DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Scheme of Instructions and Syllabus
for Under Graduate & Post Graduate Studies

B.Tech & M.Tech.
in Computer Science & Engineering
2014 - 2015

Visvesvaraya National Institute of Technology, Nagpur

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

SN	Name	Qualification	E-Mail id
1.	O.G.KAKDE	Ph.D.	ogkakde@cse.vnit.ac.in
2.	S.R. SATHE	Ph.D.	srsathe@cse.vnit.ac.in
3.	P.S. DESHPANDE	Ph.D.	psdeshpande@cse.vnit.ac.in
4.	U.A. DESHPANDE	Ph.D.	uadeshpande@cse.vnit.ac.in
5.	A.S. MOKHADE	M.Tech.	asmokhade@cse.vnit.ac.in
6.	R.B. KESKAR	M.Tech.	rbkeskar@cse.vnit.ac.in
7.	A. TIWARI	ME	atiwari@cse.vnit.ac.in
8.	M.P. KURHEKAR	M.Tech.	mpkurhekar@cse.vnit.ac.in
9.	D.D. SHRIMANKAR	M.Tech.	dshrimankar@cse.vnit.ac.in
10.	M.M. DHABU	ME	meeradhabu@cse.vnit.ac.in
11.	S.A. RAUT	M.E.	saraut@cse.vnit.ac.in
12.	M.A. RADKE	MS	Mansi.radke@cse.vnit.ac.in

iii) Scheme of Examination / Instruction – B.Tech. Computer Science Engineering Credit Requirements (2013 admitted batch):

Overall Credit Structure

Category	Credits
Basic Sciences + Engineering Sciences (Credits from First Year)	78
CSE Departmental Core	152
CSE Departmental Electives	78-90
Humanities	0
OC	0-12
Total	**320**

Notes :

1. Given below is scheme for 3rd semester to 8th semester B.Tech, Computer Science Engg.
2. As per the scheme the course on "Technical Communication", offered by Humanities department in the 6th semester is compulsory. In view of this, credits for other Humanities courses in the scheme is 0 (zero).
3. Credits that can be earned out of OC courses during 3rd to 8th semester are 0-12. To earn these 0-12 credits during 3rd – 8th semester, a student can register for, at maximum, 2 OC courses.
4. A student may decide to register for lesser number any OC courses in that case he/she will cover remaining of the 12 OC credits from DE courses.
5. There are 2 slots provided for OC courses in the scheme below, one each in 7th and 8th semester. If a student wishes to register for OC course, he/she can do it *one OC course per semester* ***only in*** (*one or two of*) ***these two semesters***.
6. If a student does not register for an OC course in a particular semester allotted for OC (i.e. in 7th or 8th semester), he/she has to compensate for these 6 credits by registering for a DE course **in the same semester**. For example, if a student does not register for OC

course in 7th semester, he/she will have to register for a DE course in the same semester, so that total credits earned in any particular semester is maintained.

7. A student should register for at-least 60 credits, from DE/OC courses in 7th and 8th semester together. This requires a student to cover at-least 36 credits from DE/OC courses in the 7th semester. More details given below.

Details of Credits :

III Semester				IV Semester			
Code	Course	L-T-P	Cr	Code	Course	L-T-P	Cr
Core				**Core**			
MAL 205	Numerical Analysis and Probability Theory (NAPT)	3-0-0	6	CSL 222	Computer Organization	3-0-0	6
ECL 209	Digital Circuits and Logic Design	3-0-0	6	MAL 206	Linear Algebra and Applications	3-0-0	6
EEL 210	Electrical Sciences	3-0-0	6	CSL 214	Data Structures and Program Design II	3-0-2	8
CSL 202	Discrete Mathematics and Graph Theory	3-1-0	8	CSL 204	Concepts in Programming Languages	3-0-2	8
CSL 213	Data Structures and Program Design I	3-0-2	8	CSL 223	Microprocessor-based Systems	3-0-2	8
CSP 224	Software-Lab-I	0-0-2	2				
Elective				**Elective**			
CSL 224	Introduction to Web Programming	2-0-2	6	CSL 225	Advanced Web Programming	2-0-2	6
ECL 208	Analog Circuits	3-0-0	6	EEL 211	Control Systems	3-0-0	6
				ECL 215	Signals and Systems	3-0-0	6
Current Sem : DC:36; DE :6 (1 of 2)			**42**	**Current Sem : DC:36; DE:6 (1of 3)**			**42**
Cumulative : DC :36; DE 6;			**42**	**Cumulative: DC:72; DE:12**			**84**

Note for Even Semester registration:

Students of 2010 entry batch and earlier, who have not cleared, as yet,

a. Integral-Transforms and Partial Differential Equations" OR / AND
b. Physics-2

must register for "Linear Algebra" so as to meet their DC credit requirements.

V Semester				VI Semester			
Code	Course	L-T-P	Cr	Code	Course	L-T-P	Cr
Core				**Core**			
CSL 307	Theory of Computation	3-0-0	6	CSL 315	Database Management Systems	3-0-2	8
CSL 312	Operating Systems	3-0-0	6	CSL 316	Language Processors	3-0-2	8
CSL 313	Analysis of Algorithms	3-0-0	6	CSL 308	Software Engineering	3-0-0	6
CSL 303	Introduction to OO Methodology	3-0-2	8	CSL 317	Computer Networks	3-0-2	8
CSP 314	Software Lab II	0-0-2	2				
CSL 306	System Programming	3-0-2	8				
Elective				**Elective**			
CSL 304	Neurofuzzy Techniques	3-0-0	6	HUL 301	Technical Communication	3-0-0	6
CSL 305	Computer Graphics	3-0-0	6	CSL 318 / CSL 319	Business Information Systems (BIS)/ Internet Technologies (1 of 2)	3-0-0	6
Current Sem : DC:36; DE :6 (1 of 2)			**42**	**Current Sem : DC:30; DE:12 (2of 3)**			**42**
Cumulative : DC :108; DE 18;			**126**	**Cumulative: DC:138; DE:30**			**168**

Note- BIS and Internet Technology are grouped together and only one of this group can be registered by a student.

