COMPUTER SCIENCE

III Semester II Year

paper Code	Subject		Teaching hours			Credit point
			L	**T**	**P**	
03BCS101	Digital Electronics		3	0	0	3
03BCS102	Electronic Devices & Circuits		3	1	0	4
03BCS103	Data Structure and Algorithms		3	0	0	3
03BCS104	Discrete Mathematical Structures		3	0	0	3
03BCS105	Mathematics III		3	1	0	4
03BCS106	Electives (Any one of the following)		3	0	0	3
	1	Intellectual Property Rights				
	2	Internet Technology				
	3	Management Information Systems				
03BCS201	Digital Electronics Lab		0	0	3	2
03BCS202	Electronics Lab		0	0	3	2
03BCS203	Data Structure Lab		0	0	3	2
03BCS204	Humanities and Social Science		0	0	3	2
03BCS301	Discipline & Extra Curricular Activities		0	0	4	1
TOTAL			**18**	**2**	**16**	**29**

IV SEMESTER

Subject Code	Name of Subject		Teaching hours			Credit point
			L	T	P	
04BCS101	Principles of Programming Languages		3	0	0	3
04BCS102	Microprocessor and Interfaces		3	0	0	3
04BCS103	Object Oriented Programming		3	1	0	4
04BCS104	System Software		3	0	0	3
04BCS105	Statistics and Probability Theory		3	1	0	4
04BCS106	Electives (Any one of the following)		3	0	0	3
	1	Open Source Technology				
	2	E-Commerce				
	3	Analog & Digital Communication				
04BCS201	Communication Lab		0	0	3	2
04BCS202	Microprocessor Lab		0	0	3	2
04BCS203	Object Oriented Programming Lab		0	0	3	2
04BCS204	System Software Lab		0	0	3	2
04BCS301	Discipline & Extra Curricular Activities		0	0	4	1
TOTAL			**18**	**2**	**16**	**29**

V SEMESTER

Subject Code	Name of Subject		TEACHING HOURS			CREDIT POINT
			L	T	P	
05BCS101	Software Engineering		3	1	0	4
05BCS102	Computer Architecture		3	0	0	3
05BCS103	Database Management Systems		3	0	0	3
05BCS104	Computer Graphics		3	0	0	3
05BCS105	Telecommunication Fundamentals		3	1	0	4
05BCS106	Electives (Any one of the following)		3	0	0	3
	1	Logic & Functional Programming				
	2	Information Theory and Coding				
	3	Advanced Data Structure				
05BCS201	Software Engineering Lab		0	0	3	2
05BCS202	Computer Architecture Lab		0	0	3	2
05BCS203	Database Management Lab		0	0	3	2
05BCS204	Computer Graphics Lab		0	0	3	2
05BCS301	Discipline & Extra Curricular Activities		0	0	4	1
TOTAL			**18**	**2**	**16**	**29**

VI SEMESTER

Subject Code	Name of Subject		TEACHING HOURS			CREDIT POINT
			L	T	P	
06BCS101	Operating Systems		3	1	0	4
06BCS102	Computer Networks		3	0	0	3
06BCS103	Design & Analysis of Algorithms		3	0	0	3
06BCS104	Embedded Systems		3	0	0	3
06BCS105	Theory Of Computation		3	1	0	4
06BCS106	Electives (Any one of the following)		3	0	0	3
	1	Digital Signal Processing				
	2	Advanced Software Engineering				
	3	Microwave and Satellite Communication				
06BCS201	Shell Programming Lab		0	0	3	2
06BCS202	Network lab		0	0	3	2
06BCS203	Web Programming lab		0	0	3	2
06BCS204	Microcontroller lab		0	0	3	2
06BCS301	Discipline & Extra Curricular Activities		0	0	14	1
TOTAL			**18**	**2**	**16**	**29**

VII SEMESTER

Subject Code	Name of Subject	TEACHING HOURS			CREDIT POINT
		L	T	P	
07BCS101	Compiler Construction	3	0	0	3
07BCS102	Data Mining And Ware Housing	3	0	0	3
07BCS103	Logic Synthesis	3	0	0	3
07BCS104	Artificial Intelligence	3	0	0	3
07BCS105	Multimedia Systems	3	0	0	3
07BCS106	Electives (Any one of the following) 1 Service Oriented Architectures 2 Optical Communication 3 Real Time Systems	3	0	0	3
07BCS201	Compiler Design Lab	0	0	3	2
07BCS202	Data Mining And Ware Housing Lab	0	0	3	2
07BCS203	Logic Synthesis Lab	0	0	3	2
07BCS204	Project Stage I	0	0	2	1
07BCS205	Practical Training Seminar	0	0	3	2
07BCS301	Discipline & Extra Curricular Activities	0	0	4	1
TOTAL		**18**	**0**	**18**	**28**

VIIISemester

VIII SEMESTER

Subject Code	Name of Subject		TEACHING HOURS			CREDIT POINT
			L	T	P	
08BCS101	Information System and Securities		3	1	0	4
08BCS102	CAD FOR VLSI Design		3	1	0	4
08BCS103	Advanced computer Architectures		3	1	0	4
08BCS104	Electives (Any one of the following)		3	1	0	4
	1	Distributed Systems				
	2	Image Processing				
	3	Natural Language Processing				
08BCS201	Information System and Securities		0	0	3	2
08BCS202	VLSI Design Lab		0	0	3	2
08BCS203	X-Windows Programming Lab		0	0	3	2
08BCS204	Project Stage II		0	0	6	3
08BCS205	Seminar Presentation		0	0	3	2
08BCS301	Discipline & Extra Curricular Activities		0	0	4	1
TOTAL			**12**	**4**	**22**	**28**

DIGITAL ELECTRONICS

Course/Paper: 03BCS-101
BCS Semester III

Prerequisite:

- Fundamentals of Analog and Digital Electronics
- Introduction to Analog
- Introduction to Digital
- Combinational Logic
- Introduction to NAND and NOR Logic
- Digital Design Project

Course objectives:

This course provides in depth knowledge of switching theory and the design techniques of digital circuits, which is the basis for design of any digital circuit, and to develop an in-depth understanding of the operation of microprocessors. Machine language programming

Unit	Content
I	Number systems, Coding Schemes: BCD, Excess-3, Grey, r's and (r-l)'s complement. Boolean Algebra, Fundamental theorems, Simplifications of Boolean expressions. Logic gates and their truth table. Gate implementation and Truth table of Boolean functions.
II	Standard forms of Boolean functions. Minterm and Maxterm designation of functions. Simplification of functions on Karnaugh maps, Incompletely specified functions. Cubical representation of Boolean functions and determination of prime implicants. Selection of an optimal set of prime implicants. Multiple output circuits and map minimization of multiple output circuits. Tabular determination of multiple output prime implicants.
III	Combinational circuits – Adder, subtractor, encoder, decoder, multiplexer. Design of Combinational circuit using Multiplexers.
IV	Flip Flops: RS, J-K, D, T. Sequential circuits. Clock, pulse and level mode sequential circuits. Analysis and design of sequential circuits. Synthesis of state diagrams, Finite memory circuits, equivalence relations equivalent states and circuits, determination of classes of indistinguishable states and simplification by implicants tables. Mealy and Moore machines, state assignment and memory element input equations, Partitioning and state assignment.
V	Switching Devices. Positive and Negative logic of OR, AND, NOR, NAND, XOR and XNOR gates. Logic Family: RTL, DTL, DCTL, TTL, RCTL, ECL, HTL, MOS and CMOS logic circuit. Speed and delay in logic circuits, integrated circuit logic and noise immunity.

Text books:

1. Albert Paul Malvino and Donald P. Leach, Digital Principles and Applications, (Fourth Edition) Tata Graw Hill Publishing Company Ltd, New Delhi.
2. Advanced microprocessors and peripherals – A. K. Ray & K. M. Bhurchandani, TMH, 2nd Edition 2006
3. Digital logic and state machine design – David. J Comer Oxford University Press

4. R.P. Jain, Modern Digital Electronic, Tata Mc Graw Hill Publishing Company Ltd. New Delhi.

Reference book

1. Salivahanan and S. Arivazhagan, Digital Circuits and Design, Vikas Publishing House Pvt. Ltd.
1. Adel S. Se&a, and Kanneth C. Smith, Microelectronic Circuits, Oxford University Press
2. Microprocessor and interfacing – N. Senthil Kumar, M. Saravanan S. Jeevananthan s. K.
3. Shah Oxford University Press
4. Digital Design. M. Morris Mano.

Course outcomes:

Upon completion of the course, students should possess the following skills:

- Be able to manipulate numeric information in different forms, e.g., different bases, signed integers, various codes such as ASCII, Gray, and BCD.
- Be able to manipulate simple Boolean expressions using the theorems and postulates of Boolean algebra and to minimize combinational functions.
- Be able to design and analyze combinational circuits and to use standard combinational functions/building block to build more complex circuits.

ELECTRONIC DEVICES & CIRCUITS

Course/Paper: 03BCS-102
BCS Semester III

Pre requisite : Analog Circuit Analysis

Course objectives:

This course provides substantial knowledge of the function pn junctions, pn- and zener diodes, bipolar and field-effect devices, basic analog and digital circuits. The course will discuss the device parameters of diodes, BJT's, field effect transistors, single stage amplifiers using common emitter, common base and common collector configurations as well as all configurations of FET amplifiers including CMOS and multi stage amplifiers, differential amplifier and output stages.

Unit	Content
I	Diode circuits: Diode as a circuit. Element, load line concept, clipping & clamping circuits, voltages multipliers.
II	Devices: construction, characteristics and working principles of the following devices. Diodes, BJT, JFET, MOSFET, UJT, photo diodes, LEDs, photo transistors. Solar cells. Thermistor, LDR.
III	Transistors: transistor characteristics, current components, current gains. Alpha and vita. Operating point. High bride model, h- parameter equivalent circuits. Ce, Cb and Cc configuration. Dc and ac analysis of Ce, Cc and Cb amplifiers. Evers-

	moll model. Biasing and stabilization techniques. Thermal run away, thermal stability. Equivalent circuits and blessing of JFETs and MOSFETs. Low frequency Cs and Cd JFET amplifiers. FET as a voltage variable resistor.
IV	Small signal amplifiers at low frequency: analysis of BJT and FET, dc and rc coupled amplifiers. Frequency response, mid band gain, gains at low and high frequency. Analysis of dc and differential amplifiers, Mil ers' Theorem. Cascading transistor amplifiers, Darlington and cascaded circuits. Emitter and source followers.
V	Oscillators: concept of feedback classification, criterion for oscillation. Tuned collector, Hartley, Colpitts , rc- phase shift , Wein bridge and crystal oscil ators, astable, monostable and bistable multivibrators . Schmitt trigger.

Text books:

1. Johnson, DE, Johnson, JR and Hilburn, JL: Electric Circuit Analysis. Prentice Hall, 2nd edition
2. R Boylestad and L Nashelsky: Electronic Devices and Circuit Theory, Prentice Hall
3. Thomas L. Floyd: Digital Fundamentals, 8th Edition, Prentice Hall
4. David F. Hoeschele: Analog-to-Digital and Digital-to-Analog Conversion Techniques, 2nd Edition, John Wiley & Sons

References: Books:

- J. Millman & C.C. Halkias: Integrated Electronics, Tata Mc-Graw Hill Publishing Ltd., New Delhi Mc-Graw Hill.
- Millman Grabel –Micro electronica, Mc-Graw Hill.
- Robert Boylestand & L.Nashelsky Electronic devices & circui theory.
- Sedra Smith- Microelectronic Circuits, Oxford Press, India.

Learning Outcome:

Gain an intuitive understanding of the role of power flow and energy storage in electronic circuits; Develop the capability to analyze and design simple circuits containing non-linear elements such as transistors using the concepts of load lines, operating points and incremental analysis; Learn how the primitives of Boolean algebra are used to describe the processing of binary signals and to use electronic components such as MOSFET's as building blocks in electronically implementing binary functions; Learn how the concept of noise margin is used to provide noise immunity in digital circuits.

DATA STRUCTURE AND ALGORITHMS

Course/Paper:03BCS-103
BCS Semester III

Prerequisites:
Students should have basic knowledge of C programming constructs and should be able to write basic C programs.

Course Objective
This course focuses on the design and analysis of algorithms and the use of data structures. Through the introduction of the most widely used data structures employed in solving commonly encountered problems (e.g. lists, trees, and graphs), students will learn different ways to organize data for easy access and efficient manipulation. Algorithms to solve classic problems (e.g. searching, sorting, hashing, graph algorithms, etc) will be presented, as well as classic algorithm design strategies (e.g. divide-and-conquer and greedy algorithms). Computational complexity theory will be introduced for studying the efficiency of the algorithms covered in the course.

Unit	Content
I	Data Structure: Definition, Implementation, Operation, Application, Algorithm writing and convention. Analysis of algorithm, Complexity Measures and Notations. Arrays: Representation of arrays (multidimensional), Address calculation using column and row major ordering. Linked Lists : Implementation, Doubly linked list, Circular linked list, unrolled linked list, skip-lists, Splices, Sentinel nodes, Application (Sparse Matrix, Associative Array, Functional Programming)
II	Stacks : Definition, Implementation, Application (Tower of Hanoi, Function Call and return, Parentheses Matching, Back-tracking, Expression Evaluation) Queues : Definition, deque, enque, priority queue, bounded queue, Implementation, Application
III	Tree: Definition of elements, Binary trees: Types (Full, Complete, Almost complete), Binary Search Tree, Traversal (Pre, In, Post & Level order), Pruning, Grafting. Application: Arithmetic Expressions Evaluation Variations: Indexed Binary Tree, Threaded Binary Tree, AVL tree, Multi-way trees, B tree, B+ tree, Forest, Trie and Dictionary
IV	Graphs: Elementary definition, Representation (Adjacency Matrix, Adjacency Lists) Traversal (BFS, DFS)Application: Spanning Tree (Prim and Kruskal Algorithm), Dijkstra's algorithm, and Shortest path algorithms.
V	Sorting: Bubble, Selection, Insertion, Quick, Radix, Merge, Bucket and Heap sorts. Searching: Hashing, Symbol Table, Binary Search, Simple String Searching.

Text Books

1. Programming in ANSI C:E Balagurusamy, Third edition, TATA McGraw HILL
2. Data Structure using C, Aaron M. Tenenbaum, Yedidyah Langsam &Moshe J. Augenstein, Pearson Education/PHI, 2006
3. Ellis Horowiz and Sartaj Sahani : Fundamentals of data structure with Pascal. Galgotia Book Source, 1994.

References:

- Aho A.V & Ullman J.E. : Data Structure & Algorithms.
- Aron M. Tannenbaum & Others: Data Stmctures usmg C, Prentice Hall, 1992 ,
- Mary E.S. Loomis: Data Management & File Structure, PHI, 1991.
- Bhagat Singh & Thomas Naps: Introduction to Data Structure
- Trembley & Sorenson: An Introduction to Data Structure with Application, McGraw-Hill, 1984.

Learning Outcome

Students who complete this course will be able to:

- Write programs that use data structures such as: arrays, linked lists, stacks, queues, trees, hash tables, and graphs.
- Compare and contrast the cost and benefits of dynamic and static structure implementations.
- Choose the appropriate data structure for modeling a given problem.
- Describe the concept of recursion and give examples of its use, identifying the base case and the general case of a recursively defined problem.
- Compare iterative and recursive solutions for elementary problems.
- Determine when a recursive solution is appropriate for a problem.
- Determine the time and space complexity of simple algorithms and recursively defined algorithms.
- Implement both a greedy and a divide-and-conquer algorithm to solve problems.
- Implement the most common sorting algorithms.
- Design and implement an appropriate hash function.
- Design and implement a collision-resolution algorithm for a hash table.
- Solve problems using the fundamental graph algorithms, including depth-first and breadth- first search, topological sort, minimum spanning tree algorithm, and single-source shortest path.

DISCRETE MATHEMATICAL STRUCTURES

Course/Paper: 03BCS-104
BCS Semester III

Prerequisites:

You should be familiar with sequences and series, limits, and integration and differentiation of univariate functions.

Course objective:

This course provides a foundation for Computer Science. Many other areas of Computer Science require the ability to work with concepts from discrete mathematical structures. Discrete structures include topics such as set theory, logic, graph theory, and probability theory. The material in discrete structures is pervasive in the areas of data structures and algorithms but appears elsewhere in Computer Science as well. In this course, you will learn about (1) sets, relations and functions; (2) basic logic, including propositional logic, logical connectives, truth tables, propositional inference rules and predicate logic; (3) proof techniques, including the structure of mathematical proofs, direct proofs, disproving by counterexample, proof by contradiction; (4) basics of counting, including counting

arguments, the pigeonhole principle, permutations and combinations, solving recurrence relation; (5) graphs and trees; and, (6) discrete probability, including finite probability space, axioms of probability, conditional probability.

Unit	Content
I	Formal Logic: Statement, Symbolic Representation and Tautologies, Quantifiers, Predicator and validity, Normal form. Propositional Logic, Predicate Logic, Logic Programming and Proof of correctness.
II	Proof, Relation and Analysis of Algorithm Techniques for theorem proving: Direct Proof, Proof by Contra position, Proof by exhausting cares and proof by contradiction, principle of mathematical induction, principle of complete induction. Recursive definitions, solution methods for linear, first-order recurrence relations with constant coefficients.
III	Graph Theory: Graphs - Directed and Undirected, Eulerian chains and cycles Hamiltonian chains and cycles, Trees, chromatic number, connectivity and other graphical parameters Applications. Polya's Theory of enumeration and its applications.
IV	Sets and Functions: Sets, relations, functions, operations, equivalence relations, relation of partial order, partitions, binary relations. Transforms: Discrete Fourier and Inverse Fourier Transforms in one and two dimensions, discrete Cosine transform.
V	Monoids and Groups: Groups, Semi groups and Monoids cyclic semi graphs and sub monoids, Subgroups and cosets. Congruence relations on semi groups. Morphism, Normal sub groups. Structure off cyclic groups, permutation groups and dihedral groups elementary applications in coding theory.

Textbook:

1. Discrete Mathematics and its Applications - Kenneth H. Rosen 7th Edition -Tata McGraw Hill Publishers – 2007
2. Robert J.Mc Eliece : Introduction to Discrete Mathematics, Tata Mc. Graw Hill, India.
3. Kenneth H. Rosen, Discrete mathematics and Applications, Tata Mc. Graw Hill, India.
4. C.l.Liu ; elements of Discrete Mathematics Tata McGraw Hill publishing Company Ltd., 2000

References:

- 1.Richard johnsonbaugh disvrete mathematics prearson Asia 2001.
- John Truss : Discrete Mathematics for Computer Scientists, Pearson Education, Asia, 2001.
- Lipschutz : Discrete Mathematics, Tata Mc. Graw Hill India.

Learning Outcomes

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

- solve problems which require knowledge of elementary discrete probability concepts
- solve problems which involve discrete data structures such as sets, relations and discrete functions.
- construct valid mathematical arguments (proofs) and understand/apply mathematical statements (theorems)

- solve problems which require computation of permutations and combinations of a set
- analyze a problem to create relevant recurrence equations and solve the equations
- apply basic counting principles to solve a variety of problems
- apply the mathematical concepts learned to various areas of computer science

MATHEMATICS III

Course/Paper: 03BCS-105
BCS Semester III

Prerequisite: MATH 1140 OR COMP 1300

Course objectives:
This course is designed to introduce students to the techniques, algorithms, and reasoning processes involved in the study of Mathematical Foundation of Computer Science. Students will be introduced to set theory, elementary and advanced counting techniques, equivalence relations, recurrence relations, graphs, and trees. Through their study of these topics students will develop a greater understanding of the breadth of mathematics and will acquire a familiarity with concepts, structures and algorithms that are essential to the field of computer science and applied mathematics.

Unit	Content
I	Introduction: Engineering application of optimization, Statement and classification of optimization problem, single variable and multivariable optimization with and without constraints.
II	Linear Programming: Formulation of Linear Programming problem, Graphical Approach, General Linear Programming problem, Simple Method. Duality in Linear Programming and Transportation Problems.
III	Project Scheduling: Project Scheduling by PERT and CPM Network Analysis. Sequencing Theory: General Sequencing problem n-jobs through 2 machines & 3 machines and 2-jobs through machine.
IV	Transform Calculus – Laplace Transform with its simple properties, applications to the solution of ordinary and partial differential equation having constant coefficients with special reference to the wave and diffusion equation. Fourier transforms and solution of particular differential equation with constant coefficient.
V	Numerical Methods – Solution of Algebraic and transcendental equations, interpolation- finite differences, inverse interpolation, numerical differentiation and integration, numerical solution of differential equations and partial differential equations, solution of difference equation.

Text Books :

1. Gokhroo et al:Higher Engg mathematics-III
2. Grewal- Maths for Engineers.
3. R.K. Jain & S.R.K. Iyenger, Advance Engineering Mathematics, Narosa Publication House, 2002.
4. Chandrika Prasad, Advanced Mathematics for Engineers, Prasad Mudralaya, 1996

Reference books

- Mathematical Structures for Computer Science: Judith Gersting, W.H. Freeman Press
- Concrete Mathematics, Ronald Graham, Donald Knuth, and Oren Patashnik, 2nd Edition - Pearson Education Publishers - 1996.
- Combinatorics: Topics, Techniques, Algorithms by Peter J. Cameron, Cambridge University Press, 1994 (reprinted 1996).

Learning Outcomes:

1. Have knowledge of the concepts needed to test the logic of a program.
2. Have an Understanding in identifying structures on many levels.
3. Be aware of a class of functions which transform a finite set into another finite set which relates to input output functions in computer science.
4. Be aware of the counting principles.
5. Be exposed to concepts and properties of algebraic structures such as semi-groups, monoids and Groups.

ELECTIVE:-

INTELLECTUAL PROPERTY RIGHTS

Course/Paper: 03BCS-106.1
BCS Semester III

Prerequisite : None

Course objectives

The subject aims to provide an overview of methods and approaches to manage intellectual property as strategic resources for enhancing the competitiveness for organizations. Upon completion of this subject, students should be able to accomplish the following objectives:

1. Understanding, defining and differentiating different types of intellectual properties (IPs) and their roles in contributing to organizational competitiveness
2. Understanding the Framework of Strategic Management of Intellectual Property (IP).
3. Appreciating and appraising different IP management (IPM) approaches and describing how pioneering firms initiate, implement and manage IPM programs,
4. Explaining how to derive value from IP and leverage its value in new product and service development
5. Exposing to the Legal management of IP and understanding of real life practice of IPM.

Unit	Content
I	Basic Concepts of Intellectual Property: Introduction to intellectual property rights, Intellectual property laws and the Internet, Trade Related Aspects of Intellectual Property Rights
II	Patents: Introduction to patent law and conditions for patentability, Procedure for obtaining patents, Rights of a patentee, Patent infringements, Biotechnology patents and patents on computer programs, Patents from an international perspective

III	Trademark and Geographical Indications: Statutory authorities and registration procedure, Rights conferred by registration, Licensing, assignment and transfer of trademark rights, Trademark infringement, Geographical Indication of Goods & Appellations of Origin
IV	Copyright: Registration procedure and copyright authorities, Assignment and transfer of copyright, Copyright infringement and exceptions to infringement, Software copyright
V	Designs: Introduction to the law on Industrial Designs, Registration and piracy, International perspective, Introduction to the law on semiconductor layout design, Registration, commercial exploitation and infringement

Text books:

1. Cheung, C.F., Wang, W.M., Tse, Y.L. and Ma Ricky "Knowledge-based Intellectual Property Management for Technology Development Industry", *Journal of Knowledge Management Practice*, Vol. 14, No. 2, http://www.tlainc.com/articl335.htm (2013).
2. Cheung, C.F., Wang, W.M., Xu, X. and Willoughby, Kelvin W. "A Knowledge-Based System for Assessing and Managing Intellectual Property Managerial Risks for Small-and-Medium Sized Technological Enterprises", *International Journal of Intellectual Property Management*, Vol. 7, No. 1/2, p.57-83 (2014).
3. Cornish, William Rodolph & Llewelyn, David. Intellectual property: patents, copyright, trade marks and allied rights. Sweet & Maxwell, 8/e, 2013.
4. Cornish, William Rodolph. Cases and materials on intellectual property. Sweet & Maxwell, 5/e, 2006.

Reference books:

- Gruner, Richard S., Ghosh,Shubha and Kesan, Jay P. Intellectual Property in Business Organizations: Cases and Materials 2006 http://www.lexisnexis.com/store/catalog/productdetail.jsp?pageName=relatedProducts&catId=&prodId =58918
- Sullivan, P.H., Value-Driven Intellectual Capital: How to Convert Intangible Corporate Assets into Market Value. John Wiley & Sons, Inc., Hoboken, New Jersey, 2000.
- Kieff, F. Scott, Newman, Pauline, Schwartz, Herbert F. and Smith, Henry E., Principles of Patent Law, 6th ed., Foundation Press, 2013.

Learning outcomes:

Upon completion of the subject, students will be able to:

- identify different types of Intellectual Properties (IPs), the right of ownership, scope of protection as well as the ways to create and to extract value from IP
- recognize the crucial role of IP in organizations of different industrial sectors for the purposes of product and technology development
- identify activities and constitute IP infringements and the remedies available to the IP owner and describe the precautious steps to be taken to prevent infringement of proprietary rights in products and technology development

INTERNET TECHNOLOGY

Course/Paper: 03BCS-106.2
BCS Semester III

Prerequisite: None

Course objectives
This subject aims to introduce the basic concepts and essential knowledge of the applications and technology of the Internet and World Wide Web. It provides a conceptual framework to understand the operation of the Internet and to understand how computers connect and communicate with each other. This subject also helps to develop students' analytical ability in network technology.

Unit	Contents
I	INTRODUCTION : Internet connection concepts- server, client and parts, Domain Name Systems, Telephone, cable and satellite connections- Dialup, ISDN, ADSL and leased line based connection, cable and DSS accounts,Web TV and Intranets, ISP features.
II	INTRANETS: What is Intranet? – Intranet Vs LANs Components of an Intranet Workstations and client software, Server and Network operating systems, Network Cards, Cabling and Hubs, Steps for creating an Intranet, Maintenance and connecting to Internet.
III	E-MAIL TECHNOLOGY: Features and Concepts- Message headers, Address book, Attachment, Filtering and forwarding mails.
IV	VIDEO CONFERENCING AND INTERNET TELEPHONY: Voice vs Video conferencing, Video conferencing hardware and features of video conferencing software, digital telephony as ISDN application, H 323 protocols and multi-point conferencing.
V	WEB TECHNOLOGY: Elements of the Web- Clients and servers, Languages and protocols Web page and Websites, special kinds of Web sites, Web Resources- Search Engines, Message boards, clubs, News groups and chat, Web page creation concepts- planning, Navigation, Themes and Publishing , Analyzing web traffic- Log file data, analyzing log files and products for analyzing web traffic

Text books:

1. Internet Technologies and Information Services (Library and Information Science Text Series) by Joseph B. Miller
2. The Internet Revolution: The Not-for-Dummies Guide to the History, Technology, and Use of the Internet by J. R. Okin
3. Understanding the Internet: A Clear Guide to Internet Technologies (Computer Weekly Professional Series) by Keith Sutherland

Reference books:

- Internet Technologies at Work (Mike Meyers' Computer Skills) by Fred T. Hofstetter
- Leonardo to the Internet: Technology and Culture from the Renaissance to the Present (Johns Hopkins Studies in the History of Technology) by Thomas J. Misa
- Media Technology and Society: A History: From the Telegraph to the Internet by Brian Winston

Learning outcomes:
Upon completion of the subject, students will be able to: (a) understand the basic concepts and applications of the Internet and World Wide Web. (b) apply relevant Internet knowledge to enhance their understanding of other networking situations. (c) use current Internet Technology necessary for daily life application

MANAGEMENT INFORMATION SYSTEM

Course/Paper: 03BCS-106.3
BCS Semester III

Prerequisite: None

Course objectives:
At the end of the course, it is expected that students are able to understand the usage of Information Systems in management. The students also would understand the activities that are undertaken in acquiring an Information System in an organization. Further the student would be aware of various Information System solutions like ERP, CRM, Data warehouses and the issues in successful implementation of these technology solutions in any organization.

Unit	Content
I	Introduction: MIS concept, Definition, role & Impact of MIS, Process of management, organization structure & behavior.
II	Basic of Management Information System: Decision Making, Information concepts, System concepts & control Types of system handling system complexity System development model.
III	Development of Management Information System: Requirement and implementation of MIS, Choice of information Technology for Management Information System.
IV	Application of Management Information system: Application in manufacturing sector using for personal management, Financial management, Production Management, Material Management, Marketing Management Application in Service Sector.
V	Enterprise Resource Planning (ERP): EMS, ERP, Benefits implementation, EMS & MIS. Case Studies: Application of SAP technologies in manufacturing sector.

Text books:

1. Kenneth, Laudon and Jane Laudon (2006). MIS: Managing the Digital Firm. Pearson Education.
2. James, A. O'Brien (2006). Introduction to Information Systems. Tata McGraw Hill.
3. Goyal, D.P. (2007). Management Information Systems, Macmillan India Ltd.
4. Turban, E., McLean, E. and Wetherbe, J. (2001). Information Technology for Management: Making Connections for Strategic Advantage. John Wiley and Sons.
5. Jawadekar, W. S. (2004). Management Information Systems. Tata McGraw Hill.

References books:

- Murdick R. G., Ross JE. & Claggett J.R. : Information system for Modern Management, 3rd Edn.,PHI, 1997.
- James A.O Brien: Management Information Systems, Galgotia Pubn., 1994.
- Wigarders K, Svensson A., Sehong L. : Structured Analysis & Design of Information Systems, Mc Graw-Hill book Co. 1986.
- Locus: Analysis, Design and Implementation of Information system, 3rd Edn., McGraw-Hili Book Co.
- Jawedker: Information System for Management.
- Anderson Lavid L., Post Gerald V. : Management Information System; Tat Mc Graw hilI, 3rd, 1999.

Learning Outcome:

The student would understand the classifications of MIS, understanding of functional MIS and the different functionalities of these MIS. This would be followed by case study on Knowledge management.

Laboratories:--

DIGITAL ELECTRONICS LAB

Course/Paper: 03BCS-201
BCS Semester III

Prerequisite :

1. Introductory computer programming in a high level language
2. Electric circuits (prerequisite)
3. Logic design (prerequisite)

Lab objectives:

1. Understand electrical conduction in solid state materials
2. Analyze and design dc and switching circuits containing diodes and transistors
3. Analyze and design combinational logic circuits at the transistor level
4. Develop skill with computer-based circuit simulation

Experiments:

1. Experimental study of characteristics of CMOS integrated circuits.
2. Interfacing of CMOS to TTL and CMOS.
3. Study of various combinatorial circuits based on: AND/NAND Logic blocks and OR/NOR Logic blocks.
4. Study of following combinational circuits: Multiplexer; Demultiplexer and Encoder. Verify truth tables of various logic functions.
5. To study various waveforms at different points of transistor bistable multi vibrator and its frequency variation with different parameters.
6. To study transistor a stable multi vibrator.
7. To design a frequency driver using IC-555/timer.
8. To study Schmitt trigger circuit.
9. To study OP-AMP as Current to voltage and voltage to current converter comparator.

10. BCD to binary conversion on digital/IC trainer.
11. Study various Flip flops and construct Parallel-in-Serial-out register. Testing of digital IC by automatic digital IC trainer.

Text books:

1. