerald and Wheatley : Applied Numerical Analysis, Addison-Wesley.
- Spiegel, M.R.; Theory and problems of Probability and statistics; McGraw-Hill Book Company; 1980.
- K.S. Trivedi: Probability Statistics with Reliability, Queuing and Computer Science applications, Prentice Hall of India Pvt. Ltd.

CSL202 Discrete Maths & Graph Theory (DC) (L-T-P-C: 3-1-0-4)

Pre-requsite : None

Syllabus

- Set theory, operations on sets – relation and functions, continuity, partial order, equivalence relations, Peano axioms and induction.
- Mathematical logic, propositions, predicate logic, formal mathematical systems, algebra, homomorphism automorphism.
- Elements of Theory of some algebras, semigroups, monoids, groups.
- Rings, fields, lattices, Boolean Algebra
- Graphs, hypergraphs, transitive closure, trees, spanning trees
- Combinatorics , generating functions, recurrences, Counting theorem and applications.
- Eulerian tours, Hamiltonian cycles, Planar Graphs, Connectivity, Colorability, Line Graphs

Text/References:

- Kolman," Discrete Mathematical Structures for Computer Science", Busby
- Liu C.L" Combinatorial Mathematics", McGraw Hill Book Compan

Course Objectives:

Given the knowledge of Discrete Maths and Graph Theory, a third year CSE student will be able to apply proof techniques of discrete mathematics, and should be able to solve problems on sets, partial orders, groups and graphs.

Course Outcomes:

- Student should be able to use different proof techniques.
- Students would be able to argue about limits by using PigeonHole principle.
- Solve problems based on set theory, Permutations and Combinations, as well as Discrete Probability.
- Students will be able to solve mathematical problems on partial orders, and group theory.
- Students would be able to model and analyze computational problems in graph theoretical framework.

CSL 226 : Digital Circuits and Microprocessor (DC) (DCM) (L-T-P-C: 3-1-2-5)
Pre-requisites : None

Syllabus

- Motivation for digital logic and digital circuits/systems, Analog vs. Digital Systems, basic concepts on SSI, MSI, VLSI circuit classification. Boolean algebra, Postulates and Theorems. Binary Codes: Weighted, non weighted, error detecting and error correcting codes
- Logic Gates, Truth tables, Sum of products, product of sums, Minimization of functions, Karnaugh maps and Simplification of logical functions using Quine-Mclausky method.
- Combinational Circuit: Adders (ripple and carry look-ahead addition) and substractors Decoders/Encoders, multiplexers/ Demultiplexers, code converters, realizing functions using Decoders, Multiplexers
- Sequential Circuits: Flip-flops and latches: D, T, J/K flip-flops, Master Slave Flip flops, shift registers. Counters (Synchronous/Asynchronous), different module counters with reset/clear facility, asynchronous and synchronous design using state and excitation tables. FSM implementation (Sequence Detector) .
- 8085 based Microprocessor organization, memory, I/o organization
- Address decoding, memory, I/O interfacing concepts
- 8085 addressing modes, Instruction set, basic timing diagram
- Assembly language programming, 8085 Interrupts, priorities.
- 8085 Interfacing with PPI –8255

Text Books :

- Kohavi Zvi, “Switching & Finite Automata Theory”, TMH
- M.Morris Mano, “Digital Design”, Pearson education
- Stephen Brown , Vranesic Z, “Fundamentals of Digital Logic with VHDL Design”, TMH
- Bhaskar J, “VHDL Primer”, B.S. Publication
- 8085 Microprocessor by Ramesh Gaonkar

References:

- Barry Brey, The Intel Microprocessors, PHI
- Hall D.V., Microprocessors and Digital Systems, McGraw International.

Course Objectives:

- Understand, analyze and evaluate the performance of various components of digital systems of medium complexity.
- Understating of the architecture and programming issues of microprocessor. Apply the acquired knowledge to interface various programmable devices to the microprocessor and able to build the medium scale microprocessor based system.

Course Outcomes:

- Ability to understand the basic concepts about the digital circuits, microprocessor architecture, programming techniques, and interfacing.
- Ability to apply the acquired knowledge to analyze, design and solve the problems in the concerned area.
- The student should be able to build, evaluate, and propose solutions to the problems in the area of digital circuits, of medium complexity, that are based on SSIs, MSIs, and programmable logic devices

CSL213 : Data Structures & Program Design – I (DC) (DSPD-I) (L-T-P-C: 3-1-2-5)

Pre-requisite: CSL101 : Computer Programming

Syllabus

- Types and operations, Iterative constructs and loop invariants, Quantifiers and loops, Structured programming and modular design, Illustrative examples, Scope rules, parameter passing mechanisms, recursion, program stack and function invocations including recursion, Overview of arrays and array based algorithms - searching and sorting, Divide and Conquer – Mergesort, Quicksort, Binary search, Introduction to Program complexity (Big Oh notation), Recurrence relations. Sparse matrices.
- Structures (Records) and array of structures (records). Database implementation using array of records. Dynamic memory allocation and deallocation. Dynamically allocated single and multi-dimensional arrays.
- Concept of an Abstract Data Type (ADT), Lists as dynamic structures, operations on lists, implementation of linked list using arrays and its operations. Introduction to linked list implementation using self-referential-structures/pointers.
- Stack, Queues and its operations. Implementation of stacks and queues using both array-based and pointer-based structures. Uses of stacks in simulating recursive procedures/ functions. Applications of stacks and queues.
- Files, operations on them, examples of using file.

Reference Books

- The C programming language: Brian Kerninghan and Dennis Ritchie, PHI-EEE (or Pearson)
- How to Solve it by Computer: R. G. Dromey, Pearson Education
- Data Structures & Program Design in C : Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.

Course Objectives :

Given knowledge about structured programming, third semester CSE students should develop skills to create error free and efficient programs; by applying data-structures fundamentals and program analysis techniques.

Course Outcomes :

- Appreciation and practice of structured programming
- Ability to formulate the problem, devise an algorithm and transform into code
- Ability to identify loop invariants and come up with pre/post conditions for a loop and default values. Ability to recognize the errors by analyzing loop invariants and pre/post conditions, without executing the program.
- Ability to analyze the complexity/efficiency of the algorithm and develop ability to improve the same
- Understanding different programing techniques and make an informed choice amongst them
- Understanding of the program/function implementation internally by the OS, concept of program stack etc.
- Understanding and analysis of different sorting algorithms, their advantages and disadvantages, selection of appropriate algorithm as per the properties of given data set
- Appreciation of concept of dynamic memory allocation and its utilization, dynamic data structures and implementation
- Understanding of concept of Abstract Data Type and implementations.
- Ability to communicate about program/algorithm efficiency and recognize a better solution

CSP201 : Software Lab – I (DC) (L-T-P-C: 1-0-2-2)

Pre-requisites : None

Syllabus

- Introduction to Linux/Unix OS - ls, wc, chdir, mkdir, chmod, cd, mv, df, du, netstat, ps, more, set, env, setenv, chgrp, man, rm, rmdir, grep, vi, tar, untar, uuencode, find, cat, history, ping, ifconfig, traceroute,
- Installing Linux (or any variant)
- Introduction to using different tools for identification of possible errors in C program – gdb, concepts of "core dump", backtracing using "bt", using "info" to dump all registers, creating watch-list / watch variables.
- DDD (Data Display Debugger) – introduction and usage.
- IDE for code development
    a. Using DevCpp and/or VisualStudio
    b. Create a project, using multiple .c and .h files with cross-refrences
    c. Setting compiler options and linker options
    d. Understanding different settings
- Unix tools - Awk, sed, Emacs
- Bash scripting – variables, conditionals, loops, finding logged in users
- Parameter passing to C program from shell (argc / argv)
- HTML, XML, XSD and HTML / XML parsing

Assignments :

- Using/Creating/Modifying/copying Files via C programs. Reading- from / writing-to files
- Creating a grade card preparation program from individual subject marks stored in files and creating the result.
- Creating a simple website / homepage
- Creating calculator program
- Translating date / time across different time-zones (with and without daylight saving)

Course Objectives:

To effectively use the Unix programming environment - shell, file system, scripts, pipes, regular expressions, filters, program development tools and to use scripting languages, such as Awk, to automate tasks and write simple programs

Course Outcomes :

- Effectively use the Unix programming environment - shell, file system, scripts, regular expressions, filters, program development tools.
- Automate tasks and write simple programs using scripting languages, such as Awk.
- Develop good programming style, organization, interface, and documentation habits.
- Use of effective procedures and tools for building, debugging, testing, tuning, and maintaining programs.
- Use of tools and write programs to assist in developing programs.

CSL 224 : Introduction to Web Programming(DE) (L-T-P-C: 3 – 0 – 2–4)

Pre-requisites : None

Syllabus

- Internet fundamentals, LAN,WAN, Introduction to common Internet terms, www.
- Basics of networking, DNS, URL, firewall, proxy, DHCP, Web protocols – http and https.
- Introduction to E-Commerce and security issues, digital certificates, certifying authorities, encryption and digital signature, authentication.Viruses and virus scanners, phishing and data stealing.
- Introduction to search engines, web crawlers, social networks, Internet telephony.
- Designing web pages: HTML, forms, DHTML, XML, CSS. Extensible Hypertext Mark up Language (XHTML): XHTML syntax, headings, linking, images, special characters and horizontal rules, lists, tables, forms, internal linking, meta elements.

- Introduction to Web Server – Setting up and configuration of Apache Tomcat server, Accessing pages from another machine.
- Server Side Programming: Introduction to web programming with PHP.
- Client side programming with Javascript: Understanding AJAX using a toolkit.
- Introduction to Python - Statements and Control Flow, Expressions, Methods, Typing, Libraries and Developmental Environment, Web Programming using Python.

Text Book

- Deitel H.M. and P. J. Deitel, Internet & World Wide Web - How to Program, Prentice-Hall.
- Goodman D, Morrison M., JavaScript Bible; Wiley India
- Lutz, Mark, Learning Python (4th ed.). O'Reilly Media

Reference Book

- Garfinkle S., Spafford G; Web Security, Privacy and Commerce; O'Reilly, 2002.
- Atkinson L., Core PHP Programming, Prentice Hall.
- N.P.Gopalan, Akilandeswari, Web Technology, Prentice-Hall.

Course Objectives :

- Aware about different tools for Web Programming.
- Background of working on web.
- Construct efficient web pages with CSS and Javascript.
- Demonstrate competency in the use of common HTML code.
- Able to design efficient client as well as server side scripts.

Course Outcomes :

- How Internet evolve, what are fundamental components and their working
- To understand the basic web tools.
- To design web pages with basic as well as advanced tools.
- To know the e-commerce and to study the various internet threats
- To understand Information security majors and minor issues.

ECL208 : Analog Circuits (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite: None

Syllabus

- Semiconductors, mobility, conductivity, diffusion, continuity equation. Theory of P-N junction, diode characteristics, diode resistances, diode capacitances, switching properties, breakdown of diodes.
- Bipolar junction transistor, transistor configuration & characteristics, breakdown of transistors, power transistors, thermal runaway of transistor, biasing of transistor, FETS, FET characteristics, biasing of FETS.
- Low frequency small signal equivalent circuits of BJT & FETS, Gain, input/output impedances of equivalent circuits of BJTS & FETS, High frequency small signal equivalent circuits of BJT & FETS, difference amplifiers. Power amplifiers.
- Feedback amplifiers, theory of feedback, advantages of negative feedback, feedback configurations, oscillators R-C & L-C, crystal oscillators.
- Operational amplifiers- Properties and characteristics study of typical opamp,
- Performance limitations, application of opamps- summer, inverter, integrator, differentiator, instrumentation amplifier.
- Limiters, log/antilog amplifiers, multipliers, function generators, waveform generators

Text/References:

- Millmann, Halkias : "Integrated Electronics", McGraw Hill
- Millmann, Halkias : "Electronics Devices Circuits", McGraw Hill
- Garud, Jain: , "Electronic Devices & Linear Circuits", Tata McGraw Hill
- Tobey, G:" Operational Amplifier "Tata McGraw Hill
- Gayakwad Ramkant, : "Op-amps & linear integrated circuits ",PHI.

CSL204 : Concepts in Programming Languages (DC) (L-T-P-C: 3-0-2-4)
Pre-requisite: None

Syllabus

- Definition of Programming language . Syntax , semantics. High - level languages. Implementation of high-level languages, Compilers and Software interpreters. Data elements, identifiers binding, binding time, binding identifiers to names, binding of attributes, importance of binding time. Concept of r-value and l-value . Effect of environment on a language. Language paradigms.
- Data type, elementary data type, structured data type, elements of specification and implementation of data type. Implementation of elementary data types : integer, real, character, Boolean and pointer. Implementation of structured data types. Vectors & arrays, records and files. Type checking, type conversion and initialization.
- Evolution of data type concept. Abstract data type, encapsulation. Design and implementation of new data types through subprograms. Subprogram definition and activation, their implementation, parameter passing, generic subprograms.
- Sequence control structures used in expressions and their implementation. Sequence control structures used between statements or group of statements and their implementation.
- Sequence control structures used between subprograms, recursive and non recursive subprogram calls. Data control, referring environment dynamic and static scope, static chain implementation and display implementation.
- Type definition as mechanism to create new abstract data types, type equivalence, type definitions with parameters. Defining new abstracts data types Storage management issues, like static and dynamic allocation, stack based allocation and management, Heap based allocation and management

Text/References:

- Pratt Terence, "Programming Languages, Design and Implementation", PHI
- Sethi Ravi, "Programming Languages", Addison Wesley

Course Objectives :

To enable students to choose right language of implementation for a particular real world problem and also enable them to design a new programming language by considering all its aspects.

Course Outcomes :

- To provide an overview of different programming paradigms
- Improve the background for choosing appropriate programming languages for certain classes of programming problems
- Understand the implementation aspects behind different programming constructs
- Be able in principle to program in an imperative (or procedural), an object-oriented, a functional, and a logical programming language
- Understand the significance of an implementation of a programming language in a compiler or interpreter
- Increase the ability to learn new programming languages
- Increase the capacity to express programming concepts and choose among alternative ways to express things
- Simulate useful features in languages that lack them
- Be able in principle to design a new programming language
- Make good use of debuggers and related tools

CSL214 : Data Structures & Program Design – II (L-T-P-C: 3-1-2-5)

Pre-requisite: CSL1213 : Data Structures & Program Design – I

Syllabus

- Lists - Singly-linked lists, doubly linked lists and circular linked lists. List traversal, insertion, deletion at different positions in the linked lists, concatenation, list-reversal etc. Mergesort for linked lists. Applications of lists in polynomial representation, multi-precision arithmetic, hash-tables etc. Multi linked structures and an example application like sparse matrices. Implementation of priority queues.

- Trees , binary trees, binary trees- basic algorithms and various traversals. Binary Search Trees (BSTs) and insertion, deletion in BSTs. Height-balanced (AVL) trees, insertion/deletion and rotations. Heaps and heapsort. Splay trees.
- Multi-way trees and external sorting - B-trees, Red-black trees. Introduction to B+ trees. Tries. Applications of the above mentioned trees.
- Generalisation of trees to graphs – their representation & traversals. Dijkstra's shortest path algorithm, topological sort, all-pairs shortest paths, minimum spanning trees. Huffman coding. Introduction to network flow problem.
- Introduction to Skip lists, data structures for disjoint set representation.

Reference Books

- Data Structures & Program Design in C : Robert Kruse, G. L. Tondo and B. Leung PHI-EEE.
- Fundamentals of Data Structures in C : E. Horowitz, S. Sahni, and S. Anderson-Freed, University Press
- The C programming language: Brian Kerninghan and Dennis Ritchie, PHI-EEE (or Pearson)

Course Objectives:

Given knowledge about basic and advanced data structures, 4th semester B.Tech. CSE students should develop skills to choose/design/use appropriate data structures based on the problem constraints (time-and-space) and properties of data-set.

Course Outcomes:

- Demonstrate ability to apply knowledge of dynamic data structures like linked-lists, trees and graphs. Ability to formulate the problem, devise an algorithm and transform into code.
- Ability to understand how a newer data structure gets designed as per the requirements and constraints.
- Introduction to different algorithmic programing techniques like greedy algorithms, dynamic programming etc. and ability to make an informed choice amongst them
- Ability to communicate about program/algorithm/data-structure efficiency (time and space) and recognize a better solution. Understanding of properties, advantages and disadvantages of different data structures.
- Developing a real-life application as a s/w project with team work.

CSL222: Computer Organization (DC) (L-T-P-C: 4-0-0-4)

Pre-requisite: None

Syllabus

- Addressing methods, their application in implementation of HLL constructs and data
- structures, instruction formats, expanding opcode method, subroutine linkage in PDP-11 and 68000, zero address machine such as HP3000.
- Processing unit, bus architecture, execution of a complete instruction, sequencing of control signals, microprogrammed control, microinstruction format, microinstruction sequencing, bit slice concept.
- Arithmetic, number representations and their operations, design of fast address, signed multiplication, Booth's Algorithm, bit-pair recording, division , floating point numbers and operations, guard bits and rounding.
- Main memory organization, various technologies used in memory design, higher order memory design, multimodule memories and interleaving, cache memory , concept of cache memory, mapping functions, replacement algorithms. Input-output organization, I/O mapped I/O and memory mapped I/O, Direct Memory Access, interrupts and interrupt handling mechanisms, device identification, vectored interrupts, interrupt nesting, I/O interfaces, synchronous vs. asynchronous data transfer, I/O channels.
- Computer peripherals, I/O devices such as video terminals, video displays, graphic input devices, printers, magnetic disk, magnetic tape , CDROM systems.
- RISC philosophy, pipelining, basic concepts in pipelining, delayed branch, branch prediction, data
- dependency, influence of pipelining on instruction set design, multiple execution units, performance
- considerations, basic concepts in parallel processing & classification of parallel architectures

Text/ References :

- Computer Organization , Hamacher, Carl V. et al, McGraw Hill
- Structured Computer Organization , Tanenbaum A.S, Prentice Hall of India Ltd
- Computer Organization & Design,The Hardware/ Software Interface, Patterson D. A ,Hennessy J. L.

Harcourt Asia, Second Edition.

Course Objectives :

Given the knowledge of digital circuits, and data structures, a fifth semester B. Tech. CSE student will be able to understand the basic principles on which computers work, analyze their performance and appreciate the issues affecting modern processors.

Course Outcomes :

- Students will learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design.
- Students will be able to identify where, when and how enhancements of computer performance can be accomplished.
- Students will learn the sufficient background necessary to read more advance texts as well as journal articles on the field.
- Student will see how to use concepts of computer organization in real-life settings using various PC performance improvements.
- Students will also be introduced to more recent applications of computer organization in advanced digital systems.

MAL206 : Linear Algebra and Applications (L-T-P-C: 3-0-0-6)

Pre-requisite:

Syllabus

- Matrices: Review of Matrix Algebra; Rank of matrix; Row reduced Echelon form; Determinants and their properties; Solution of the matrix Equation Ax = b; Gauss elimination method,
- Vector Space; Subspaces; Linear Dependence/Independence; Basis; Dimension; Linear transformation; Range Space and Rank; Null Space and Nullity; Rank nullity theorem, , Matrix Representation of a lineartransformation; Linear Operators on R^n and their representation as square matrices; Invertible linear operators; Inverse of a non-singularmatrix.
- Eigenvalues and eigenvectors of a linear operator; properties of eigenvalues and eigenvectorsof Hermitian, skew-Hermitian, Unitary, and Normal matrices (including symmetric, skew-symmetric, and orthogonal matrices); Characteristic Equation; Bounds on eigenvalues;Cayley Hamilton theorem, Diagonalizability of a linear operator.
- Inner Product Spaces, Norm; Orthonormal Sets, Gram Schmidt orthogonalisation process; projectionsand least squares approximation.
- Optimization: Modeling and formulation of optimization problems; Least cost and Covex domain; Linear programming and Simplex Algorithm (Big M and Two Phase Method); Duality and the primal dual method.

Text Books:

- Hoffman and Kunze : Linear Algebra, Prentice Hall of India, New Delhi
- Gilbert Strang : Linear Algebra And Its Applications (Paperback) , Nelson Engineering (2007)

Reference Books :

- V. Krishnamoorthy et al : An introduction to linear algebra , Affiliated East West Press, New Delhi P.G. Bhattacharya, S.K. Jain and S.R.
- Nagpaul : First course in Linear Algebra, Wiley Eastern Ltd., New Delhi
- K.B.Datta : Matrix and Linear Algebra, Prentice Hall of India, New Delhi

CSP202 : Software Lab – II (L-T-P-C: 1-0-2-2)

Pre-requisite : CSP 201 Software Lab – I

Syllabus :

- Introduction to advanced unix/linux commands and tools – redirection, piping, redirection, mount, unmount, lynx, ftp, telnet, pipe, su, rlogin, rcp, cron, cksum, cmp, ln,
- Makefile – writing Makefile, compilation via make
- Versioning system – CVS – versioning and branching.
- Profilers –static and dynamic profiling, Code coverage tools, memory leak tools and usage
- Perl / Ruby – For pattern matching via regular expressions (Perl) and other assignments
- Installing Linux (or any variant / Ubuntu) on virtual machine
- Basic assignments in Apache / Mysql
- Installing and using Apache Web server to develop a website,
- Installing and using mysql to store some data
- Introducing Language editors and debuggers– Visual Studio .Net (C++), Eclipse (Java).
- OS related exercises – Accessing iNode, Creation of threads, fork / join, creation of semaphore / mutex, assignments on synchronizing threads. Pthreads and Java threading APIs. Creation of a shell.
- Mini projects like (1) Implementation of garbage collector (2) Software interrupt handler

Course Objectives :

Given the knowledge of Software Lab 1, a fifth semester B. Tech. CSE student will be able to install an OS, operate advanced commands in UNIX, use an IDE for building small projects and implement algorithms and concepts used in operating system.

Course Outcomes :

- Understand the installation of operating systems, web server and databases and their configuration.
- Understand advanced commands of UNIX and automate tasks using scripts.
- Understand software versioning and maintenance.
- Understand programming languages like Perl and Ruby.
- Learn to use IDE like Visual Studio, Eclipse etc. to write large software and debug them.
- Learn to implement and use OS features like fork, threads, mutex etc.
- Build a mini project using the overall concepts learned in the lab.

CSL 225 :Advanced Web Programming (DE) (L-T-P-C : 2 -0 – 2– 3)

Pre-Requisite : CSL 224 Introduction to Web Programming

Syllabus :

- Basic Java, Concept of class, Data Types, Variables, Loops, Arrays, operators, Control statements, Packages and Interfaces.
- Introduction and use of IDE: Eclipse or NetBeans – How to install and use, debugging java programs, search and replace, setting java paths, using java libraries
- Introduction to applets, Deployment of applets, Components, Events, Layout Managers, Windows & Dialogs, Images, Menus.
- Introduction to I/O and Stream Handling, autoboxing, Collections, Generics, Exception Handling.
- Awt & swing frameworks, Servlets, JSP, JSP tags, user-defined tags, concept of session and application.
- Web Programming using MVC model, case studies of frameworks like struts, stripes, etc.
- Advanced Python – System Tools, Script Execution Context, File and Directory Tools, Parallel System Tools, GUI Programming, Graphical User Interfaces, GUI Coding Techniques, Complete GUI Programs,
- Introdcution to Network Scripting, Client-Side Scripting, The PyMailGUI Client, Server-Side Scripting,
- The PyMailCGI Server Tools and Techniques, Databases and Persistence, Data Structures, Text and Language, Python/C Integration

Text Book

- Herbert Schildt, Java: The Complete Reference, Seventh Edition, McGraw Hill Book Company.
- Bruce Eckel,Thinking in Java, Fourth Edition, Prentice Hall.
- Lutz M., Programming Python 4th edition, O'Reilley (2011)

Reference Book

- JSP 2.0: The Complete Reference, Second Edition, McGraw Hill Book Company Phillip Hanna
- Struts: The Complete Reference, McGraw Hill Book Company, 2nd Edition James Holmes

Course Objectives :

- Aware about different tools for Web Programming.
- Background of working on web.
- Construct efficient web pages with CSS and Javascript.
- Demonstrate competency in the use of common HTML code.
- Able to design efficient client as well as server side scripts.

Course Outcomes :

- Aware about advanced tools for web programming.
- Demonstrate important techniques and issues in designing and building enterprise web systems
- Understanding of relevant enterprise web development technologies such as Enterprise Java
- Demonstrate advanced python tools

EEL211 : Control System (DE) (L-T-P-C: : 3-0-0-3)

Pre-requisite: None

Syllabus :

- Introduction to the need for automation and automatic control, use of feedback, broad spectrum of system application.
- Concept of transfer function, closed loop transfer function, Block Diagram Reduction technique. Elementary idea of control system Components, Electrical and Electromechanical .
- Time Response of Systems: First order and second order systems. Concept of gain & time. Constants, Steady state error, type of control system, approximate methods for higher order systems.
- Stability of control systems: conditions of stability, characteristic Equation, Rouths stability Criterian. Frequency Response method of analysing linear system, polar and Bode plot, Elementary ideas about Nyquist stability Criterion, Gain Margin & Phase Margin Concepts.
- State variable method of analysis: Formation of state model in Vector-Matrix form, Relation between transfer function & State variable.
- Sampled Data systems: Introduction, Sample & Hold Circuits, Transforms, Stability of discrete time systems.

Text/ References:

- Nagrath, Gopal ; Control System Analysis
- D' Azzo J.J.; Houpis C.H.; Linear Control System Analysis
- Gopal. M; Control Systems: Principle & Design.
- Kuo B. C.; Automatic Control Systems.

MAL203 - Numerical Methods and Computation (DE) (L-T-P-C :3-0-1-4)

Objective:

The objective of this subject is to make the students aware of the numerical methods for the solution of scientific problems which cannot be solved analytically.

Interpolation :

- Existence, Uniqueness of interpolating polynomial, error of interpolation - unequally spaced data; Lagrange'sformula, Newton's divided difference formula. Equally spaced data : finite difference operators and their properties, Gauss's forward and backward, Sterling's formulae - Inverse interpolation - Hermite interpolation.

- Differentiation : Finite difference approximations for first and second order derivatives.

- Integration : Newton-cotes closed type methods; particular cases, error terms - Newton cotes open type methods - Romberg integration Gaussian quadrature; Legendre formulae.

- Solution of nonlinear and transcendental equations: RegulaFalsi method, Newton-Raphson method, Newton Raphson method for system of nonlinear equations.

- Solution of linear algebraic system of equations: LU Decomposition, Gauss-Seidal methods; solution of tridiagonal system. Ill conditioned equations. Eigen values and eigenvectors : Power and Jacobi methods.

- Solution of Ordinary differential equations:
- Initial value problems: Single step methods; Taylor's, Euler's, Runge-Kutta methods, Implicit RungeKutta methods Boundary value problems: Finite difference methods, Shooting method.

Text Books:

Jain, Iyengar and Jain : Numerical Methods for Engineers and Scientists, Wiley Eastern.
S. D. Cante and C. de Boor, Elementary Numerical Analysis, an algorithmic approach, McGraw-Hill.

Reference Books:

Gerald and Wheatley : Applied Numerical Analysis, Addison-Wesley.
Aitkinson : Numerical Analysis, John Wiley and Sons.

CSL307 : Theory of Computation (DC) (L-T-P-C: 3-1-0-4)

Pre-requisite : None

Syllabus :

- Preliminaries - Sets, operations, relations, transitive closure, countability and diagonalisation, induction and proof methods- pigeon-hole principle and simple applications - concept of language - grammars and production rules - Chomsky hierarchy.
- Regular grammars, deterministic finite automata - non determinism, conversion to deterministic automata- e-closures, regular expressions, finite automata, regular sets.
- Pump lemma for regular sets- closure properties of regular sets, decision properties for regular sets, minimization of automata.
- Context - free languages, parse trees and ambiguity, reduction of CFGS, Chomsky and Griebach normal forms, push - down Automata (PDA), non determinism, acceptance by two methods and their equivalence, CFLs and PDAs – Pumping lemma for context free languages, Closure and decision properties of CFLs.
- Timing machines – variants, recursively enumerable (r.e.) sets, recursive sets, TM as computer of function, decidability and solvability, Halting Problem, reductions, Post correspondence Problem (PCP) and unsolvability of ambiguity problem of CFGs.
- Introduction to recursive function theory - primitive recursive and partial recursive functions Church -Turing thesis - convergence of view points of what "computibility" is : Semi formal treatment.

Text/References:

- Martin John, "Introduction to languages and the theory of computation", TMH
- Motwani Hopcroft, Ullman, "Introduction to Automata Theory, Languages and computation", Pearson Education

Course Objectives :

- Ability to model computation.

- Understand the limitations of each model of computation.
- To know the applicability of model of computation to different problems.
- Develop analytical thinking and intuition for problem solving situations in related areas of theory of computation.
- To know the limitations of computation, i.e. the unsolvability of problems.

Course Outcomes :

- Given a language the student can find the appropriate machine for recognition of that language.
- The student is able to convert machine to grammar and vice versa.
- The student gains knowledge about hierarchy of languages.
- The student develops analytical thinking and intuition for problem solving.
- The student is able to show whether a given problem is solvable or unsolvable.

CSL313 : Design and Analysis of Algorithm (DAOA) (DC) (Credits : 3-1-0-4)

Pre-requisite: Data Structures and Program Design, Probability theory, Mathematics.

Syllabus :

- Mathematical foundations, summation of arithmetic and geometric series, Σn, Σn2, bounding summations using integration, recurrence relations, solutions of recurrence relations using technique of characteristic equation and generating functions.
- Asymptotic notations of analysis of algorithms, analyzing control structures, worst case and average case analysis, amortized analysis, sorting algorithms such as selection sort, insertion sort, bubble sort, heap sort, lower bound proof, elementary and advanced data structures with operations on them and their time complexity.
- Divide and conquer basic strategy, binary search, quick sort, merge sort, Fast Fourier Transform etc. Greedy method - basic strategy, application to job sequencing with deadlines problem, minimum cost spanning trees, single source shortest path etc.
- Dynamic Programming basic strategy, multistage graphs, all pairs shortest path, single source shortest paths, optimal binary search trees, traveling salesman problem.
- Basic Traversal and Search Techniques, breadth first search and depth first search, connected components. Backtracking basic strategy, 8-Queen's problem, graph coloring, Hamiltonian cycles etc. NP-hard and NP-complete problems, basic concepts, non deterministic algorithms, NP-hard and NP-complete, Cook's Theorem, decision and optimization problems, polynomial reduction

Text/ References:

- Introduction to Algorithms : Cormen T.H. et.al : Prentice Hall of India
- Computer Algorithms : Horowitz, Sahani, Rajsekharan , Galgotia Publications Pvt.Ltd
- Fundamentals of Algorithms : Brassard, Bratley , Prentice Hall

Course Objectives:

To enable the students to design good efficient solutions to real world problems and understand the complexity of their solution in terms of memory space and computer time.

Course Outcomes :

- Appreciate the need for analysis of algorithms.
- How to analyze the best-case, average-case and the worst-case running times of algorithms using asymptotic analysis.
- Know the standard design techniques of algorithms and know the conditions in which each particular technique is to be applied.
- Design efficient algorithms for problems encountered in common engineering design situations.
- Know the limitations on the time complexity of algorithms i.e. the theory of
- NPComplete problems.

CSL309 : Operating Systems (DC) (L-T-P-C: 3-0-2-4)

Pre-requisite: Data Structures

Syllabus :

- Introduction, basic h/w support necessary for modern operating systems - Services provided by OS, system programs and system calls - brief discussions of evolution of OS - real time and distributed systems : a brief overview of issues.
- File systems, user interface - disk space management and space allocation strategies - examples from UNIX, DOS, Windows etc - directory structures - disk caching - file system consistency and logs - disk arm scheduling strategies.
- Processes and 3 levels of scheduling - process control block and context switch - goals of scheduling and different scheduling algorithms - threads : user-level and kernel level.
- Memory management techniques - contiguous and non-contiguous - paging and segmentation - translation look-aside buffers (TLB) and overheads - virtual memory and demand paging- page faults and instruction restart - problems of large address spaces - page replacement algorithms and working sets - miscellaneous issues.
- Process cooperation and synchronization - mutual exclusion and implementation - semaphores, conditional critical regions and monitors - classical inter - process communication problems - message passing.
- Deadlocks and strategies for handling them - protection and security issues - access lists, capabilities, cryptographic techniques - introduction to distributed systems.
- Linker and Loader - Concept of static and dynamic relocation, external symbols, design of linker, design of object file for different loading schemes.
- Common Object file format - Structure of object file and executable file, section or segment headers, symbol table, concept of storage class, string various, data types line insert, character, arrays structures.
- Device Drivers - Device programming, system drivers, non system drivers, virtual drivers, Incorporation of driver routines, Basic device driver operation, character and block drivers.

Text/References:

- Tanenbaum A, "Modern Operating Systems", PHI 2nd Ed
- Silberchatz & Galvin, "Operating System Concepts", Addison Wesley

Course Objectives :

Given the knowledge of digital circuits, computer organization and data structures, a sixth semester B. Tech. CSE student will be able to study and apply concepts relating to operating systems, such as concurrency and control of asynchronous processes, deadlocks, memory management, processor and disk scheduling, parallel processing, and file system organization

Course Outcomes :

- Understand the structure and design issues of operating systems.
- Learn about and understand theoretical concepts and programming constructs used for the operation of modern operating systems.
- Understand concepts of OS management domains like process, memory, file systems, storage etc.
- Familiarity with operating systems like Unix.
- Gain practical experience with software tools available in modern operating systems such as semaphores, system calls, sockets and threads.

CSP300 : Software Lab – III (DC) (L-T-P-C: 1-0-2-2)

Pre-requisite: Software Lab-II

Syllabus :

Main theme - Use of open source tools

- Advanced use of Apache Web server: Installing and using Apache Web server in load sharing manner (Configuration of 2 or more server hosting a common website
- Advanced use of MySQL server:
  - Installing and using mysql servers in load sharing manner (Configuration of 2 MySQL instances in master-slave mode).
  - Database operations via programs written in C/C++ or Java.
- Java Native Interface (JNI) – Calling C / C++ code from Java and vice versa.

- Automatic testing tools – Junit, NUnit,
- Advanced use of open source cloud platforms:
    - Integration of gmail with google calendar – from gmail you should be able to schedule an appointment with all the recipients of the mail.
    - Creating a website on Salesforce cloud for tracking inventory from east, west, north, south regions in India separately.
    - Accessing google-map via google-map APIs
- Downloading and Installing Hadoop on 3 to 4 machines and writing a distributed sorting program on the same.
- Creating web-services
    - Using Axis-2 (Java) or gSoap library (C / C++)
    - Introduction to SSL. Use digital certificates to encrypt / decrypt data in transfers
        - Notes - Keytool in Java allows to create / store / manipulate certificates
        - Also, refer www.thawte.com for free download/creation of a certificate
- Introduction to Android Platform and APIs / libraries provided. A sample game / application on Android.
- System utilities - Source code control system, make, link editor, symbolic debugger, pattern matching language like awk
- Learning software engineering tools
    - Design tools - Rational Rose / visio
    - Memory leaks - Purify /
    - Code Coverage tools
    - Testing tools – Loadrunner, Winrunner

Course Objectives :

Given the knowledge of advanced linux and open source tools, a seventh semester B. Tech. CSE student will be able to design and solve advanced programming problems and work in web-based environment.

Course Outcomes :

- The students should get understanding of the advanced Linux operating system. Get to know the necessary tools for Linux administration and script development and learn about special features offered by Linux.
- To inform students about various elements that make the advanced web work, and the way they relate to open-source platforms.
- To substantially strengthen students' programming ability by requiring them to program a number of complex programming problems.
- To provide exposure to a broad range of open-source technologies including encryption and working with databases.

CSL303 : Introduction to OO Methodology (IOOM) (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite: CSL214: Data Structures and Program Design – I

Syllabus :

- Object Oriented Programming, Features of object oriented programming languages like data encapsulation, inheritance, polymorphism and late binding.
- Concept of a class, Access control of members of a class, instantiating a class, static and non-static members, overloading a method.
- Deriving a class from another class, access control of members under derivation, different ways of class derivation, overriding of a method, run time polymorphism.
- Concept of an abstract class. Concept of an interface. Implementation of an interface.
- Exception and exception handling mechanisms. Study of exception handling mechanisms in object-oriented languages
- Introduction to streams, use of stream classes. Serialization and de-serialization of objects.
- Templates, Implementation of data structures like linked lists, stacks, queues, trees, graphs, hash table etc. using object oriented programming languages.
- Introduction to concept of refactoring, modeling techniques like UML, Design patterns.

- **Laboratory Work : Practicals based on above mentioned syllabus**

Text/References:

- Bjane Strostrup, "The C++ programming language", Addison-Wesley
- Herbert Shildt, "C++: The Complete Reference", 4th Edition
- Arnold Ken, Gosling J, "The Java Programming Language", Addison Wesley

- Matt Weisfeld, "The Object-Oriented Thought Process", Pearson
- Cox Brad, "Object –Oriented Programming: An Evolutionary Approach", Addison –Wesley

Course Objectives :

Given the knowledge of data structures, a fifth semester B. Tech. CSE student will be able to design software applications using object oriented techniques and implement the applications using any of the object oriented programming languages like C++ or Java.

Course Outcomes :

- Understand the necessity of encapsulation, data hiding, inheritance, and exception handling.
- Formulate a software application and propose an object oriented design.
- Write generic programs using the standard template library.
- Study and use design tools like UML, design patterns etc.

CSL304 : Neuro Fuzzy Techniques (NFT) (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite : CSL213 & CSL214: Data Structure & Program Design I & II

Syllabus :

- Neural Networks: History, overview of biological neuro-system, mathematical models of neurons, ANN architecture, Learning rules, Learning Paradigms-Supervised, Unsupervised and reinforcement Learning, Learning Tasks, ANN training Algorithms-Single layer perceptron, multi-layer perceptron, Self-organizing Map, Applications of Artificial Neural Networks.
- Introduction to fuzzy set, Operations on fuzzy sets, Fuzzy relation, Fuzzy implication, approximate reasoning, Fuzzy rule-based systems, Fuzzy reasoning schemes, Fuzzy logic controller.
- Implementing fuzzy IF-THEN rules by trainable neural nets. Fuzzy neurons, Hybrid neural networks, Neuro-fuzzy classifiers.

Text/ References:

- Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence; Roger Jang, Tsai Sun, Eiji Mizutani, PHI.
- Soft Computing and Its Applications : R.A. Aliev, R.R. Aliev
- Neural Network: A Comprehensive Foundation; Simon Haykin, PHI.
- Elements of artificial Neural Networks; Kishan Mehtrotra, S. Ranka, Penram International Publishing (India).
- Fuzzy Logic with Engineering Applications; Timothy Ross, McGraw-Hill.
- Neural Networks and Fuzzy Systems: Bar Kosko , PHI.

Course Objectives :

Introduce students to soft computing concepts and techniques and foster their abilities in designing and implementing soft computing based solutions for real-world problems.

Course Outcomes :

- Appreciation of the unified and exact mathematical basis as well as the general principles of various soft computing techniques.
- Appreciation of basic knowledge about the theory and key algorithms that form the foundation for artificial neural network and practical knowledge of learning algorithms and methods
- Appreciation of basic knowledge about the theory and key algorithms that form the foundation for fuzzy logic and become aware of the use of fuzzy inference systems in the design of intelligent or humanistic systems.
- Ability to understand the principles, advantages, limitations and possible applications of learning.
- Ability to identify and apply the appropriate learning technique to classification, pattern recognition, optimization and decision problems.

CSL305 : Computer Graphics (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite: None

Syllabus :

- Basic fundamentals of random scan, raster-scan devices, LCD displays - point and line drawing techniques and algorithms - input/output devices and interactive techniques.
- Polygon filling methods: Seed fill, edge flag algorithm etc. - scan conversion techniques - anti aliasing techniques - clipping algorithms, Polygon clipping, Viewing transformation, Windowing transformation.
- Linear transformation: rotation, scaling, translation in 3D -homogeneous coordinates - normalized device coordinates - windowing and viewporting, Cartesian Coordinates, Word view etc.
- Curve generation - cubic splines, Beziers, blending of curves- other interpolation techniques, Displaying Curves and Surfaces, Shape description requirement, parametric function.
- Review of 3D vector algebra - parallel and perspective projections and transformation - hidden line/ surface elimination - shading and rendering - ray tracing techniques.
- Graphics software packages - segmentation and display files - graphics standards - graphics and computer networks - basic principles of X windows, X terminals, Functions for segmenting display files.

Text / References :

- Procedural Elements of Computer Graphics : Rogers : McGraw Hill.
- Principles of Interactive Computer Graphics : Newman, Sproull,
- McGraw Hill
- Computer Graphics : Hearn, Baker, PHI, India
- Introduction to Computer graphics: Foley, Vanpam, Hughes, Philips,
- Foley, Vanpam, Hughes, Philips

Course Objectives :

- Ability to understand various graphics packages. Understand display, manipulation and storage of pictures and experimental data for proper visualization using a computer.
- Understand internal design of display devices like CRT EGA/CGA/VGA/SVGA monitors, flat panal and plasma displays. Study of frame buffers
- Understand what are 2D transformation and matrices, 3D graphics and viewing w.r.t. 2D screen co-ordinate systems.
  - Study of scan converting line, circle, ellipse, polygon filling and clipping.
  - Understand RGB, HSV and CMY colour space
  - Understand curve and surface representation, hidden surface detection model.

Course Outcomes :

- Students will demonstratean understanding of contemporarygraphics hardware
- Students will create interactivegraphics applications in C++ using oneor more graphics applicationprogramming interfaces , write program functions to implement graphicsprimitives.
- Students will write programs that demonstrate geometricaltransformations
- Understand contemporary graphics principles and graphics hardware, demonstrate geometrical transformations.
- Understand and demonstrate 2D image processing techniques, demonstrate 3D image processing techniques.
- Understand and demonstrate computer graphics animation, Create interactive graphics applications in C++ using one or more graphics application programming interfaces.

CSL 318 : Business Information Systems (BIS) (DE) (L-T-P-C: 3 – 0 – 2 –4)

Pre-requisite: None

Syllabus :

- Introduction to Business Information system : Types of information system, Introduction to e-commerce and related issues. Electronics commerce framework, anatomy of e-commerce applications, consumer applications, organization, Electronic commerce and World Wide Web, Network infrastructure for electronics commerce.

- Electronic Commerce-Frame work, anatomy of E-Commerce applications, E-Commerce Consumer applications, E-Commerce organization applications.

- Modes of electronic commerce: Overview – EDI – Migration to open EDI – E commerce with WWW/Internet – Commerce Net Advocacy – Web commerce going forward.

- Approaches to safe electronic Commerce – Overview – Source – Transport Protocols – Secure Transactions – Secure Electronic Payment Protocol – Secure Electronic Transaction – Certificates for Authentication – Security on Web Servers and enterprise networks.Electronic cash and electronic payment schemes – Internet Monitory Payment and Security requirements – payment and purchase order process – online electronic cash.

- Master card/ Visa Secure electronic transaction: Introduction – Business requirements - Concepts - Payment Processing. Email and Secure Email Technologies for Electronic Commerce: Introduction – The means of Distribution – A model for Message Handling – How Does a Email Work.
- Internet Resources for Commerce: Introduction – Technologies for Web Servers – Internet Applications for commerce – Introduction to Web based computing, three tier architecture, introduction to web page designing, introduction to web servers .Case studies, UML , CORBA/COM , DCOM, JSP/ASP.

Text Books:

- Daniel Minoli, Emma Minoli : Web Commerce Technology Hand Book, McGraw Hill
- Whiteley David: E-Commerce, TMH

References:

- Ravi Kalakotar, Andrew B.Whinston : Frontiers of Electronic Commerce, Addison-Wesley

Course Objectives :

- To understand the different information systems in various business domains
- To understand the e-Commerce applications their modes
- To understand the approaches to safe e-commerce and various transactions involved in it
- To understand the various types of cards for secure electronic transactions
- To analyse and understand the usage of internet resources for commerce

Course Outcome

- To obtain sound knowledge in the theory, principles and applications of Business Information System.
- Design and develop web based transactional system given their specifications and within performance and cost constraints.
- Acquire and understand web based business models, e-payments and payment gateways.
- Perform experiments using different technologies of Business Intelligent System.

CSL308 : Software Engineering (DC) (L-T-P-C: 3-0-0-3)

Pre-requisite: CSL 201 and CSL 101

Syllabus :

- Software Engineering Process & Management : Generic view, Capability Maturity Model, Process models-waterfall, evolutionary, incremental etc., unified process, agile view, project management, metrices estimation, project scheduling, risk management.
- Software engineering Principles and Practice : Communication, planning and modeling practices, system engineering and modeling, business process engineering requirement analysis, system analysis- flow oriented and class oriented modeling using data modeling concepts.
- Software Design Engineering : Design Concepts : Abstraction Architecture, pattern modularity, information hiding, design classes, refacoring etc., Design of web application, architectural design, component level design, user interface design.
- Software Testing and Quality Management : Testing strategies, testing for object oriented software testing for web applications, validation testing etc. Black box testing, white box testing, Basis path testing. Testing for specialized environments, architectures and application. Quality concepts, quality assurance, software reviews, statistical quality assurance.
- Software configuration management and advance topics : Elements of configuration management system, process configuration for web engineering, component-based development, clean room software engineering, formal methods, software reengineering, Software Maintenance

Text/References:

- Software Engineering by Ian Sommerville ; Pearson Ed
- Software Engineering: A Practitioner's Approach by Roger Pressman ; Tata-McGraw Hill

Course Objectives :

- To develop an ability to look at the Computer Science discipline from Software Engineering Systems perspective
- To develop understanding of generic processes of software development and learn different techniques and methodologies used in development of large software systems
- To develop analytical ability to employ various strategies in selecting from various models of different stages of software development
- To develop ability to understand role of teamwork in software development and ability to effectively communicate in written forms at various stages of the developmental process
- To develop ability to pursue life-long learning as required for software developers for different skills at conceptual, strategic, and operational level

Course Outcomes :

- to look at the large scale software development from a broader perspective, and function in multidisciplinary teams
- to apply knowledge gained in the course to practical software development situations in methodical way
- to design software systems to meet desired needs with realistic constraints
- to communicate effectively in software development activities
- to get an idea about contemporary issues in Software development and engage in life-long learning, understand professional and ethical responsibility

CSL315 : Database Management Systems (DC) (L-T-P-C: 3-0-2-4)

Pre-requsite : Data structure and algorithm, operating systems

Syllabus :

- Database system concepts and Architecture - concept of relational database, Relational data model , Relational algebra, SQL-the relational database standard, ER and EER model.
- Database design theory - Functional dependencies and normalization, relational database design algorithms, practical database design and demoralization, Relational constants, programmatic ways for implementing constraints, triggers, Chase algorithm.
- Physical database design - Concept of physical and logical hierarchy, storage structures like cluster, index organized table, partitions, various table storage parameters and block storage parameters, concept of index, B-trees, hash index, function index, bitmap index.

- Process and memory management in database - Various types of tasks in database, database buffer management, log buffer management code reuse, concept of two tier and N-tier architecture, data dictionary and catalog information database recovery technique. Arier Algorithm for recovery.
- Query optimization and performance tuning - Various techniques for query optimization, strong and weak equivalence, cost base optimization, Use of different storage structures in query optimization.
- Transaction Processing - Transaction and system concepts, Desirable properties of transaction, Schedules and recoverability, serializability of schedules, concurrency control, lock base protocols and time stamp based protocols, read consistency.

Text & References :

- Fundamentals of Database Systems : Elmasiri and Navathe, Addisson Wesley, 2000
- Principles of Database Systems : Ullman , Golgotia Publications 1988

Course Objectives :

- Understand how to perform basic operations with DBMS.
- Understand advance concepts like analytical functions, ROLLUP and CUBES, multitable DML operations.
- Understand database design process using ER diagram and Normalization.
- Understand validation framework like integrity constraints, triggers and assertions.
- Understand ACID properties and their implementation.
- Understand concurrency control mechanism using lock based protocols and timestamp based protocols.
- Understand various storage structures and query optimization.

Course Outcomes :

- CO1:To obtain sound knowledge in the theory, principles and applications of database management system.
- CO2:Design and develop data model given their specifications and within performance and cost constraints.
- CO3:Acquire and understand new knowledge, use them to develop data centric application and to understand the importance of lifelong learning.
- CO4:Perform experiments in different disciplines of database management system.

CSL316: Language Processors (DC) (L-T-P-C: 3-0-2-4)

Pre-requisite: CSL 307 : Theory of Computation

Syllabus :

- Introduction to compilers, compilers and translators, phase structure of a typical compiler, Number of passes, ideas about lexical analysis, syntax analysis, code optimization and code generation, design of lexical analyzer.
- Syntax specification of programming languages**,** Design of top-down parser, bottom up parsing technique, LR parsing algorithm, Design of SLR, LALR,LR parsers. Dealing with ambiguity of the grammar.
- Study of syntax directed definitions and syntax directed translation schemes as notational frame work to specify the translations. Using syntax directed translation schemes for translation of expressions, controls structures, declarations , procedure calls.

- Storage allocation and run time storage administration, symbol table management, Error detection and recovery, error recovery in LR parsing, error recovery in LL parsing, Automatic error recovery in YACC.
- Introduction to Important code optimization techniques, loop optimization, control flow analysis, data flow analysis, setting up data flow equations to compute reaching definitions, available expressions, Live variables. Problems in code generation , simple code generator code generation from DAG, Peephole optimization
- Assembler, Macroprocessor - Concept of assembler, design of single pass and two pass assembler, forward reference, design of output file of assembler, concept of macro, macro call within macro, macro definition within macro, recursive macro calls, design of macro processor.
- Lexical Analysis - Role of lexical analyzer, recognition of tokens, tool for study of lex.

Text/ References :

- Principles and practice of compiler writing : Aho, Sethi , Ullman , Addison Wesley
- Compiler Design in C : Alan Holub , PHI

- Crafting a compiler : Fischer and LeBlanc , Addison Wesley
- Principles of Compiler Design : Aho A. V., Ullman J.D , Narosa Publishing House.

Course Objectives :

Given the knowledge of languages and grammars with tools to process them, a sixth semester B. Tech. CSE student will be able to solve problem of creating a language processing tool and understand ways of its optimization.

Course Outcomes :

- This course introduces students how a compiler translates the higher level language into machine language.
- The student should be able to analyze issues associated with the implementation of higher-level programming languages.
- To inform students about different parsing techniques, techniques to generate intermediate code and different optimization techniques. Understanding of compiler optimization techniques would enable students to write reasonably efficient programs.
- The students will also appreciate the need of understandable error reports, accurate and reliable object code, and efficient use of in-memory data structures.

CSL317 : Computer Networks (DC) (L-T-P-C: 3-0-2-4)

Pre-requisite : Signals and Systems

Syllabus :

- Computer Networks, evolution of Computer Networks, application of Computer Networks.
- Layered Network Architecture: requirement for layered approach, basic concept of layering in the network model, define entities, protocols, interface in networking context, ISO's OSI Reference Model, functions of the seven layers of OSI Model , TCP/IP model, difference between OSI and TCP/IP model
- Data and Signal: Define data, signal. Time domain and frequency domain representation of signal, bandwidth of a signal and medium, Sources of impairment, Attenuation, distortion, noise, data rate Limits and Nyquist bit rate, between Bit Rate and Baud Rate, Sources of noise. FDM and TDM, synchronous and asynchronous TDM
- Transmission Media: Various Transmission Media - guided and unguided media, characteristics of the popular guided transmission media: Twisted-pair, Coaxial cable, Optical fiber, Sources of transmission impairment, Shannon Capacity
- Network Topology- what is network topology, characteristics of the following topologies: Mesh, Bus, Star, Ring, Tree, Unconstrained

- Medium Access Control (MAC) Techniques - goals and requirements of Medium Access Control (MAC) techniques, key issues related to MAC techniques, Classify various contention based techniques such as ALHOA, CSMA, CSMA/CD and CSMA/CA. MAC techniques: Polling, Token passing. FDMA, TDMA, CDMA.
- IEEE 802 LANs - basic characteristics of LANs, operation of IEEE 802 LANs , 802.3 - CSMA/CD-based (Ethernet), 802.4 – Token bus-based, 802.5 – Token ring-based, Compare performance of the three LANs
- Introduction of High Speed LANs, Fast Ethernet and Gigabit Ethernet, wireless LANs
- Need for wireless LAN, limitations and challenges of wireless LAN IEEE 802.11 WLAN -Transmission media, Topology, Medium Access Control, Security
- Interfacing to the media and synchronization: modes of communication, Asynchronous and Synchronous modes of communication. Error Detection and Correction: need for error detection and correction, simple parity check, 2-D parity check, checksum, cyclic redundancy check., Hamming's code
- Flow Control and Error Control : need for flow and error control, Stop-and-wait flow control, Sliding-window flow control, Stop-and-wait ARQ, Go-back-N ARQ, Selective-repeat ARQ, Selective-repeat ARQ.
- HDLC: how HDLC works, piggybacking in HDLC, data transparency in HDLC

- Switching Techniques: Circuit Switching - need for circuit switching , how circuit switching takes place, space-division and time-division switching, Packet Switching - need for packet switching, how packet switching takes place, difference between virtual-circuit and datagram type packet switching, Message switching, Compare circuit switching, packet switching, message switching.
- Need for internetworking, Introduction of internetworking devices- Hubs, Switches, Bridges, Router, Gateways
- Internet Protocol (IP): different classes of IP addresses, concept of subnet masking, sub-netting super-netting, network address translation table, ARP/RARP protocol, fragmentation and reassembly, ICMP protocols, key features of IPv6
- Transport layer: Connection establishment and release – timer management - multiplexing - flow control working of TCP and UDP. QoS parameters,
- ATM network, ATM signaling, PNNI routing I ATM

- Application Layer Protocols: DNS, Telnet, ICMP, RPC, SMTP, FTP, SNMP
- Routing Algorithms: Adaptive routing, Non-adaptive routing, Dijkstra's SP algo, flooding, flow based, distance vector routing, linked state routing, RIP- routing information protocol, OSPF - (Open shortest path first), BGP - Border gateway protocol: operation of the BGP protocol
- Congestion Control: causes for congestion, effects of congestion , various open-loop and close-loop congestion control techniques: The leaky bucket algorithm , The token bucket algorithm, Choke packets, Load shedding, jitter control, distinguish between flow and congestion control

CN Lab :

Programs based on

1. Using TCP sockets or Network socket programming
2. Client-server application for chat
3. PC to PC file transfer using serial port
4. Implementation of Shortest path routing
5. Implementation of Sliding Window Protocol
6. Implementation of Address Resolution Protocol
7. Implementation of Open Shortest Path First Protocol
8. Using n/w simulators like: NS2, DLC/DLL simulator
9. Implementation of multi thread client server application.
10. Implementation of TCP/IP Echo
11. Using simple UDP
12. Using RPC / RMI

Text:

- Tanenbaum A. S, "Computer Networks", PHI 4th Edition
- James F. Kurose and Keith W. Ross : Computer Networking A Top-Down Approach Featuring the Internet, 3rd Edition.
- Peterson, Davie, "Computers Networks", Elsevier 3rd Edition
- William Stallings, "Data and Computer Communications", PHI 6th Edition

References:

- Simon Haykin, "Communication Systems", John Wiley 4th Edititon
- Douglas Comer, "Computer Networks and Internets", Addison Wesley 2nd Edittion
- Peterson, Simon, "Computer Networks: A Systems Approach", Pearson Education, Asia
- Behrouz A Forouzan : Data Communication and Networking, 4th Edition.

Course Objectives :

- Understand TCP/IP and ISO OSI network layer.
- Study of various layers functions. Understand LAN, WAN, MAN and VLAN.
- Understand practically working of L2 switch, L3 switch and Routers and their functionality.
- Practically understand the working of hubs/switches/routers and security.
- Can evaluate performance of various MAC layer protocol.
- Ability to write program using socket programming.
- Ability to implement protocol for two systems and for a group of systems.
- Performance evaluation of protocols for AdHoc networks.
- Performance evaluation of EPABX, ISDN system and VOIP.
- Evaluation of protocol on QualNet software.

Course Outcomes :

Student will be able to

- Defining, using and implementing Computer Networks and the basic components of a Network system , explain the importance of data communications, how communication works in data networks and the internet, recognize the different internetworking devices and their functions.
- Explain the role of protocols in networking, Analyze the role and services and features of the various layers of data networks, analyze the features and operations of various application layer protocols such as Http, DNS, Telnet, FTP and SMTP.
- Knowing and Applying pieces of hardware and software to make networks more efficient, faster, more secure, easier to use, able to transmit several simultaneous messages, and able to interconnect with other networks, differentiating the various types of network configurations and applying them to meet the changing and challenging networking needs of organizations,defining and analyzing the circuits available for voice and data networks, their transmission speeds (bandwidth), and how they are packaged for commercial use.
- Definining the different protocols, software, and network architectures, defining the concept of local area networks, their topologies, protocols and applications , analyzing why networks need security and control, what errors might occur, and how to control network errors.

CSP302 : Software Lab-IV (DC) (L-T-P-C: 1-0-2-2)

Course outline:

- Cloud systems, Cloud technologies, hands-on project on Cloud infrastructure
- Hadoop usage and understanding of map-reduce paradigm
- No-SQL database storage, and query
- XML- XML Namespace, XML Schema, XML Schema Design Pattern, XPath, XSLT, XSLT Advanced Usage,
- Service oriented architecture, and web services
- Web services styles:
  - SOAP-standard web services that use XML-formatted SOAP messages
  - RESTful web services, which leverage HTTP-standard methods and use XML
- Working with JavaScript and web services
- Create and use SOAP-standard web services
- Security and authentication
- Study and write software that interoperates with a well-known existing public web services application programming interface
- Survey of web services implementations on other platforms (e.g. PHP, Java)

Course Objective: Students will be able

- To learn how and why Cloud systems work and details of various cloud technologies
- Understanding installation and use of Hadoop
- Understand the installation and use of no-sql database
- To create and use web services, as well as study the web services standards, technologies, platforms, and development tools.
- How to effectively secure web services.

Course Outcomes:

- Install and use OpenStack/ CloudStack open source cloud platform
- Create an free account and use commercial cloud offering like Amazon/ BlueMix/ SalesForce/ Google/ Azure/ Rackspace
- Examine the design of task and data parallel distributed algorithms for Clouds and use them to construct Cloud applications.
- Install Hadoop and understand map-reduce. Create an application that uses map-reduce.
- Install and use no-sql database like mongodb/cassendra/hbase/neo4j
- Understand and describe the standards and technologies of modern web services implementations.
- Effectively use market-leading development tools to create and use web services.
- Identify and select the appropriate framework components in the creation of web service solutions.
- Students will be able to coordinate the use of information security tools within an organization.
- Securing web services.

MAL304–Financial Mathematics (DE) (L-T-P-C: 3-0-0-3)

Prerequisite:

Probability and Statistics for Engineering (MAL 205) and Operations Research (MAL 301).

Objective:

- The main objective of this course is to introduce the financial mathematics and is applications to marketing.
- Some Basic definitions and terminology.
- Basic Theory of Option Pricing: Single and Multi-Period Binomial Pricing Models, Cox-Ross-Rubinstein (CRR) Model, Black-Scholes Formula for Option Pricing as a Limit of CRR Model.
- Brownian and Geometric Brownian Motion, Theory of Martingales. Stochastic Calculus, Stochastic Differential Equations, Ito's Formula to Solve SDE's.Applications of Stochastic Calculus in Option Pricing.
- Mean-Variance Portfolio Theory: Markowitz Model of Portfolio Optimization and Capital Asset Pricing Model (CAPM). Limitations of Markowitz Model and New Measures of Risk.
- Interest Rates and Interest Rate Derivatives: Binomial Lattice Model, Vasicek, Hull and White Models for Bond Pricing.

Text books:

- D. G. Luenberger: Investment Science, Oxford University Press.
- M. Capińsky and T. Zastawniak: Mathematics for Finance: An Introduction to Financial Engineering, Springer.
- Thomas Mikosch: Elementary Stochastic Calculus with Finance in view, World Scientific.
- Suresh Chandra, S. Dharmaraja, AparnaMehra, R. Khemchandani: Financial Mathematics: An Introduction, Narosa Publishing House.

Reference Books:

- S. E. Shreve: Stochastic Calculus for Finance, Vol. I & Vol. II, Springer.
- Sean Dineen: Probability Theory in Finance: A Mathematical Guide to the Black-Scholes Formula, American Mathematical Society, Indian edition.

HUL301 : Technical Communication (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite: Communication Skills I – Year

Syllabus

- Defining technical writing – Basics of Technical Communication – Barriers to Communication –Objectives Audience Recognition and Involvement
- Correspondence: Memos - Letters – Job Search
- Visual Appeal – Document Design – Graphics – Electronic Communication – Email – Online help and Websites
- Technical Application: Descriptions – Instructions and User Manuals
- Report Strategies: Research - Summary – Types of Reports
- Oral Presentations and Group Discussion
- Grammar, Punctuation, Spellings and Mechanics of Writing

Text / References :

- Gersen S J and S M Gersen, Technical Writing: Process and Product, Pearson Education Asia
- Rutherford : Basic Communication Skills for Technology, Pearson Education Asia
- Lesikar et al : Lesikar's Basic Business Communication, Tata McGraw Hill
- Shirley Taylor: Communication for Business, Pearson Education Asia

Course Objectives :

- To sharpen the technical writing skills of the students
- To expose them to modern styles of oral communication
- To enable them to use appropriate language

Course Outcomes :

- At the end of this course students will be able to write technical reports, proposals Letters, applications etc.
- Students will be able to make effective presentations.
- Students will be confident to take part in the group discussion.

CSL443: System &Network Security (DC) (L-T-P-C: 3-0-2-4)

Syllabus :

Cryptography

- Classical Ciphers: Affine, Playfair , Hill Cipher;
- Modern Block and Stream Ciphers: DES, AES, RC4, A5/1; Block Modes of Operation: ECB, CBC, CFB, OFB, CTR
- Asymmetric Key Cryptosystems: RSA; Digital Signatures: DSS;

Message Integrity and Authentication

- Hash and MAC: SHA-512

Key Management:

- Kerberos , Digital Certificates, PKI

Entity Authentication

- One-Way Authentication, Mutual Authentication, Dictionary Attacks, Centralized Authentication, The Needham-Schroeder Protocol, Zero knowledge protocols

Network &Transport Layer Security and Applications

- IPSec; SSL/TLS
- The EMV payment protocol

Software Vulnerabilities

- Phishing, Buffer overflow, cross site scripting and SQL injection

System Security

- Viruses, Worms, and other Malware: Virus and Worm Features, Internet Scanning Worms, Mobile Malware and Botnets

Textbooks :

1. Forouzan, Cryptography and Network Security, TMH
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Hand-book of Applied Cryptography, CRC Press.

References :

1. Bruce Schneier Applied Cryptography , 2nd Edition John Wiley & Sons 1996
2. Douglas Stinson Cryptography Theory and Practice CRC Press 1995
3. Pfleeger and Pfleeger, Security in Computing, Pearson

Course Objectives :

This course will explore the International Standards Organizations Open System Interconnect (ISO OSI) network stack and discuss common security weaknesses, vulnerabilities, attack methods, and mitigation approaches. This course will provide a comprehensive list of security issues related to general networking design and development.

Course Outcomes :

- Develop an understanding of information assurance as practiced in computer operating systems, distributed systems, networks and representative applications.
- Gain familiarity with prevalent network and distributed system attacks, defenses against them, and forensics to investigate the aftermath.
- Develop a basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.

- Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.

CSL408 : Introduction to Embedded Systems (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite Course : CSL312 : Operating Systems, Computer Architecture Organization, Computer Netwoks.

Syllabus :

- Embedded systems overview, Hardware / Software co-design, Examples of embedded systems, Components of Embedded systems – sensors, actuators, micro-controller processor Technology, IC technology, Issues in Design Technology
- Scheduling Paradigms for real time systems - Static Priorities, Static Schedules, Dynamic Scheduling, Pre-emptive, Non-pre-emptive, Rate Monotonic, EDF
- Real World Issues like - Task Assignment, CPU utilization, Blocking,Unpredictability, performance Measures
- Hard & Soft Real Time Systems, Real Time Operating System – concepts, RTOS services, capabilities, Resource Management
- Programming Languages for Embedded Systems - Desired Language Characteristics,
- Tools for building Embedded systems, Embedded Software Development Methodology
- Issues in real time databases, real time communications, Fault Tolerant Techniques –
- Fault Types, Detection, Recovery

Text/References:

- Frank Vahid, "Embedded System Design- A Unified Hardware / Software Introduction", John Wiley & Sons
- Krishna C.M. , Kang G. Shin, "RTS : Real Time Systems", McGraw Hill
- David Simon, "An Embedded Software Primer", Addison Wesley, -2000

Course Objectives :

- Ability to understand and design embedded hardware, challenges in designing and implementing real rime systems.
- Understand real time task scheduling, resource sharing and dependencies among real-time tasks
- Ability to make choices from among available embedded hardware and OS for any specific real time systems.
- Ability to present different design decisions made for real time system implementations, and their experimental evaluation.

Course Outcomes :

Student will be able to

- Describe the differences between the general computing system and the embedded system, also recognize the classification of embedded systems, become aware of the architecture of the ARM processor and its programming aspects (assembly Level)
- Become aware of interrupts, hyper threading and software optimization, design real time embedded systems using the concepts of RTOS, analyze various examples of embedded systems based on ARM processor
- To work on design and development of protocols related to real-time communication, to develop real-time algorithm for task scheduling , the students will be able to design, simulate , built and debug complex combinational and sequential circuits based on an abstract functional specification.

CSL412 : Artificial Intelligence (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite Course : CSL313 :Analysis of Algorithms

Syllabus :

- Introduction: What is AI? , History, Overview, Intelligent Agents, Performance Measure, Rationality, Structure of Agents, Problem-solving agents, Problem Formulation, Uninformed Search Strategies
- Informed (Heuristic) Search and Exploration, Greedy best first search, A* search, Memory bounded heuristic search, Heuristic functions, inventing admissible heuristic functions, Local Search algorithms, Hill-climbing, Simulated Annealing, Genetic Algorithms, Online search

- Constraint Satisfaction Problems, Backtracking Search, variable and value ordering, constraint propagation, intelligent backtracking, local search for CSPs, Adversarial Search, Games, The minimax algorithm, Alpha-Beta pruning, Imperfect Real-Time Decisions, Games that include an Element of Chance
- Knowledge Based Agents, Logic, Propositional Logic, Inference, Equivalence, Validity and Satisfiability, Resolution, Forward and Backward Chaining, DPLL algorithm, Local search algorithms, First Order Logic, Models for first order logic, Symbols and Interpretations, Terms, Atomic sentences, complex sentences, Quantifiers, Inference in FOL, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution
- Planning, Language of planning problems, planning with state-space search, forward and backward state-space search, Heuristics for state-space search, partial order planning, planning graphs, planning with propositional logic
- Uncertainty, Handing uncertain knowledge, rational decisions, basics of probability, axioms of probability, inference using full joint distributions, independence, Baye's Rule and conditional independence, Bayesian networks, Semantics of Bayesian networks, Exact and Approximate inference in Bayesian Networks

Text / References :

- Artificial Intelligence a Modern Approach : Russel and Norvig , Pearson Education, 2nd
- Artificial Intelligence – A Practical Approach : Patterson , Tata McGraw Hill, 3rd

Course Objectives :

Given the knowledge of algorithms and data structures, a seventh semester B. Tech. CSE student will be able to design intelligent agents that solve problems using exploratory search techniques, perform reasoning using formal logic, and work in uncertain environments using probabilisitic reasoning techniques.

Course Outcomes :

- Formulate problems so that exploratory search can be applied.
- Implement optimal, heuristic and memory bounded search techniques.
- Represent knowledge using formal logic and design algorithms to work in a semi-observable environment using logical reasoning.
- Design and develop practical algorithms for solving real-life planning problems.
- Implement probabilistic reasoning techniques to work in uncertain environments.

CSL412 : Spatial Databases (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite: Data structures and Program Design I, Data Structures and Program Design II and Databases, Design and Analysis of Algorithms.

Syllabus :

- Introduction and Motivation (1 week)
  - Application Domains of Geographical Information Systems (GIS), Common GIS data types and analysis, OGC standards and reference geometry model.
- Models of Spatial Data (3 weeks)
  - Conceptual Data Models for spatial databases (e.g. pictogram enhanced ERDs). Logical data models for spatial databases: raster model (map algebra), vector model (OGIS/SQL1999).
- Spatial query languages (2 weeks)
  - Need for spatial operators and relations, SQL3 and ADT. Spatial operators, OGIS queries.
- Spatial storage and indexing Methods (3 weeks)
  - Clustering methods (space filling curves), Storage methods (R-tree, Grid files), Concurrency control (R-link trees), Compression methods for raster and vector data Spatial indexing.
- Processing spatial queries , Optimization (3 weeks)
  - Spatial selection, joins, aggregates, nested queries, buffers. Strategies for range query, nearest neighbor query, spatial joins (e.g. tree matching), cost models for new strategies, impact on rule based optimization, selectivity estimation.
- Mining spatial databases (2 weeks)
  - Clustering, Spatial classification, Co location patterns, Spatial outliers.
- Typical Assignments:
  - Solving spatial queries, implementing clustering techniques, implementation of spatial classification.

Text Books and / or other required material :

- Spatial Databases – A Tour by Shashi Shekhar and Sanjay Chawla

Reference Books :

- Spatial Databases with Applications to GIS, P. Rigaux, M. Scholl, A. Voisard, Morgan Kaufmann, 2002

Course Objectives

Enable students to understand various aspects of spatial databases, their implementation and mining them and be able to write and optimize spatial queries.

Course Outcomes:

- Understanding basics of Spatial Databases.
- Understanding the implementation issues of spatial datbases.
- Understanding various models for spatial data.
- Understanding spatial operators and OGIS queries in detail.
- Understanding spatial queries and optimizing them.
- To understand the basic concepts of mining spatial data.

CSL436 : Information Retrieval (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite : CSL 213: Data Structures and Program Design I and Data Structures and Program Design II

Syllabus :

- Boolean retrieval (1 week)
- the term vocabulary and postings lists, (1 week)
- Dictionaries and tolerant retrieval, (1 week)
- Introduction to index-construction and index-compression (1 week)
- Scoring, term weighting and the vector space model (2 weeks)
- Computing scores in a complete search system, Evaluation in information retrieval, Introduction to Relevance feedback and query expansion. (2 weeks)
- Probabilistic information retrieval, review of basic probability theory, the probability ranking principle, the binary independence model (2 weeks)
- Language models for information retrieval, Language modeling versus other approaches to IR, Text classification and Naive Bayes, Bayesian Network approaches to IR. (2 weeks)
- Vector space classification, Support vector machines and machine learning on documents, Fl at clustering, Hierarchical clustering, Matrix decomposition and latent semantic indexing. (2 weeks)
- Introduction to Web search basics, Web crawling and indexes, Link analysis (1 weeks)
- Typical Assignments : Based on techniques studied, implementation of those techniques, study of research papers.

Text Books

- An Introduction to Information Retrieval: Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Cambridge University Press, Cambridge, England, 2009
- Information Retrieval: Implementing and evaluating search engines: Stefan Büttcher, Charles L. A. Clarke, Gordon V. Cormack, MIT Press, 2010

Reference Books

- Information Retrieval: Algorithms and Heuristics : David A. Grossman, Ophir Frieder, Springer.
- Information Retrieval: Data Structures and Algorithms by Frakes, Pearson.

Course Objectives

Enable students to understand the various aspects of an Information retrieval system and its evaluation and to be able to design such a system from scratch

Course Outcomes

- Understanding the basics of Information retrieval like what is a corpus, what is precision and recall of an IR system
- Understanding the data structures like Inverted Indices used in Information retrieval systems
- Understanding the basics of web search
- Understanding the different techniques for compression of an index including the dictionary and its posting list
- Understanding the different components of an Information retrieval system
- Developing the ability of develop a complete IR system from scratch

CSL441: Paradigms in Programming (L-T-P-C: 3-0-0-3)

Pre-requisite : CSL204: Concepts in Programming Languages , CSL303: Introduction to OO Methodology

Syllabus :

- Review of the program development process, Issues in program design, Structured programming, Data and control abstractions, Programming with assertions. Reasoning about programs and proving correctness of programs. Ideas behind imperative, applicative, object oriented and logic programming paradigms such as typing, expressions, pure functions, recursion, higher order functions, encapsulation, inheritance, goal satisfaction, backtracking, unification. Some of the ideas behind the implementation of the paradigms. Course to be centered around problems and applications that demonstrate the main themes.

- Functional programming languages and lambda Calculus, Typed Programming Languages, Aspect Oriented Programming, code refactoring – examples can be illustrated using languages like Haskell, F#, C#, Java, as and where applicable.

Text Books

- Harold Abelson, Gerald Jay Sussman and July Sussman, Structure and Interpretation of Computer Programs, 2nd edition, The MIT Press, 1996.
- David A. Watt, Programming Language Concepts and Paradigms, Prentice-Hall, 1990.

Reference Books

Rajeev Sangal, Programming Paradigms in Lisp, McGraw Hill, 1991.

Course Objectives :

- Ability to understand the data and control abstractions
- Appreciation of the utility of assertions in reasoning about programs and in proving correctness
- Exposure to different programming paradigms like imperative, declarative, functional and logic paradigms and their usage in different applications
- Understanding Functional Programs and their implementation using lambda calculus
- Understanding Logic based programs and their implementation strategies

CSL517 : Pattern Recognition (DC) (L-T-P-C: 3-0-2-4)

Pre-requisite: Probability theory, Linear Algebra

Syllabus :

- Applications of pattern recognition, statistical decision theory, probability of events, Random variables, Estimation of parameters, Minimum Risk Estimators.
- Bay's Theorem, conditionally independent features. Decision boundries, Estimation of error rates, characteristics curves.
- Histograms, Kernel and window parameters, Nearest Neighbour classification techniques, Adaptive Decision boundries, clustering.
- Artificial Neural Networks, Nets without hidden layers and with hidden layers. The back propagation Algorithm, Hopfield Nets.
- Gray level scaling transformations. Equalization smoothing transformations. Edge detection Logarithmic Gray scale level scaling.
- Scene segmentation and labelling , counting objects, Hough Transforms, Eigenvector line fitting , Fourier transforms.

Text / References :

- Pattern Classification : Richard O.Duda, Peter E.Hart, David G.Shork, John Wiley & Sons 200, 2nd Edittion
- Pattern Recognition and Image Analysis , Earl Gose, Richard Johnsonbough , Steve Jost . Prentice Hall
- of India
- Pattern Recognition and Image Processing : Sing Tze bow; Marcel Dekker

Course Objectives :

- Understand how to build classifiers using known probability distribution.
- Understand how to build classifiers using non parametric methods.
- Understand how to build classifiers using perceptron model and syntactic methods
- Understand how to build linear, nonlinear classifiers using SVM model.
- Understand theory of unsupervised learning.

Course Outcomes :

- To obtain sound knowledge in the theory, principles and applications of pattern recognition
- Apply knowledge of mathematics, science, and engineering in the design and development of recognition system.
- Perform experiments on different software packages either obtain from external parties or developed by themselves and analyse the experimental results.
- Design and develop software for character recognition, image recognition given their specifications and within performance and cost constraints.
- Ability to understand the computing needs of inter-disciplinary scientific and engineering disciplines and design and develop algorithms and techniques for achieving these.

CSL522: Advances in Compiler Construction (DC) (L-T-P-C: 3-0-2 -4)

Pre-requisite Course : Desirable **-** Basic Compilers course in B.Tech.

Syllabus :

- Review of compiler fundamentals – lexical analysis, parsing, semantic analysis and intermediate code generation, error recovery, run time storage management, code generation.
- Code optimization – Peephole optimization, control flow analysis, data flow analysis, dependence analysis, redundancy elimination, loop optimization, procedural and inter procedural optimization, instruction scheduling.

- Compiling for High performance architectures, Compiling for scalar pipeline, compiling for vector pipeline, super scaler and VLIW processors, compiling for multiple issue processors, compiling for memory hierarchy. Parallelization and Vectorization, Dependence and dependence testing.
- Loop Normalization, Induction variable Exposure, Enhancing Fine Grained Parallelism, Loop Interchange, Scalar Expansion, Scalar and Array Renaming, Node splitting, Index-set splitting, Loop skewing

Reference Books

- Optimizing Compiler for Modern Architecture: A dependence based approach , Randy Allen, Kennedy
- Advanced Compiler Design and implementation : Steven S. Muchnick
- Engineering & Compiler : Keith D. Cooper & Linda Torczon: Morgan Kaufmann

Course Objectives :

Given basic knowledge of compiler techniques, students should be able to apply compiler optimization techniques so as to design an optimizing compiler.

Course Outcomes :

- Appreciation of parsing and code generation techniques
- Understanding of optimizations problems and issues, data flow analysis framework and mathematical modeling
- Appreciation of role of machine specific issues in compiler construction, the choice of instructions, the availability of registers etc.
- Ability to combine different optimization techniques to achieve the overall objective of program efficiency
- Appreciation of optimization techniques for multi-processor machines and parallelizing optimization schemes

Course Code & Title: - CSL523 : Advanced Computer Architecture (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite:

Syllabus :

- Classes of computers, Trends in technology, power and costs, dependability, quantitative principles of computer design, Introduction to computing models.
- Principles of scalable performance, performance metrics and measures, speedup performance laws, advanced processor technology, superscalar and VLIW processors, Verified memory, cache memory organizations, shared memory organizations. Memory hierarchy, cache performance, protection and examples of virtual memory, cache coherence.
- Pipeline and superscalar techniques, linear pipeline processors, reservation and latency analysis, collision free scheduling, pipeline schedule optimization, instruction pipeline design, arithmetic pipeline design, super scalar and super pipeline design.
- Multiprocessors and multi-computers, Brief overview of SIMD, MIMD, vector architectures and multi-core architectures.
- Elementary theory about dependence analysis, techniques for extraction of parallelism, branch prediction, dynamic scheduling, multiple issue and speculation, limits on instruction level parallelism, Thread level parallelism

Text / References :

- Computer Architecture : A Quantitative Approach : Hennessy and Patterson : Morgan Kaufmann: 4th
- Advanced Computer Architecture, Kai Hwang , McGraw Hill
- Advanced Computer Architectures : A design space approach, Sima D, Fountain T. and Kacsuk P, Pearson Education

Course Objective :

Expose the ideas and techniques that define the art of computer architecture, organization, and design. Provide the students with architectural framework and foundation needed to understand future trends in the design.

Course Outcome :

- Make the students aware about various trends in computer design, architecture of advanced processors.
- Realization about issues related to instruction level, thread level, data level parallelism in multi/many core systems, memory organization & optimization techniques.

- Understand the design issues with shared/distributed memory systems, multi / many core / GPGPU architecture

Course Code & Title: - CSL530 : Topics in Bioinformatics (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite: CSL 313 – Analysis of Algorithms

Syllabus :

- Basics of biology (1 Week)
- Sequences: Problem statement, Edit distance and substitution matrices, Global and local alignments, Spliced alignment, Space-efficient sequence alignment, Multiple alignment, (2 Weeks)
- Structures: Protein alignment, Protein structure prediction (1 Week)
- Phylogenetic trees: Large parsimony and small parsimony problems, Probabilistic approaches, Grammar-based approaches (1 Weeks)
- Overview of Gene Control, Working of Genetic Switches, Introductory Systems Biology, The biochemical paradigm, genetic paradigm and the systems paradigm (2 Weeks)
- Building an Organism Starting From a Single Cell -Quorum Sensing – Programmed Population Control by Cell-Cell Communication and Regulated Killing; Gene regulation at a single cell level- Transcription Networks -basic concepts -coherent Feed Forward Loop (FFL) and delay gate -The incoherent FFL -Temporal order, Signaling networks and neuron circuits -Aspects of multi-stability in gene networks. (3 Weeks)
- Modeling biological systems, Hidden Markov models (2 Weeks)
- Miscellaneous topics: Pathways and networks, Microarrays, Biomedical images, Genetic Algorithms and applications, (2 weeks)

Text books and/or other required material:

- "An Introduction to Bioinformatics Algorithms" by Jones, Pevzner. MIT Press.
- "Algorithms on Strings, Trees and Sequences" by Gusfield. Cambridge University Press.
- “An Introduction to Systems Biology: Design Principles of Biological Circuits” by Alon. Chapman & Hall/CRC Press.

Course Objective

Given the knowledge of string-based and statistical algorithms in Bioinformatics, a second semester M. Tech. CSE student will be able to model biological problems as theoretical computational formulations and reapply the solutions back to the biological world.

Course Outcomes

- This course introduces students to the basic string based computational methods and algorithms that can be used to understand the cell and biological systems.
- Students will know algorithms and programming techniques like dynamic programming, hashing, and suffix trees.
- The course focuses on computational approaches to: genetic and physical mapping; genome sequencing, assembly, and annotation.
- This course will help students develop multidisciplinary approach to the systematic analysis and modeling of complex biological phenomena.
- Students are also made aware of the currently emerging research areas in the fields of computational and systems biology.

CSL539: Formal Methods in Program Design (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite : An undergraduate course in Theory of Computation is desirable.

Syllabus :

- Foundations of parallel programming, Nondeterminism, Absence of control flow, Synchrony and Asynchrony, States and assignments, extricating proofs from program text, separation of concerns : correctness and complexity, programs and implementation.
- UNITY program structure, Assignemnet statement, Assign section, Initially-section, Always-section, Proving assertions about Assignment statement, Quantified assertions, conventions about priorities of logical relations.
- Fundamental concepts, proofs and theorems about: Unless / Ensures / Leads-to / Fixed-point. Proving bounds on progress.
- Introduction about Architecture and Mappings. All pairs shortest path problem: solution strategy, formal description, proof of correctness, creating the program. Implementation on sequential architectures, parallel synchronous architectures, asynchronous shared-memory architecture, and distributed architecture. Complexities on each of the architectures.
- Formal description and programs for saddle-point-of-a-matrix, reachability in directed graphs, prime number generation, comparing two ascending sequences, computing the maximum of a set of numbers, Boolean matrix multiplication.
- Program structuring, program composition by Union, Union theorem, composing specifications, substitution axiom, hierarchical program structures, superposition and superposition theorem, design specifications.
- Introduction to communicating processes.

Text / References

- Parallel Program Design, A foundation: K. Mani Chandy, Jaidev Misra, Addison-Wesley Publishing.
- The Science of Programming: David Gries, Springer.

Reference Books:

- Logic for Computer Science: Foundations of Automatic Theorem Proving : Jean Gaullier, Harper & Row Computer Science Technology Series.

Course Objectives:

Given basic knowledge of programming, students should be able to apply formal concepts used in program design, so as to argue about the correctness of the program.

Course Outcomes :

- Formal design of programs using concepts like determinism / non-determinism, synchrony/asynchrony.
- Appreciation of separation of concerns like correctness and complexity of programs
- Demonstrate ability to understand of optimizations concept of progress of a program and proofs thereof, architecture and program mappings, program structuring and program composition.
- Work individually to develop programs and proofs for the programs that demonstrate the learning of the theoretical concepts.

ECL445: Digital Signal Processing (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite:

Syllabus :

- Discrete time signals; Sequences; representation of signals on orthogonal basis; Sampling and reconstruction of signals.
- Discrete time systems; attributes, Z- Transform, Analysis of LSI systems, frequency analysis, Inverse Systems.
- Signal flow graph representation, DF1, DF2, parallel and cascade form. Finite word-length effects in Digital Filters
- Discrete Fourier Transform (DFT), Fast Fourier Transform algorithms.
- Design of FIR Digital Filters: Window method, Park-McClellan's Method.
- Design of IIR Digital Filters: Butterworth, Chebyschev approximations. Lowpass, Bandpass Bandstop and Highpass filters. Bilinear, impulse invariant frequency transformations.

- **Text/ References :**
  - Oppenheim & Schafer, "Discrete Time Signal Processing", PHI Ltd
  - Proakis John and Manolakis D.G, "Digital Signal Processing: Principles Algorithms and Applications", Prentice Hall 1992.
  - Cavicchi Thomas J, "Digital Signal Processing", Wiley 2002
  - Mitra S.K , "Digital Signal Processing A Computer -Based Approach", Tata McGraw- Hill

MAL 407 : Statistics & Optimization Techniques (DE) (L-T-PC- :3–0–0–3)

Pre-requisite:

Syllabus :

Statistics

- Sampling Theory : Population Parameter, Sample Statistics, Sampling distributions, Sample mean, Sampling distribution of means, the sample variance, the sampling distribution of variance.
- Estimation Theory: Point estimate and interval estimates, reliability, confidence interval estimates of population parameters, confidence intervals for means, proportions and variance.
- Tests of Hypothesis and Significance: Statistical decisions, tests of hypotheses and significance, Type I and Type II errors, level of significance, one tailed and two tailed tests. Tests involving small samples and large samples, fitting theoretical distributions to sample frequency distribution, The chi-square test for goodness of fit.

O. R. Techniques

- Linear Programming: Formulation of linear programming problem, Graphical solution- simplex method (including Big M method and two phase method), dual problem- duality theory, dual simplex method, revised simplex method.
- Transportation problem: existence of solution-degeneracy- MODI method; Assignment problem- traveling salesman problem
- Nonlinear programming problem (NLPP): Constrained NLPP,
- Lagrange's multipliers method – convex NLPP, Kuhn-Tucker conditions.

Text Books:

- Probability and Statistics, Author, M.R. Speigel , Publisher, McGraw Hill
- Operation Research, Author, H.A. Taha, Publisher, Prentice Hall of India Pvt. Ltd.

Reference Books:

- Introduction to Optimisation : Operations Research, Author, J.C. Pant, Publisher, Jain Brothers, New Delhi
- Probability and Statistics for Engineers, Author, Miller and Freund, Publisher,

Course Objectives :

- Modeling of various real life problems of operation research
- Learning of various methods in operation research
- Learning of new techniques in operation research

- Learning of applications of statistics in various real life problems
- Learning of different methods of statistics for data analysis

MAL 403 – Numerical Linear Algebra (DE) (L-T-P-C: 3-0-0-3)

Objective:

The objective of this subject is to expose student to understand the basic importance of Linear Algebra and numerical linear algebra and its applications to science and engineering.

Special Matrices, Vector and Matrix Norms, SVD. Floating Point Numbers and Errors. Stability, Conditioning and Accuracy. Gauss Elimination and Linear Systems, LU Factorization using Gaussian Elimination, Stability of Gaussian Elimination, Basic Results on Existence and Uniqueness, Some Applications Giving Rise to Linear Systems of Problems, LU Factorization Methods, Conditioning and Pivoting, Inverses and Determinants. Iterative Methods for Large and Sparse Problems: Gauss Seidal, SOR, Chebyshev Acceleration, Conjugate Gradient Method, Preconditioning. QR Factorization, SVD, and Least Squares Solutions. Numerical Eigenvalue Problems, Generalized Eigenvalue Problem.

Text Books:

- G. H. Golub and C. F. van Loan: Matrix Computations, Johns Hopkins University Press, 1984.
- L. N. Trefethen and D. Bau, III: Numerical Linear Algebra, SIAM, 1997.
- G. Allaire and S. M. Kaber: Numerical Linear Algebra, Springer, 2007.
- B. N. Datta: Numerical Linear Algebra and Applications, Springer, 2008.

CSL407 : Data Mining & Data Warehousing (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite: CSL403 : Database Management Systems

Syllabus :

- Introduction to Data warehousing **-** Application of Data warehousing and mining, Datawarehouse development life cycle, Data warehouse analysis, CUBE, ROLL UP and STAR queries.
- Data Warehouse Design **-** Massive denormalisation, STAR schema design ,Data ware house Architecture, OLAP, ROLAP and MOLAP , concepts of Fact and dimension table
- Space Management in Data warehouse - Schemas for storing data in warehouse using different storage structures, B-tree index, hash index, clusters, Bitmap index functional index, domain index, Data partitions.
- Performance and Tuning - Query optimization, memory management, process management. I/o management for Data warehouse.
- Data Mining Tools –Association rules, a priori algorithm, Fp-trees algorithm, constraints and solution.
- Cluster analysis- paradigms, DBSCAN, cluster algorithms.
- Mining tools- decision trees and applications.

Text / References :

- Jiawei Han, Micheline Kamber, “Data mining- Concepts & Techniques”, Morgan Kaufmann
- Michale Corey, Michale Abbey; Oracle 8i Data Warehousing; Tata McGraw Hill.
- Navathe and Elmasry ; Fundamentals of Database Systems; Addison Wesley, 2000
- Arun Pujari; Data Mining; Orient Longman, 2003

Course Objectives :

Necessity of separate data warehouse, design of warehouse, introduction of various OLAP queries, introduction to various data mining tools, application of tools to various problems.

Course Outcomes :

- Identify the scope and necessity of Data Mining & Warehousing for the society.
- Describe the designing of Data Warehousing so that it can be able to solve the root problems.
- To understand various tools of Data Mining and their techniques to solve the real time problems.
- To develop ability to design various algorithms based on data mining tools.
- To develop further interest in research and design of new Data Mining techniques.

CSL409 : Introduction to Distributed Systems (DE) (L-T-P-C: 3-0-2-4)

Pre-requisite Course : Operating Systems

Syllabus :

- Motivation and goals, broad overview and advantages of distributed systems main characteristics absence of global clock and state and possibility of large network delays
- Issues in distributed systems such as transparency, scalability, security, resource management etc. theoretical foundation - Lamport's clocks -Chandy-Lamport Global State recording algorithm - termination detection.
- Distributed mutual exclusion - Lamport, Ricart - Agrawal non-token based algorithm - token based algorithms - comparative performance analysis.
- Distributed deadlock detection issues - central and distributed detection algorithm - agreement protocols - model of processor failures - Byzantine agreement and other problems - solutions and applications.
- Distributed file systems - design issues - case studies with emphasis on NFS - distributed shared memory - coherence and coherence protocols - design issues and case studies.
- Distributed scheduling - issues , load distributing algorithms - load sharing policies and case studies - task migration and issues
- Recovery : introduction and basic concepts - backward and forward error recovery,
- Checkpointing : synchronous and asynchronous - atomic actions and commit protocols - voting protocols - reliable communication - cryptography : private and public - implementation issues , RSA algorithm - authentication in distributed systems - Kerberos case study.

Text/References:

- Singhal and Shivratri, "Advanced concepts in Operating Systems", McGraw Hill
- Coulouris, "Distributed Systems", AWL Press. Pearson Education
- Tanenbaum, "Modern Operating Systems", PHI

Course Objectives :

- Appreciation of the fundamentals, advantages, and challenges in designing and implementing distributed systems.
- Appreciation of the differences in the handling of issues like mutual exclusion, deadlock detection, fault handling, etc. in a centralized system and a distributed system.
- Ability to write distributed programs using sockets, RPC/RMI, etc.
- Ability to make intelligent choices from among available algorithms and techniques for the design of distributed systems subject to specific design and performance constraints.
- Ability to communicate effectively about different design decisions made for distributed system implementations, and their experimental evaluation.

CSL410 : Topics in Graph Theory (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite Course : CSL 202 – Discrete Maths and Graph Theory

Syllabus :

- Introduction to Graphs, Paths, Cycles and Trails, Vertex Degrees and counting, Directed graphs, Trees and distances, Spanning trees, Shortest paths, Matchings and covers, Dominating Sets, Bipartite Matchings, Cuts and connectivity, K-connected and K-edge connected graphs, Network flow problems.
- Colorings of graphs, Structuring of K-chromatic graphs, chordal graphs.
- Planer graphs, Embeddings, Euler's formula, Parameters of planarity.
- Line graphs and edge coloring, Hamiltonian Cycles. Applications in Switching and coding theory. Euler's Cycles
- Electrical Network analysis and operations Research., Ramsey Theory

Text/References:

- Narsingh Deo, "Graph Theory with applications to Engineering and Computer Science", PHI
- Frank Harary, "Graph Theory", Narosa Publishers

Course Objectives :

Given the knowledge of Graph Theory, a seventh semester CSE student will be able to convert the every day problems to graph theory problems and apply proof techniques. He should be able to solve problems on different graph theoretical constructs.

Course Outcomes :

- This course introduces students to the basic computational methods and abstractions to graphs.
- Students will be able to explain and apply the basic methods of graph Theory. They will be able to use these methods in design and analysis of algorithms.
- Students should also become able to identify graph theory problems in a natural way even when they appear in a different setting. In the later part of the course the goal is to deepen students' knowledge of graph theory by showing interrelations of some seemingly loosely-related concepts and further develop problem solving skills.
- Students are also exposed to currently emerging research areas in the fields of social networking that uses graphs.

CSL411 : Software Project Management (DE) (L-T-P-C: 3-0-0-3)
Pre-requisite Course : CSL308: Software Engineering

Syllabus :

- Overview of Software Project Management, The Project Life Cycle, Software Development Life Cycle Models, Life Cycles and Metrics, Process Maturity: SEI CMM
- Estimation Techniques of IT, Project Scoping, Project Planning, Project Control, Project Phase-Out, Risk Management, Configuration Management
- People Management, Team Dynamics, Net Present Value, Project Portfolio Management, Software Quality Assurance, Project Leadership

Text Books:

- R.K. Wysocki et al. : Effective Project Management: Traditional, Agile, Extreme, 5th Edition, Wiley India, 2011.
- C. Jones : Applied Software Measurement, Assuring Productivity and Quality, McGraw Hill

References:

- 1. D. I. Cleland : Project Management, Strategic Design and Implementation, 3rd edition, McGraw-Hill.

Course Objectives :

- The student will understand the requirements for the content of a project management plan.
- The student will be able to write a plan for a small project according to an established standard.
- The student will understand the role of the manager in each phase of the software development life cycle.
- The student will appreciate the key roles man- agers play in software development efforts.
- The student will appreciate economic and customer-driven factors and their role in the eventual form of the software product

Course Outcomes :

- Match orgnizational needs to the most effective software development model
- Understanding the basic concepts and issues of software project management
- Effectively planning the Software projects and employ mechanisms for tracking the software projects
- Implementing project plans through managing people, commnication and change
- Developing skills for tracking and controlling software deliverables and address real-world management challenges

CSL 430 Business Intelligence (DE) (L-T-P-C :3–0–0– 3)

Pre-requisites :

Syllabus :

- **Introduction to Business Intelligence**

  What is Business Intelligence, Why do we need Business Intelligence, EIS, MIS, DSS & BI , Information Pyramid – Data, Information, Knowledge & Intelligence. Basis for Operational, Tactical & Strategic Decision Making, OLTP Vs. OLAP, Requirements Gathering in BI through Business Questions, BI in various Domains and Functional Area

- **Principles of Dimensional Modelling**

  Foundation for Fact based decision making, The STAR and SNOWFLAKE schema, Pros & Cons of the STAR/SNOWFLAKE Schema Dimensional Model, Slowly Changing Dimension tables, Fact-less Fact Tables, Aggregation Strategy, Time Dimension

- **Business Intelligence System Architecture**

  Need for Enterprise Class Business Intelligence Infrastructure, The BI Ecosystem, Building Blocks of a N-Tier BI System – Servers & Communication Protocols, The Central Repository – Metadata, Information Consumption User Interfaces – Desktop Vs. Web Vs. Mobile, Open Architecture, Scalability, Performance in BI – In Memory Analytics

- **BI Project Lifecycle**

  Typical BI Project Lifecycle, Requirements Gathering & Analysis – Functional & Non-Functional Requirements, Reports & Dashboards Design – Mock-up and Storyboarding, Testing in a BI Project, BI Project Deployment, Post Production Support
  Introduction to Enterprise Class BI Tool
  First Level of Abstraction of the Data Warehouse in MicroStrategy, Building the Schema Objects – Attributes, Facts, Transformation & Hierarchies, Building Reusable Application Objects – Metrics, Filters, Prompts, Five Styles of BI, Building Reports – Grids & Graphs, Report Manipulation over the Web – Pivoting, Sorting, Drilling, Exporting etc., Setting up Report Distribution, Report Project

Text Book

- Turban E., Sharda R., Delen D., King D., Business Intelligence, Pearson Education.

Reference Book

- Sabherwal R. and Becerra-Fernandez I., Business Intelligence, Wiley.
- Kimball R., Ross M., The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence, Wiley and Sons (2010).

Course Objectives :

- Understand concepts in Business Intelligence
- Get acquainted with the tools used in Business Intelligence
- Understand modeling and implementation of Business Intelligence

Course Outcomes :

- To obtain sound knowledge in the theory, principles and applications of Business Intelligent System.
- Design and develop multidimensional data model given their specifications and within performance and cost constraints.
- Acquire and understand mathematical concepts to develop data centric decision models.
- Perform experiments in different disciplines of Business Intelligent System.

CSL 431 - Introduction to Cloud Computing (DE)(L-T-P-C:3–0–0-3)

Pre-requisites :

Syllabus :

- Introduction and Motivation for cloud computing, Cloud Computing principles.
- Cloud system architectures, Delivery models - infrastructure-as-a-service, platform-as-a-service and software-as-a-service, Types of Clouds – public, private and hybrid clouds.
- Virtualization, Infrastructure and Data storage Management, Architecture and design of storage and compute clouds.
- Authentication, Authorization and Accounting, Cloud Security, privacy, policy and compliance.
- Cloud reliability, disaster recovery and fault-tolerance.
- Cloud Economics - Metering, Monitoring and Pricing, Viability of Cloud.
- Cloud programming frameworks, cloud interfaces, Interoperability and standards.
- Case studies such as Amazon Web Services, Windows Azure and Google AppEngine.

Text Book

- Hwang K., Dongarra J., Fox G.C., Distributed and Cloud Computing, Morgan-Kaufman.

Reference Book

- Buyya R., Broberg J., Goscinski A. M., Cloud Computing – Principles and Paradigms, Wiley.

Course Objectives :

Given the knowledge of languages and grammars with tools to process them, a seventh semester B. Tech. CSE student will be able to understand current cloud computing technologies, including technologies for Infrastructure as a Service, Platform as a Service, and Software as a Service.

Course Outcomes :

- Understanding of different layers of cloud computing, infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Introduction of practical IaaS, PasS, SaaS, including Amazon ECS, GAE, Force.com, Microsoft Azure, etc.
- Gain knowledge of cloud storage system design issues, including directory management, data placement, and consistency issues. Practical cloud storage system solutions including GFS, Big Table, HDFS, etc.
- Student should have good knowledge of authentication, authorization and secure access in the cloud. Introduction to cloud security. Secure computation in the cloud.
- Insights into the virtualization technologies: Hypervisor, emulation, and application VM. Platform virtualization, storage virtualization, and network virtualization.

CSL 437: Enterprise Resource Planning (ERP) (DE) (L-T-P=C:3-0-0-3)

Prerequisites : None

Syllabus :

- Definition / concept of an enterprise, Enterprise Resource Planning (ERP), the main misconceptions about ERP, Evolution of ERP, reasons for explosive growth of ERP in the market, tangible and intangible benefits of ERP systems, Limitations of ERP, Concept of business integration and how it is achieved by ERP systems, discussion on whether companies can develop their own ERP packages or should go for ERP implementation.

- Why should software engineers learn ERP systems. Different perspectives on ERP Systems – Business Perspective, Operations Perspective, Technology perspective.

- ERP and related technologies, Definition and example of Management Information Systems, Definition , concept and example of Decision Support Systems, Definition and example of Executive Information Systems, Introduction to Data Warehousing, Introduction to Data Mining, Concept of Online Analytical Processing, concept of Supply chain management, concept of Supply Relationship Management, concept of Customer Relationship Management, concept of Product Lifecycle Management.

- ERP implementation process, gap analysis, importance of end user training, concept of business consultants, definition of vendors, concept of domain experts, definition of end users, Concept of customization, ERP Implementation guidelines, reasons why an ERP fails, change management (Not s/w change management but other changes for e.g. Part change approval cycle in say a manufacturing process)

- Electronic commerce, role of ERP in electronic commerce, names of major ERP players in the market and brief about them Introduction to various ERP Modules. There are many modules, instructor will choose a few as per what the time permits

- ERP Case studies there are many and the instructor will select a few, Future of ERP. Impact of Business Intelligence and Cloud Computing on ERP systems.

Text Books

- Enterprise Resource Planning: Mary Sumner, Pearson Fifth Edition, 2009
- Enterprise Resource Planning: Alexis Leon Tata McGraw-Hill
- Reference BooksConcepts in Enterprise Resource Planning: Bret Wagner, Ellen Monk, Ceneage Learning India, 2012
- Enterprise Resource Planning Systems: Daniel E. O'Leary, Cambridge University Press

Course Objectives :

- Understand necessity of ERP in organization.
- Learn modeling and implementation details of ERP
- Learn customization issues in ERP implementations.

Course Outcomes :

CSL439 : Human Computer Interface (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite: CSL214 : Data Structures and Program Design – II , CSL303:Introduction to Object Oriented Methodology

Syllabus :

- Introduction to Human-Computer Interaction, Conducting User Studies, Managing design processes, Evaluating User Interfaces, Design guidelines, principles and theory
- The Media Equation, Design of Everyday Things, Direct Manipulation and Virtual Environments, Interaction Devices, Command and Natural Languages
- Collaboration and social media participation, Design issues –Quality of Service, Balancing function and fashion, Information search and visualization.

Text Book:
- Ben Shneiderman, Maxine Cohen, Steven M. Jacobs, Catherine Plaisant: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5th Edition, Pearson, 2010

References:
- Donald Norman: Design of Everyday Things, Basic Books (2002).

CSL521 : Software Architecture (SA) (DE) (L-T-P-C: 3-0-0-3)

Pre-requisite: CSL308 Software Engineering or equivalent

Syllabus :

- Software process and the role of modeling and analysis, software architecture,and software design (1 week)
- Software Modeling and Analysis: Analysis modeling and best practices, traditional best practice diagrams such as DFDs and ERDs (2 weeks)
- Software Architecture: architectural styles, architectural patterns, analysis of architectures, formal descriptions of software architectures , Architectural description languages and tools (3 weeks)
- Software Design: design best practices, design patterns, design case studies, component technology, object oriented frameworks, distributed objects, interoperability standards, case studies., software quality (3 weeks)
- UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns, documenting software architecture, reconstructing software architecture. (2 weeks)
- Middleware components, programming models, implementation, systems qualities Moving from qualities to architecture and views Components and COTS, Economics- Driven Architecture, Software product line, Software architecture future.(2 weeks)
- Issues in Software Architecture: Scalability and interoperability issues, web application architectures, case studies. (2 weeks)

Text/Reference Material :

- M. Shaw, "Software Architecture Perspectives on an Emerging Discipline", PHI
- Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Pearson Education Asia
- R. Taylor, N. Medvidovic, E. Dashofy, "Software Architecture – Foundations, Theory, and Practice", Wiley India
- Jan Bosch, "Design and Use of Software Architectures", Addision-Wesley-Pearson Education
- Christine Hofmeister, Robert Nord, Dilip Soni, "Aoolied Software Architecture", Addision-Wesley-Pearson Education
- Dikel, D.Met Al, "Software Architecture: Organizational Principles and Pattern", Prentice Hall

Course Objective :

The students should know how to develop different architectural views of an architecture, addressing specific concerns of stakeholders. It involves making a large number of trade-offs between concerns of different stakeholders. There may be different acceptable solutions, and the solution eventually chosen depends on how the balancing between stakeholder concerns is made. The students should know how to do an assessment of an architecture.

Course Outcomes:

On completion of this course students will be able to:

- Design and understand software architecture for large scale software systems.
- Recognize major software architectural styles, design patterns, and frameworks
- Describe a software architecture using various documentation approaches and architectural description languages
- Develop architectural alternatives for a problem and select among them
- Use well-understood paradigms for designing new systems

CSL447 : Introduction to Parallel Computing (DE) (L-T-P-C : 3-0-2-4)

Pre-requisite – none

Syllabus :

- Introduction to Parallel Computing, Scope of Parallel Computing, Dependence Analysis: Single loop, Double loop, Perfect Loop Nest, Method of Bounds, Method of Elimination. (1)
- Loop Tiling for Parallelism: Mathematical Background, Non-Singular Transformations and Permutability, Rectangular Tiling, Parallelepiped Tiling. (2)
- Parallel Programming Platforms, Principles of Parallel Algorithm Design, Basic Communication Operations, Analytical Modeling of Parallel Programs, Programming Using the Message-Passing Paradigm, Programming Shared Address Space Platforms, Applications:

Dense Matrix Algorithms, Sorting, Graph Algorithms, Search Algorithms for Discrete Optimization Problems, Dynamic Programming, Fast Fourier Transform (3)

- Overview of OpenMP, Writing a First OpenMP Program, OpenMP Language Features, Get Good Performance by Using OpenMP, Using OpenMP in the Real World (4)

Text/Reference Material :

- Dependence Analysis by Utpal Banerjee, Kluwer Academic Publishers, Boston / Dordrecht / London
- Loop Tiling for Parallelism by Jingling Xue, Springer Science+Business Media, LLC
- Introduction to Parallel Computing, Second Edition by Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Publisher: Addison Wesley January 16, 2003, ISBN: 0-201-64865-2
- Using OpenMP by Barbara Chapman, Gabriele Jost, Ruud Van Der Pas

Course Objective :

Students will be able to:

- 1. Understand the scope, design and model of parallelism
- 2. Know the parallel computing architecture.
- 3. Know Tiling transformation for Loop Parallelization.
- 3. Know the Characteristics, Model and design of parallel algorithms.
- 4. knows programming models and methods for basic computer science problems.
- 5. Analytical modeling and performance of parallel programs.
- 6. Solve a complex problem with message passing model and programming with MPI.
- 7. Analyze complex problems with shared memory programming with openMP.

Course Outcomes:

- Well verged with fundamental concepts of parallelism.
- Develop message-passing parallel programs with OpenMP
- Develop message-passing parallel programs with MPI
- Able to Apply shared memory parallel program concepts with Java threads and OpenMP.
- Can Illustrate multi-threaded and message passing parallel algorithms.
- Able to Design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.
- Develop ability to Reconstruction of emerging parallel algorithms with MPI.
- Able to Compare and contrast various parallel algorithms using shared memory and MPI.
- Develop skill to compute contemporary parallel algorithms.
- Develop skill analyzing parallel computing problems
- —solving problems posed in class
- 11. Develop skill writing parallel programs
- —programming assignments
-

PHY4xx: Quantum Computation and Quantum Information (DE) (L-T-P-C:3-0-0-3)

Pre-requisite – PHL101: Physics

Syllabus :

I. Introduction to quantum mechanics

- Hilbert space

- Unitary and stochastic dynamics
- Classical and quantum information
- Density operators and correlations

II. Quantum computation
- Classical computation
- Quantum Circuits, qubit operations
- Quantum Entanglement
- Shor factorization
- Quantum Search Algorithms

III. Noise and error correction
- Quantum Noise and Quantum operations
- Graph states and codes
- Quantum error correction
- Fault-tolerant computation

IV. Quantum Information and Applications
- Entropy and Information
- Data Compression
- Quantum cryptography

Text Books

1. M. A. Nielsen and I. L. Chuang , Quantum Computation and Quantum Information, Cambridge University Press.

Reference Books

R. B. Griffiths: Consistent Quantum Theory (Cambridge 2002) http://quantum.phys.cmu.edu/CQT/index.html

Image and Video processing (DE) (L-T-P-C: 3–0–2-4)

Pre-requisites: Discrete Mathematics, Design and Analysis of Algorithms

Syllabus:

- Introduction and applications of image and video processing. Fundamentals of Digital Image and Video; acquisition, sampling and quantization. Representation of digital images as matrices, types of images (binary, grayscale, color, indexed).
- Basic topology: neighbors and neighborhoods, adjacency, connectivity, Regions and boundaries.
- Image enhancement: Intensity transformations and Spatial filtering, Histogram equalization, Averaging, Smoothing, Laplacian filters etc.
- Image restoration: Types of Noise and removal methods - Mean filter, Median, Min, Max, Midpoint, Adaptive filters etc.
- Basics of filtering in the frequency domain: The Fourier transform - basic definitions and examples. Steps for filtering in the frequency domain.
- Binary mathematical morphology: Erosion and Dilation, Opening and Closing, The Hit or Miss Transformations, extraction of connected components.
- Image and Video Segmentation - Edge Detection and enhancement, Thresholding, Edge based and Region based Segmentation techniques.
- Object detection in videos: Basics of background modeling and foreground detection, connected component labeling,
- Case study of applications like automated video surveillance.

Text Book

- R.C. Gonzalez and R.E. Woods , "Digital Image Processing", Pearson Education, 3rd ed.
- M. Sonka, V. Hlavac, and R. Boyle. ,"Image Processing, Analysis, and Machine Vision", Cengage Learning, 2009.

References

- "Computer Vision: Algorithms and Applications" by Richard Szeliski, Springer, 2010.
- "Fundamentals of Digital image processing", by Anil K. Jain, PHI, 2010
- Journal/Conference papers and Reference manuals of Image/Video processing tools.

Course Objectives:

Beyond the obvious applications in entertainment and scientific visualization, digital images and video have become a central component of net-centered computing, human/computer interfaces, and databases, as well as data analysis for domains such as biometrics, surveillance and remote sensing. This course offers fundamentals of digital image and video processing and algorithms for most of the work currently underway in this field. Through this course, students will get a clear impression of the breadth and practical scope of digital image and video processing and develop conceptual understanding which will enable them to undertake further study, research and/or implementation work in this area.

Course Outcomes:

At the end of the course the students will be able to:

- Describe the fundamentals of image and video processing and their applications
- Develop familiarity and implement basic image and video processing algorithms.
- Select and apply appropriate technique to real problems in image and video analysis.

CSL519 : Distributed Systems

Syllabus :

The study of basic techniques in the design and development of Distributed Systems and understanding solutions of the fundamental problems in distributed systems like mutual exclusion, deadlock detection, termination detection, leader election, fault tolerance, etc. 3 lectures per week. Credit scheme - (L-T-P-C: 3-0-2-8)

Required Background or Pre-requisite: Operating Systems Course

Detailed Description of the Course

- Introduction and motivation to Distributed Systems,Characteristics, Applications, Challenges, Architecture types, Fundamental models. (2 weeks)
- Inter-process and inter-node communication using Sockets – connection oriented and connection-less, Remote Procedure Calls, Remote Method Invocation (1 week)
- Clock Synchronization Techniques, Network Time Protocol, Logical Clocks, Vector Clocks. (1 week)
- Distributed File System Design and Implementation, Case Studies of NFS, Andrew File Systems, HDFS, Distributed Resource Management. (1 week)
- Causally Ordered Broadcast and Unicast, Termination Detection – Ring based and Dijkstra Scholten algorithms, Leader Election – Ring based, Franklin's algorithm and Bully Algorithm (2 weeks)
- Distributed Mutual Exclusion – Token based algorithms – Lamport's, Ricart-Agarwala, Maekawa's algorithms, Non Token based Algorithms – Suzuki Kasami, Raymond's algorithms, comparison of different algorithms. (2 weeks)
- Distributed Deadlock Detection, Resource and Communication Deadlocks – Centralized technique, Distributed technique - edge chasing and path pushing algorithms, Hierarchical technique, Recovery from Deadlocks. (1 week)
- Fault Tolerance, Handling Crash faults –Two phase commit protocol, Non-blocking three phase commit protocol, Birman-Joseph Atomic Broadcast Protocol, Voting techniques for fault tolerance. (2 weeks)

- Recovery – forward and backward recovery, undo-redo logs, Coordinated and Uncoordinated Checkpointing and Recovery algos (1 week)
- Agreement protocols – LSP Oral Messages, Agreement using Signed Messages (1 week)

Typical Laboratory Experiments

Socket Programming, RPC, Using Sockets to implement a rudimentary Distributed File System, Implementation and Performance Evaluation of any Distributed Mutual Exclucion algorithm.

Text/References

- Singhal and Shivratri, "Advanced concepts in Operating Systems", McGraw Hill
- Coulouris, "Distributed Systems", AWL Press. Pearson Education
- Tanenbaum, "Modern Operating Systems", PHI

Course Objective

Given the knowledge of operating systems and sequential program design, the students of the second semester M. Tech. CSE will be able to design and develop fault tolerant and efficient distributed algorithms to solve large problems where data and control is distributed over different nodes.

Course Outcomes

1. Identify the advantages and challenges in designing distributed algorithms for different primitives like mutual exclusion, deadlock detection, agreement, etc.
2. Design and develop distributed programs using sockets and RPC/RMI.
3. Differentiate between different types of faults and fault handling techniques in order to implement fault tolerant systems.
4. Analyze different algorithms and techniques for the design and development of distributed systems subject to specific design and performance constraints.

CSP438 : Software Lab – III (7th Sem) **0-0-2-2**

*Main theme - Use of open source tool*s

1. Advanced use of Apache Web server: Installing and using Apache Web server in load sharing manner (Configuration of 2 or more server hosting a common website
2. Advanced use of MySQL server:
    a. Installing and using mysql servers in load sharing manner (Configuration of 2 MySQL instances in master-slave mode).
    b. Database operations via programs written in C/C++ or Java.
3. Java Native Interface (JNI) – Calling C / C++ code from Java and vice versa.
4. Automatic testing tools – Junit, NUnit,
5. Advanced use of open source cloud platforms:
    a. Integration of gmail with google calendar – from gmail you should be able to schedule an appointment with all the recipients of the mail.
    b. Creating a website on Salesforce cloud for tracking inventory from east, west, north, south regions in India separately.
    c. Accessing google-map via google-map APIs
6. Downloading and Installing Hadoop on 3 to 4 machines and writing a distributed sorting program on the same.

7. Creating web-services
    a. Using Axis-2 (Java) or gSoap library (C / C++)
    b. Introduction to SSL. Use digital certificates to encrypt / decrypt data in transfers
    Notes - Keytool in Java allows to create / store / manipulate certificates
    Also, refer www.thawte.com for free download/creation of a certificate
8. Introduction to Android Platform and APIs / libraries provided. A sample game / application on Android.
9. Learning software engineering tools
    a. Design tools - Rational Rose / visio
    b. Memory leaks - Purify
    c. Code Coverage tools
    d. Testing tools – Loadrunner, Winrunner