VII Semester				VIII Semester			
Code	Course	L-T-P	Cr	Code	Course	L-T-P	Cr
Core				**Core**			
CSD 401	Project Phase I		4	CSD 402	Project Phase II		8
CSP 438	Software Lab – III	0-0-2	2				
Elective				**Elective**			
CSL 408	Topics in Embedded Systems	3-0-0	6	CSL 406	Network Security	3-0-0	6
CSL 412	Artificial Intelligence	3-0-2	8	CSL 407	Data Mining and Data Warehousing	3-0-0	6
CSL522	Advances in Compilers	3-0-2	8	CSL 410	Topics in Graph Theory	3-0-0	6
CSL517	Pattern Recognition	3-0-2	8	CSL 409	Topics in Distributed Systems	3-0-0	6
CSL440	Fundamental Algorithms in Computational Biology	3-0-0	6	CSL 411	Software Project Management	3-0-0	6
CSL436	Information Retrieval	3-0-0	6	CSL 439	Human Computer Interface	3-0-0	6
CSL539	Formal Methods in Program Design	3-0-0	6	CSL 430	Business Intelligence	3-0-0	6
CSL523	Advanced Computer Architecture	3-0-0	6	CSL 431	Introduction to Cloud Computing	3-0-0	6
CSL441	Paradigms in Programming	3-0-0	6	CSL 437	Enterprise Resource Planning	3-0-0	6
ECL4xx	Digital Signal Processing	3-0-0	6	CSL 521	Software Architecture	3-0-0	6
				CSL 530	Top. In Bioinformatics	3-0-0	6
				MAL 407	Statistics and OR	3-0-0	6
				PHY4xx	Quantum Computation and Quantum Information	3-0-0	6
	OC (Open Course)	3-0-0	6		OC	3-0-0	6
Current Sem : DC:6; DE+OC : 36 -42 (5/6 out of 11)			**42-48**	**Current Sem : DC:8; DE+OC:18-24 (3/4 out of 12)**			**34-40**
Cumulative : DC :144; DE+OC:66-72			**210-216**	**Cumulative: DC:152; DE+OC:90**			**242**

Notes for 7th semester –

1. Minimum36 DE+OC credits must be earned in 7th semester.
2. Maximum DE credits that can be earned in 7th semester are 42 (by taking 3 courses of 8 credits each and 3 courses of 6 credits each).
3. **The courses for MTech (CSE), viz 'Advances in Compilers', 'Pattern Recognition', 'Advanced Computer Architecture' and 'Formal Methods in Program Design' are 5-level courses, those will be offered as electives for final year B.Tech. students. The pre-requisites are as follows :
    a. Advances in Compilers : pre-requisite - CSL316 Language Processors
    b. Pattern Recognition : pre-requisite –
        i. MAL206 : Linear Algebra and Applications
        ii. MAL205 : Numerical Analysis and Probability Theory
    c. Advanced Computer Architecture : pre-requisite – CSL222 : Computer Organization
    d. Formal Methods in Program Design : pre-requisite –
        i. CSL214-Data Structures and Program Design –II

In addition, the instructors for these courses may announce additional criteria for students wishing to register for these. The criteria may be based on grades scored in pre-requisite courses, overall CGPA or any other criteria and/or a combination of different criteria, that can be announced before registration.

Notes for 8th semester–

1. A student should cover a total of 60 DE+OC credits in 7th + 8th semester, together.
2. A student may opt for lesser number of courses in DE+OC category (3 instead of 4) in 8th semester, in case his requirement of DE credits gets fulfilled. For example, if a student has covered 42 DE credits in 7th semester, he/she has to register for DE/OC courses so as to cover minimum of 18 credits. On the other hand, a student covering 36 DE credits in 7th semester will have to cover 24 DE credits in 8th semester.
3. ** - These are M.Tech. level courses, that can be made available to B.Tech. students on the need basis. The prerequisite courses are as follows
    a. Software Architecture –
        i. CSL308 Software Engineering,
        ii. CSL303 Introduction to OO Methodology
    b. Topics in Bioinformatics –
        i. CSL202 – Discrete Mathematics and Graph Theory
        ii. CSL313 – Analysis of Algorithms

    The instructors for these courses may announce additional criteria for students wishing to register for these courses. The criteria may be based on grades scored in pre-requisite courses (as applicable), overall CGPA or any other criteria and/or a combination of different criteria that can be announced before registration.

MAL205: Numerical Meth. & Prob. Theory (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite: None

- **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.

ECL209 : Digital Circuits & Logic Design (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite: None

- 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

- Basics of PN Junction Diode, Diode applications in digital circuits, Basics of Transistor , CMOS characteristics, Standard Logic Families Diode Logic, TTL, CMOS Logic.
- 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 subtractors Decoders/Encoders, multiplexers/ Demultiplexers, code converters, realizing fucntions 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) .
- Overview of VLSI design process. PAL, PLAs, PROMS, CPLD, FPGA,ASIC structure overview. Hardware description language for digital circuit implementation(VHDL). Structural level implementation, Behavirol implementation and dataflow method.

Text/References:

- 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

EEL210 : Electrical Sciences (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite:None

- Nodal analysis, Mesh analysis, Source transformation, Duality.
- Theorems: Superposition theorem, Reciprocity theorem, Thevenin's theorem, Norton's theorem, Maximum power transfer theorem.
- Two port network: Two port network parameters, their inter-relation, Interconnection of 2-Two port networks.
- Measurement of low, medium and high resistances, Elementary methods of measurement of inductances & capacitances, Generalised theory of A.C.Bridges, their uses for measurement of inductance and capacitance.
- Measuring instruments: Classification, Absolute and secondary instruments.
- Electronic instruments: Digital voltmeters, Digital multimeter, Cathode ray Oscilloscope, Syncroscope, etc.
- Generalized principle of operation of Alternators, Armature reaction, Principle of operation of Synchronous motors, Starting methods, Stepper motor.

Text/References:

- Sawhney, A.K.; A Course in Electrical and Electronics Measurements and Instrumentation; (Eleventh Edition) Dhanpat Rai & Sons, Delhi.
- Van Valkenburg ; Network analysis (Third Edition); Prentice Hall of India Private Ltd, Delhi.
- Kelkar A.R., Pandit P.S.; Network Analysis ; Prathibha Publications ; Nagpur
- Theraja.B.L and Theraja A.K.; A Text book of Electrical Technology (Vol-II) ;
- S.Chand and Company ; New Delhi

CSL202 Discrete Maths & Graph Theory (DC) (L-T-P-C: 3-0-0-6)

Pre-requsite : None

- 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

- This subject offers students an introduction to Discrete Mathematics oriented toward Computer Science and Engineering. It covers:
- Fundamental concepts of mathematics: definitions, proofs, sets, functions, relations, counting, Discrete probability theory
- Discrete structures: partial orders, lattices, groups, Boolean algrbra
- Graph Theory and Introduction to Combinatorics
- On completion, students will be able to explain and apply the basic methods of discrete mathematics.
- Students will be able to use the methods leant as part of this subject in subsequent courses in the design and analysis of algorithms, theory of computation, and compilers.
- Students would be able to reason mathematically about basic data types and structures (such as numbers, sets, graphs, and trees) used in computer algorithms and systems; distinguish rigorous definitions and conclusions; synthesize proofs,
- Students would be able to model and analyze computational processes using analytic and combinatorial methods.

CSL213 : Data Structures & Program Design – I (DSPD-I) (L-T-P-C: 3-0-2-8)

Pre-requisite: CSL101 : Computer Programming

- 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

- 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

CSP224 : Software Lab – I (3rd Sem) (L-T-P-C: 0-0-2-2)

- 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 :

- Understand the installation of operating systems
- Understand commands of UNIX and automate tasks using scripts.
- Learn to use gdb and DDD
- Learn to use AWK, SED and Emacs
- Learn to use IDE like DevCpp, Visual Studio etc. to write large software and debug them.

CSL 224 : Introduction to Web Programming(DE) 2 – 0 – 2– 6 (L-T-P-C)

- 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.

ECL208 : Analog Circuits (DE) (L-T-P-C: 3-0-0-6)

Pre-requisite: None

- 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.

CSL222: Computer Organization (DC) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- 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

- 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)

- 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:

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

Reference Books :

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

CSL214 : Data Structures & Program Design – II (L-T-P-C: 3-0-2-8)

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

- 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

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

Course Objectives

- Appreciation of dynamic data structures, advantages and disadvantages.
- Ability to formulate the problem, devise an algorithm and transform into code.
- Ability to identify problem requirements, constraints to be satisfied and ability to select the best possible data structures to satisfy the constraints.
- Ability to analyze the complexity/efficiency of the algorithm and develop ability to improve the same
- Ability to understand how a newer data structure gets designed as per the requirements and constraints.
- Understanding of advantages and disadvantages of different data structures which may be used to solve the same problem
- 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

CSL204 : Concepts in Programming Languages (DC) (L-T-P-C: 3-0-0-6)
Pre-requisite: None

- 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 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

CSL223 : Microprocessor Based Systems (L-T-P-C: 3-0-2-8)

Pre-requisite: ECL209 : Digital Circuits and Logic Design

- 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, PIC 8259, timer 8254 , keyboard / display controller - 8279
- Simple, matrix keyboard , 7 segment LED display
- Introduction to DMA using HOLD/HLDA signals,
- Architecture of 8086 microprocessor, Addressing Modes
- Instruction Set, Assembly language programming.

Text Books :

- Gaonkar R.S.; Microprocessor Architecture, Programming and Applications with 8085/8080A, Wiley Eastern Ltd.
- Liu & Gibson, Microprocessor System: 8086/8088 Family, PHI

References:

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

Course Objectives :

- To understand the basic concept of various programmable devices like microprocessor, microcontroller.
- Appreciation of various issues like internal architecture of the microprocessor, programming fundamentals at assembly language level.
- Ability to write assembly language programs to realize various high level language constructs, considering the architectural features, memory design of the underlying hardware. To realize the issues in computer architecture and organization.
- Ability to interface various programmable devices to the microprocessor and program them to perform data transfer in real life applications.
- Understand the issues related to system design with microprocessor and its peripherals, writing application programs for the designed system

CSL 225 :Advanced Web Programming (DE) 2 -0 – 2– 6 (L-T-P-C)

- 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 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 (L-T-P-C: : 3-0-0-6)

Pre-requisite: None

- 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.

CSL307 : Theory of Computation (DC) (L-T-P-C: 3-0-0-6)
Pre-requisite : None

- 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.

CSL309 : Operating Systems (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite: Data Data Structures

- 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.

Text/References:

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

Course Objectives :

- 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.

CSL313 : Analysis of Algorithm (Credits : 3-0-0-6)(DC)

Pre-requisite:

- 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:

- 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.

CSL303 : Introduction to OO Methodology (IOOM) (DC) (L-T-P-C: 3-0-2-8)

- 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 :

- Appreciation and understanding of object oriented concepts and their utility.
- Ability to formulate the problem, come up with object oriented design
- Practicing use of different features of Object Oriented Methodology like templates, exception handling, reflection etc.
- Study and use of design tools like UML, design patterns etc.
- Study different systems and apply different design methodologies based on the problem specification and objectives

CSP314 : Software Lab – II (L-T-P-C: 0-0-2-2)

- 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 :

- 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.

CSL306: System Programming (DC) (L-T-P-C: 3-0-2-8)
Pre-requisite: None

- 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 definitation within macro, recursive macro calls, design of macro processor.
- 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.
- System utilities - Source code control system, make, link editor, symbolic debugger, pattern matching language like awk.
- Device Drivers - Device programming, system drivers, non system drivers, virtual drivers, Incorporation of driver routines, Basic device driver operation, character and block drivers.
- Lexical Analysis - Role of lexical analyzer, recognition of tokens, tool for study of lex.

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

Text/References:

- The Intel Microprocessors by Barry Brey, Eight Edition, Prentice Hall pub.
- System Software by Beck and Manjula, Third Edition, Pearson pub.
- System Software by J. Nityashri, Second Edition, Tata McGrawHill pub.
- Linkers and Loaders by John Levine, Morgan Koufmann publication.

Course Objectives :

- To understand the basic machine structure.
- Able to understand any microprocessors working, programming, addressing modes, machine conversions etc.
- To know the importance of system programming.
- To study the working of different system programs
- Able to design the procedures for different system programs.
- Able to design own system programs in future.

CSL304 : Neuro Fuzzy Techniques (NFT) (L-T-P-C: 3-0-0-6)

- 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 :

- To familiarize with neural networks and learning methods for neural networks;
- To introduce basics of genetic algorithms and their applications in optimization and planning;
- To introduce the ideas of fuzzy sets, fuzzy logic and fuzzy inference system;
- To introduce techniques of Soft Computing;
- To develop skills thorough understanding of the theoretical and practical aspects of Soft Computing.

CSL305 : Computer Graphics (DE) (L-T-P-C: 3-0-0-6)

Pre-requisite:

- 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.

CSL315 : Database Management Systems (DC) (L-T-P-C: 3-0-2-8)

Pre-requsite :

- 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.

CSL316: Language Processors (DC) (L-T-P-C: 3-0-2-8)

Pre-requisite:

- 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

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 :

- To inform students about different parsing techniques, techniques to generate intermediate code and different optimization techniques.
- The student should be able to analyze issues associated with the implementation of higher-level programming languages.
- This course introduces students how a compiler translates the higher level language into machine language.
- Students should know in detail algorithms used in creating a correct and optimized translation.
- Students should be able to build a working compiler of their own.
- The students will also appreciate the need of understandable error reports, accurate and reliable object code, and strict adherence to industry standards.
- Understanding of compiler optimization techniques would enable students to write reasonably efficient programs.

CSL308 : Software Engineering (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite: CSL 201 and CSL 101

- 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

CSL317 : Computer Networks (L-T-P-C: 3-0-2-8)

- 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.

HUL301 : Technical Communication (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite: None

- Defining technical writing - producing the product - objectives - audience recognition and involvement.
- Correspondence - memos - letters - job search.
- Visual appeal - document design - graphics - electronic communication - email, online help and websites.
- Technical applications - descriptions - instructions and user manuals.
- Report strategies - research - the summary - reports, proposals and oral presentations.
- Handbook of grammar, punctuation, mechanics and spelling.

Text / References :

- Gersen and Gersen ; Technical Writing: Process and Product ; Pearson Education Asia.
- Rutherfoord; Basic Communication Skills for Technology; Pearson Education Asia.
- Lesikar et al; Lesikar's Basic Business Communication; Tata McGraw

CSL 318 : Business Information Systems (L-T-P-C: 3 – 0 – 0 – 6)

- 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:

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

References:

1. 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

CSL319: Internet Technology (L-T-P-C: 3-0-0-6)

- Evolution of Internet, TCP/IP: addressing and routing. Internet applications: FTP, Telnet, Email, Chat. World Wide Web: HTTP protocol. Designing web pages: HTML, forms, CGI scripts and clickable maps, JAVA applets, JAVAscript, JAVA servlets, Perl. DHTML, XML.

- E-Commerce and security issues including symmetric and asymmetric key, encryption and digital signature, authentication. Emerging trends, Internet telephony, virtual reality over the web, etc. Intranet and extranet, firewall design issues.

Text Books:

1. Douglas Comer : *Internetworking with TCP/IP Volume I : Principles, Protocols and Architecture*
2. D. Comer and Douglas Stevens: Internetworking with TCP/IP Volume II : Design, Implementation and Internals
3. D. Comer and Douglas Stevens: Internetworking with TCP/IP Volume III : Client Server Programming and Applications

Reference Books :

1. Richard Stevens : TCP/IP Illustrated, Vol I, II and III.

Course Objectives :

- Aware about different tools for Web Programming
- Background of working on web and security issues.

CSP438 : Software Lab – III (7th Sem) (L-T-P-C: 0-0-2-2)

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.
- Learning software engineering tools
  - Design tools - Rational Rose / visio

- Memory leaks - Purify /
- Code Coverage tools
- Testing tools – Loadrunner, Winrunner

Course Objectives :

- To understand the use of Apache Web server and installation.
- Use of MySql server and its installation.
- Operations on database via C, C++ programs.
- To study automatic testing tools.
- To study the uses of open source cloud platform
- Downloading and installing Hadoop
- Creating web services , learning software engineering tools.

CSL408 : Topics in Embedded Systems (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite Course :

- 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.

CSL412 : Artificial Intelligence (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite Course : CSL313 :Analysis of Algorithms

- 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 :

- Appreciation of fundamental problems in artificial intelligence (AI).
- Ability to generate precise formulation(s) of AI problems in terms of knowledge representation and search from imprecise English description(s).
- Ability to design intelligent agents for problem solving, reasoning, planning, and decision making.
- Ability to make intelligent choices from among available algorithms and knowledge representation schemes subject to specific design and performance constraints.
- Ability to implement and evaluate intelligent agents for representative AI problems – e.g., constraint satisfaction, automated theorem proving, etc.
- Familiarity with some current applications of AI.
- Ability to communicate effectively about AI problems, algorithms, implementations, and their experimental evaluation.

CSL522: Advances in Compiler Construction (DC) (L-T-P-C: 3-0-2 -8)
Pre-requisite Course :

- 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 :

- 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

CSL517 : Pattern Recognition (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite:

- 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 generate pattern features using various transforms based on data.
- Understand how to analyze pattern features using probability theory.
- Understand how to build classifiers using known probability distribution.
- Understand how to build classifiers using non parametric methods.
- Understand how to build linear classifiers using perception model.
- Understand how to build linear, nonlinear classifiers using SVM model.
- Understand how to build classifiers using syntactic model.
- Understand theory of unsupervised learning.

CSL440: Fundamental Algorithms in Computational Biology 3-0-0-6

Pre-requisite : MAL205: Numerical Analysis and Probability Theory

CSL313 **:** Analysis of Algorithms

- DNA and Sequence Alignment – KMP-algorithm, BLAST and FASTA, Sorting by Reversals
- Biological Databases – formats, downloading and using data
- Phylogeny – Distance based algorithms (Hamming /Euclidian distance), Suffix Trees
- Prediction of RNA secondary structure
- Gene Prediction using Bayesian Methods and Markov Chains/HMMs
- Modeling
  - Based on Cellular Automata
  - Based on Agent Based Modeling Techniques
  - Based on Partial Differential Equations
- Single Nucleotide Polymorphism and algorithms for their identification
- Microarray Data and Clustering – Hierarchical/K-Means
- Pathway Data and their analysis
- Protein Folding and Docking based on Entropy calculation

Text Books:

- S. P. Ellner and J. Guckenheimer, Dynamic Models in Biology, Princeton University Press
- J. D. Murray, Mathematical Biology: An Introduction, Springer
- I. M˘andoiu and A. Zelikovsky, Bioinformatics Algorithms - Techniques and Applications, John Wiley & Sons ltd.

References:

Course Website of Ron Shamir, Tel Aviv University (http://www.cs.tau.ac.il/~rshamir/algmb/algmb-archive.htm

Course Code & Title: - CSL523 : Advanced Computer Architecture (DE) (L-T-P-C: 3-0-0-6)

Pre-requisite:

- 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

CSL441: Paradigms in Programming (L-T-P-C: 3-0-0-6)

Pre-requisite : CSL204: Concepts in Programming Languages

CSL303: Introduction to OO Methodology

- 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

1. Harold Abelson, Gerald Jay Sussman and July Sussman, Structure and Interpretation of Computer Programs, 2nd edition, The MIT Press, 1996.
2. 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

ECL4xx: Digital Signal Processing (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- 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

CSL406: Network Security (L-T-P-C: 3-0-0-6)

- 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; Hash and MAC: SHA-512
- Key Management: Digital Certificates, PKI; Authentication: One-Way Authentication, Mutual Authentication, Dictionary Attacks, Centralized Authentication, The Needham-Schroeder Protocol, Kerberos
- Network Layer Security: IPSec; Transport Layer Security: SSL/TLS
- Non-cryptographic Protocol Vulnerabilities: DoS and DDoS, Session Highjacking and Spoofing, ARP Spoofing and Attacks on DNS
- Software Vulnerabilities: Phishing, Buffer overflow, cross site scripting and SQL injection
- Viruses, Worms, and other Malware: Virus and Worm Features, Internet Scanning Worms, Mobile Malware and Botnets
- Access Control in Operating Systems: Discretionary Access Control, Mandatory Access Control, Role Based Access Control, SELinux and Recent Trends
- RFIDs and E-Passports
- Electronic payment

Textbooks

- Forouzan, Cryptography and Network Security, TMH
- Bernard Menezes, Network Security and Cryptography, Cengage
- Radia Perlman Network Security: Private Communication in a Public World, Prentice Hall 2002

References

- Bruce Schneier Applied Cryptography , 2nd Edition John Wiley & Sons 1996
- Douglas Stinson Cryptography Theory and Practice CRC Press 1995
- Alfred Menezes, Paul van Oorschot, Scott Vanstone Handbook of Applied Cryptography CRC Press 1997
- Pfleeger and Pfleeger, Security in Computing, Pearson

Course Objectives :

- 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.

CSL407 : Data Mining & Data Warehousing (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- 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 :

- 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.

CSL410 : Topics in Graph Theory (DE) (L-T-P-C: 3-0-0-6)

Pre-requisite Course :

- 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 :

- 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.

CSL409 : Topics in Distributed Systems (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite Course : Operating Systems

- 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.

CSL411 : Software Project Management (L-T-P-C: 3-0-0-6)

- 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:

1. R.K. Wysocki et al. : Effective Project Management: Traditional, Agile, Extreme, 5th Edition, Wiley India, 2011.
2. 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

CSL439 : Human Computer Interface 3-0-0-6

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

CSL303:Introduction to Object Oriented Methodology

- 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).

CSL 430 Business Intelligence (DE) 3 – 0 – 0 - 6

- **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

CSL 431 - Introduction to Cloud Computing (DE)3 – 0 – 0 - 6

- 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 :

- To study motivation, architecture and types of clouds. To explore different aspects of cloud technology like virtualization and data storage management. To be introduces to practical aspects like cloud economics and cloud programming frameworks.

CSL 437: Enterprise Resource Planning (ERP) DE 3-0-0-6

- 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

1. Enterprise Resource Planning: Mary Sumner, Pearson Fifth Edition, 2009
2. Enterprise Resource Planning: Alexis Leon Tata McGraw-Hill

1. Reference BooksConcepts in Enterprise Resource Planning: Bret Wagner, Ellen Monk, Ceneage Learning India, 2012
2. 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.

MAL 407 : Statistics & O.R. Techniques 3 – 0 – 0 – 6

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

PHY4xx: Quantum Computation and Quantum Information 3-0-0-6

Pre-requisite – PHL101: Physics

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

iv) Scheme of Examination / Instruction – M.Tech. Computer Science Engineering Credit Requirements :

Category	Credits
Departmental Core	72
Departmental Electives	32
Basic Sciences / First Year	00
HM	00
OC	00
Total	**104**

I Semester				II Semester			
Code	Course	L-T-P	Cr	Code	Course	L-T-P	Cr
Core				**Core**			
CSL 514	Advances in Algorithms	3 0 0	6	CSL528	Cryptography & Information Security	3 0 0	6
CSL 523	Advanced Computer Architecture	3 0 0	6	CSL519	Distributed Systems	3 0 2	8
CSL 522	Advances in Compiler Construction	3 0 2	8	CSL520	Distributed and Parallel Databases	3 0 0	8
CSP 540	Software Lab	0 0 4	4	CSP529	Technical Writing and Publishing	0 0 2	2
Elective				**Elective**			
CSL 517	Pattern Recognition	3 0 2	8	CSL516	Soft Computing Techniques	3 0 0	6
				CSL521	Software Architecture	3 0 0	6
				CSL530	Topics in Bioinformatics	3 0 0	6

III Semester				IV Semester			
Code	Course	L-T-P	Cr	Code	Course	L-T-P	Cr
Core				**Core**			
CSD 501	Project Phase I	0 0 0	6	CSD 502	Project Phase II	18	
Elective				**Elective**			
CSL 524	Real Time Systems	3 0 0	6				
CSL 539	Formal Methods in Program Design	3 0 0	6				
CSL 515	Mobile Communication Systems	3 0 0	6				
CSL509	Cloud Architecture, Infrastructure and Technology	3 0 0	6				

V. Detailed Syllabus :

CSL514 : Advances in Algorithm (DC) (L-T-P-C: 3-0-0-6)

Pre-requisite:

- Asymptotic complexity, Algorithmic paradigms: Dynamic Programming, Greedy, Branch-and-bound
- Amortized analysis, Graph Algorithms: Shortest paths, Flow networks; NP-completeness, Approximation algorithms, Randomized algorithms.
- Linear programming, Special topics: Geometric algorithms (range searching, convex hulls, segment intersections, closest pairs), Numerical algorithms (integer, matrix and polynomial multiplication, FFT, extended Euclid's algorithm, modular exponentiation, primality testing, cryptographic computations , Internet algorithms (text pattern matching, tries, information retrieval, data compression, Web caching)

Text / References :

- Introduction to Algorithms : Cormen, Leiserson and Rivest : PHI
- Computer Algorithms : Baase, Gelder : Pearson Education

Course Objectives :

- Appreciate the need for analysis of algorithms.
- How to analyze the best-case, average-case and 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
- NP-completeness.
- Study approximation algorithms and randomized algorithms to address the limitations on the time complexity of complexity

CSL523 : Advanced Computer Architecture (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- 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 Objectives :

- To study qualitative principles in designing a processor, pipelined processors and their variations, issues in multi-processor and multi-core computers.

CSL522: Advances in Compiler Construction (DC) (L-T-P-C: 3-0-2 -8)
Pre-requisite Course :

- 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 interprocedural optimization, instruction scheduling. Compiling for High performance architectures, Compiling for scalar pipeline, compiling for vector pipeline, superscaler and VLIW processors, compiling for multiple issue processors, compiling for memory, hierarchy.

Text/ References :

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

Course Objectives :

- 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

CSP540 : Software Lab (L-T-P-C: 0-0-4-4)

1. Unix/Linux Lab – 1
    a. Common Commands – ls, passwd, 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, cksum, cmp, ln, lynx, gzip, gunzip
    b. Piping and redirection
    c. Editing, Scripting and Pattern Matching – vi, emacs, awk, sed, bash script – variables, conditionals, and loops
    d. Parameter passing to C program from shell (argc / argv)
    e. 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.
    f. DDD (Data Display Debugger) – introduction and usage

2. Unix/Linux Lab – 2
    a. Makefile – writing Makefile, compilation via make, compilation of C programs distributed in multiple files
    b. Versioning system like CVS – versioning and branching
    c. Linux System Administration
        i. Installing Linux in a virtual machine
        ii. Mounting/Unmounting Disks
        iii. Setting and using Path and Environment Variables
        iv. Starting telnet, ftp, smtp services and using them
        v. rcp, rsh, rlogin
        vi. Super user commands/priviledges – su, sudo, install/uninstall of packages, updating linux system
        d. OS related exercises – Accessing iNode, Creation of threads, fork / join, creation of semaphore / mutex, assignments on synchronizing threads. Pthreads and Java threading APIs.

3. Web Technologies and Networking Lab
    a. Creating your own homepage - Creating and publishing your own web-page having links and conditional display using JavaScript
    b. HTML, XML, XSD and HTML / XML parsing
    c. J2EE/.Net introduction: Using Eclipse and VisualStudio to create webpages
    d. JavaScript and JavaScript debugging

e. PHP/Perl/Python/Ruby scripting
f. Networking Commands – inetd, host, ifconfig, netstat, nslookup, ping, ssh, traceroute
g. Network Monitoring tools – Nagios, Wireshark, OpenNMS

4. Open Sources Lab

a. Using automatic testing tools – Junit, NUnit
b. Installing and using Apache Web server to develop a website, Installing and using mysql to store some data
c. Use of open source cloud platforms:
    i. Integration of gmail with google calendar – from gmail you should be able to schedule an appointment with all the recipients of the mail.
    ii. Creating a website on Salesforce cloud for tracking inventory from east, west, north, south regions in India separately.
    iii. Accessing google-map via google-map APIs
d. Java Native Interface (JNI) – Calling C / C++ code from Java and vice versa
e. Downloading and Installing Hadoop on 3 to 4 machines and writing a distributed sorting program on the same.
f. Creating web-services using Axis-2 (Java) or gSoap library (C / C++)
g. 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
h. Introduction to Android Platform and APIs / libraries provided. A sample game / application on Android.

Course Objectives :

- The students should in-depth understanding of the Linux operating system and reveals critical commands and options of using the Linux for any application.
- To understand application development, debugging and code management under Linux platform.
- Learn how to develop for and port applications to the Linux environment. Get to know the necessary tools for Linux application development and learn about special features offered by Linux.
- To inform students about various elements that make the web work, and the way they relate to PHP, Python, HTML, XML and JavaScript. The three after this one are more practical, and show some of the ways that can be used to inspect and change a web form and a web-page.
- To substantially strengthen students' programming ability by requiring them to program a number of large, interesting problems.
- To improve students understanding of object-oriented principles.
- To provide exposure to a broad range of programming areas including multi-threaded programs, communication between processes, and interacting with databases.
- To provide team programming experience.

CSL517 : Pattern Recognition (DC) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- 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 generate pattern features using various transforms based on data.
- Understand how to analyze pattern features using probability theory.
- Understand how to build classifiers using known probability distribution.
- Understand how to build classifiers using non parametric methods.
- Understand how to build linear classifiers using perception model.
- Understand how to build linear, nonlinear classifiers using SVM model.
- Understand how to build classifiers using syntactic model.
- Understand theory of unsupervised learning.

CSL528 Cryptography and Information Security (L-T-P-C: 3-0-0-6)

- Classical Ciphers: Affine, Vignere, Playfair , Hill Cipher
- Cryptanalysis: Differential and Linear
- Algebraic Structures: Groups, Rings, Finite and Galois Fields
- Random number generators and LFSRs
- Modern Block and Stream Ciphers: DES, 2DES, 3DES, AES, RC4, A5/1
- Block Modes of Operation: ECB, CBC, CFB, OFB, CTR
- Basic Number Theory: Primes, Congruences, CRT, Modular Exponentiation, DLP
- Asymmetric Key Cryptosystems: RSA, Rabin,. Elgamal, Attacks on RSA
- Elliptic Curve Cryptography
- Random Oracle Model: Pigeonhole principle, Birthday problems
- Hash and MAC: SHA-512, Whirlpool
- Digital Signatures: RSA, Elgamal. Schnorr, DSS
- Key Management: Kerberos, Diffie-Hellman, Digital Certificates, PKI
- Authentication: One-Way, Mutual, Dictionary Attacks, Centralized Authentication, The Needham-Schroeder Protocol, Zero Knowledge Techniques, Information Theory, Error Correcting Codes, Quantum Techniques in Cryptography
- Network Layer Security: IPSec, Transport Layer Security: SSL/TLS, Wireless LAN Security
- Non-cryptographic and Software Protocol Vulnerabilities, Email Security: PGP, SMIME
- Viruses, Worms, and other Malware , Firewalls

Textbooks

1. Trappe, Introduction to Cyptography with Coding Theory, 2nd edition Pearson
2. Forouzan, Cryptography and Network Security, TMH
3. Bernard Menezes, Network Security and Cryptography, Cengage
4. Radia Perlman Network Security: Private Communication in a Public World, Prentice Hall 2002

References

1. Bruce Schneier Applied Cryptography , 2nd Edition John Wiley & Sons 1996
2. Douglas Stinson Cryptography Theory and Practice CRC Press 1995
3. Alfred Menezes, Paul van Oorschot, Scott Vanstone Handbook of Applied Cryptography CRC Press 1997
4. Pfleeger and Pfleeger, Security in Computing, Pearson

CSL519 Distributed Systems (DC) (L-T-P-C: 3-0-2-8)
Pre-requisite:

- Architecture of Distributed Systems, Distributed Operating Systems, Global Knowledge, Naming, Scalability, Compatibility, Process Synchronization, Resource management, Security, Structuring, Client Server, Computing model, Message passing model, Remote procedure calls.
- Theoretical foundations, Inherent limitations of distributed systems, Lamports logical clocks, Vector clocks, Causal ordering of messages, Global state recording, Cuts of a distributed computation, termination detection.
- Distributed Mutual Exclusion, Classification of mutual exclusion algorithms, Requirements and performance measures of mutual exclusion algorithms, Non-token based algorithms, Token based algorithms, Comparative performance analysis.
- Distributed Deadlock Detection- Resource vs Communication deadlocks, graph theoretic model, deadlock prevention, avoidance, detection, Issues in deadlock detection, Centralized deadlock detection algorithms, distributed deadlock detection algorithms, Hierarchical deadlock detection algorithms.
- Agreement Protocols – Synchronous vs. asynchronous computations, model of process failures, authenticated vs. non-authenticated messages. A classification of Agreement problems, Solutions to Byzantine Agreement problem, applications of Agreement Algorithms.
- Failure recovery and Fault Tolerance, classification of failures. Backward and forward error recovery, Basic approaches of backward error recovery, recovery in concurrent systems, consistent set of checkpoints, synchronous checkpointing and recovery, asynchronous check pointing and recovery.
- Atomic actions and committing, commit protocols, non-blocking commit protocols, Votic protocols, Dynamic vote re-assignment protocols, failure resistant processor, Reliable communication.
- Distributed File systems, mechanisms, design issues, case studies. Distributed shared memory, Algorithms for implementing DSM, memory coherence, coherence protocols, Design issues, case studies.
- Distributed scheduling, Load distributing, load balancing vs. load sharing, components of a load distributing algorithm, stability, Load distributing algorithms, Performance comparison, task migration.

Text/References

- Singhal M. & Shivratri N, “Advanced concepts in operating systems”, McGraw Hill
- Lynch N., Morgan Kaufman, “Distributed Algorithms”,
- Tanenbaum, “Modern Operating Systems”, PHI EEE

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.

CSL520 Distributed and Parallel Databases (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- Architecture of Parallel Databases, Database design in parallel Database, Scheduling in parallel Database, Architecture of Distributed Database, Client Server Systems and collaborative systems.
- Storage Management in parallel Database and Distributed Database, horizontal and vertical fragmentation, Data partitions and clusters, indexing techniques.
- Transaction and concurrency control, Two phase commit, implementation of ACID properties in parallel and distributed Database.
- Query optimization, computation of join costs and access costs, semijoins and antijoins, Using Heuristics in Query optimization.
- Backup, and Recovery concepts, Transaction recovery, replication concepts, multimaster and snapshot replication conflict resolution.

- Application of parallel and distributed database in highly transactional and Data mining systems, use of parallel and distributed database in data warehousing.

Text/References:

- D. Bell and J.Grimson, "Distributed Database Systems", Addison-Wesley 1992
- S.Ceri and G.Pelasath, "Distributed Database Systems", McGraw Hill
- M. O2S4, "Principles of Distributed Database Systems", Prentice Hall , 1991

Course Objectives :

- Understand how to optimize database management system.
- Understand how to handle large concurrent operations.
- Understand how to ensure durability of data.
- Understand how to handle distributed database.
- Understand how to design systems for parallel databases.
- Understand new concepts like BIGDATA and columnar databases.

CSP529 : Technical Writing and Publishing (L-T-P-C: 0-0-2-2)

- Presenting a book chapter using powerpoint slides
- Data Analysis: Maintaining multiple results obtained over time and reporting them using charts and graphs
- Technical Documentation – Requirement/specification documentation, Design documentation, Test-cases documentation, Use-cases documentation
- Writing an installation/instruction manual
- Writing an abstract of a technical article – summarizing an article in 300 words
- Summarizing 3 papers into a report and its presentation

Reference Books:

- Strunk and White : The Elements of Style
- Gretchen Hargis et. al. : Developing Quality Technical Information: A
- Handbook for Writers and Editors, Second Edition, IBM, 2004.
- Leslie Lamport : LaTex

Course Objectives :

- To inform students about writing by engineers and computer scientists, to engineers, engineering managers, and technical writers.
- Assignments and projects include job application and resume, in-code documentation, algorithm description, survey article, proposal, progress report, formal technical report, summarization and oral presentation.
- Students would be able to effectively use Google search, understand and use manuals, create installation/instruction manuals, understand the standards and RFC's, and Present a book chapter using powerpoint slides

CSL516: Soft Computing Techniques (DC) (L-T-P-C: 3-0-2 -8)

Pre-requisite Course :

- **Neural Networks:** History, overview of biological Neuro-system, Mathematical Models of Neurons, ANN architecture, Learning rules, Learning Paradigms-Supervised, Unsupervised and reinforcement Learning, ANN training Algorithms-perceptions, Training rules, Delta, Back Propagation Algorithm, Multilayer Perceptron Model, Hopfield Networks, Associative Memories, Applications of Artificial Neural Networks.
- Fuzzy Logic: Introduction to Fuzzy Logic, Classical and Fuzzy Sets: Overview of Classical Sets, Membership Function, Fuzzy rule generation. Operations on Fuzzy Sets: Compliment, Intersections, Unions, Combinations of Operations, Aggregation Operations.
- Fuzzy Arithmetic: Fuzzy Numbers, Linguistic Variables, Arithmetic Operations on Intervals & Numbers, Lattice of Fuzzy Numbers, Fuzzy Equations. Application of Fuzzy Logic: Medicine, Economics etc.
- Introduction of Neuro-Fuzzy Systems, Architecture of Neuro Fuzzy Networks.
- Genetic Algorithm: An Overview of GA, GA operators, GA in problem solving, Implementation of GA.
- **Laboratory Work:** Design and implementation of neural networks, implementation of genetic algorithms in various search problems.

Text/References:

- Roger Jang, Tsai Sun, Eiji Mizutani, "Neuro-Fuzzy and soft Computing: A computational Approach to Learning & Machine Intelligence", PHI

- John Hertz, Anders Krogh, Richard Palmer, "Introduction to The theory of Neural Computation", Addison Wesley
- Timothy Ross, "Fuzzy Logic with Engineering Applications", McGraw-Hill
- R.A. Aliev, R.R. Aliev, "Soft Computing and Its Applications", World Scientific
- Kishan Mehrotra, C. K. Mohan, S. Ranka, "Elements of Artificial Neural Networks", Penram International Publishing (India)
- Bar Kosko, "Neural Networks and Fuzzy Systems", Prentice-Hall
- B. Yegnanarayana, "Artificial Neural Network", PHI
- Simon Haykin, "Neural Networks: A Comprehensive Foundation", PHI

Course Objectives :

- To familiarize with neural networks and learning methods for neural networks;
- To introduce basics of genetic algorithms and their applications in optimization and planning;
- To introduce the ideas of fuzzy sets, fuzzy logic and fuzzy inference system;
- To introduce students tools and techniques of Soft Computing;
- To develop skills thorough understanding of the theoretical and practical aspects of Soft Computing.

Course Code & Title: - CSL521 Software Architecture (DE) (L-T-P-C: 3-0-0-6)
Pre-requisite:

- Software process and the role of modeling and analysis, software architecture and software design.
- Software Modeling and Analysis: Analysis modeling and best practices, traditional best practice diagrams such as DFDs and ERDs,
- Software Architecture: architectural styles, architectural patterns, analysis of architectures, formal descriptions of software architectures, architectural description languages and tools, scalability and interoperability issues, web application architectures, case studies.
- Software Design: design best practices, design patterns, design case studies, component technology, object oriented frameworks, distributed objects, interoperability standards, case studies., software quality
- UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns, documenting software architecture, reconstructing software architecture.
- 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.

- **Text/References:**

  - M. Shaw, "Software Architecture Perspectives on an Emerging Discipline", PHI
  - Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Pearson Education Asia
  - 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 Objectives :

- The students will be able to appreciate relationships between system qualities and software architectures
- The students will have the understanding of the software architectural patterns and tactics, and their relationship to system qualities
- The students will be able to make attribute-driven design
- The students will be able to produce software architecture documentation
- The students will be able to do software architecture evaluation
- The students will be able to understand architectural reuse via software product lines
- The students will be able to appreciate architectures in Agile projects

CSL 530 : Topics In Bioinformatics (DE) (L-T-P-C: 3-0-0-6)
Prerequisites: Data Structures and Algorithms courses

- Basics of biology
- Sequences: Problem statement, Edit distance and substitution matrices, Global and local alignments, Spliced alignment, Space-efficient sequence alignment, Multiple alignment,
- Structures: Protein structure alignment, Protein structure prediction
- Phylogenetic trees: Large parsimony and small parsimony problems, Probabilistic approaches, Grammar-based approaches

- Systems Biology: Overview of Gene Control –Working of Genetic Switches – Introductory Systems Biology The biochemical paradigm, genetic paradigm and the systems paradigm; 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.
- Miscellaneous topics: Pathways and networks, Microarrays, Biomedical images

Text/References:

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

Additional Readings

1. Introduction to Algorithms by Cormen, Leiserson, Rivest, Stein. Prentice Hall.

Course Objectives :

- This course introduces students to the basic computational methods and algorithms that can be used to understand the cell on a molecular level and biological system at a macro level.
- 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; protein structure and folding; and molecular dynamics.
- This course will help students develop multidisciplinary approach to the systematic analysis and modeling of complex biological phenomena.
- Serving as an introduction to computational and systems biology, this course emphasizes the fundamentals of nucleic acid and protein sequence analysis, structural analysis, and the analysis of complex biological systems.
- Students are also exposed to currently emerging research areas in the fields of computational and systems biology.

CSL524 : Real Time Systems (DE) (L-T-P-C: 3-0-0-6)

Pre-requisite:

- **Real time applications:** Hard and soft real time systems, timing constraints, A Reference model of Real-time systems, temporal parameters, precedence constraints & dependencies, scheduling Hierarchy, Commonly used approaches to scheduling, cyclic and priority drive approaches, Optimality of EDF and LST
- **Clock Driven Scheduling:** Static timer driven scheduler, Cyclic Executives, Improving Average Response times of Aperiodic Jobs, Scheduling Sporadic jobs, Practical Considerations, Pros and Cons of Clock Driven Scheduling
- **Priority-driven scheduling of periodic tasks:** Fixed priority vs Dynamic Priority schemes, Maximum schedulable Utilization, Optimality of the RM and DM algorithms, As Schedulable Test for Fixed Priority Tasks, Practical Factors
- **Scheduling Apriodic and Sporadic Jobs in Priority-driven scheduling:** Deferrable Servers, Sporadic Servers, Constant Utilization. Total Bandwidth, and Weighted Fair-Queuing Servers, Scheduling of Sporadic Jobs
- **Resources and resource access control:** non preemptive critical sections, basic priority-inheritance, ceiling protocol, multiprocessor scheduling, predictability and validation of dynamic multiprocessor systems flexible applications, tasks with temporal distance constraints.
- **Real time Operating systems:** Overview, Time Services and Scheduling Mechanisms, Basic Operating System Functions, Processor Reserves and Resource Kernel, Open System Architecture, Capabilities of Commercial RTOS.

Text/References:

1. Real-Time Systems :*Jane W.S. Liu, Pearson Education Asia Pub*
2. Real Time Systems : C.M. Krishna & Kang G. Shin : McGraw Hill

Course Objectives :

- Appreciation of the need and the challenges in the design of hard and soft real time systems.
- Understanding the different scheduling techniques and the schedulability criterion.
- Awareness about issues regarding scheduling of sporadic and aperiodic jobs and their applications, e.g. in messae passing in real-time environments.
- Appreciation of the need for integrated mechanism for resource allocation and scheduling.
- Awareness about different RTOSes and their features.
- Ability to choose a scheduling technique for the design of a real time system.
- Ability to choose a suitable RTOS for a particular system.

CSL539: Formal Methods in program Design (L-T-P-C:**3-0-0-6)**

- 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 Books:

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

Reference Books:

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

Course Objectives :

- To study basic concepts used in program design like determinsim/non-determinism, synchrony/asynchrony, separation of concerns like correctness and complexity, programs and implementation. To study the concept of progress and proofs thereof. To study architecture and mappings by taking different case studies. To study program structuring and program composition.

CSL515: Mobile Communication Systems (L-T-P-C: 3-0-0-6)

- Review of radio transmission, antennas, modulation & demodulation, Radio propagation. Concept of cellular working, Multiplexing in space, frequency time, Code division multiplexing, Spread spectrum medium access methods.
- **Wireless telecom Systems:** Evolution, study of 2G system GSM. Network architecture, radio interface, System's internal interfaces, role of VLRs & HLRs. Handover algorithms, security, Operation Maintenance systems.
- **3G Systems & beyond**: Evolution towards 3G systems based on GSM & CDMA networks. Radio interface, system internal functioning, handover scenarios, security.
- **Wireless LAN systems** : Medium access control mechanism in 802.11 networks. Radio interface, protocol architecture.
- Mobile adhoc networks. Algorithms for routing & overall network function. Mobile satellite networks.
- Support for mobility: Mobile IP, TCP for mobile hosts. Other developments in the TCP/IP stack for mobility support.

Texts/References:

1. Mobile Communications - J. Schiller -Pearson Education
2. Introduction to Wireless and Mobile Systems– Agrawal and Zeng -Cengage Learning
3. Wireless and Mobile Network Architecture – Lin and Chlamtac- Wiley
4. Wireless Communications: Principles and Practice- Theodore Rappaport – Prentice Hall
5. Wireless and Mobile All-IP Networks – Lin and Pang- Wiley

Course Objectives :

- Understand the structure and design issues of mobile communication systems.
- Learn about evolution and understand theoretical concepts of modern mobile systems.
- Understand concepts of 2G, WLANs etc..
- Familiarity with adhoc and sensor networks.
- Gain experience of adding support for mobility in the protocol stack.

CSL 509 - Cloud Architecture, Infrastructure and Technology (DE) 3 – 0 – 0 - 6

- Cloud system architectures, Cloud programming frameworks and what is "infrastructure-as-a-service", "platform-as-a-service" and "software-as-a-service" , Cloud computing delivery models – public, private and hybrid clouds, cloud-in-a-box
- "Big data" concepts, storage and management, Security, scalability, privacy, lock-in, and other risks (and mitigations) for individuals and companies
- Virtualization, clustering and resource management, HPC in cloud computing
- Cloud applications considerations for updates, backups, disaster recovery and fault tolerance, Data center networks and Energy use in data centers
- Introduction to cloud enabling technologies: Introduction to Hadoop, Map-reduce, NoSQL, MongoDB, Cassandra, Web Servers, Encryption techniques, SSL
- Case Study: Design of a cloud system

Text Book :

Resse G., Cloud Application Architectures: Building Applications and Infrastructure in the Cloud,O' Reilly.

Reference Book :

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

Course Objectives :

- To study cloud frameworks, Big Data concepts, clustering, resource management and fault tolerance and introduction to cloud enabling technologies.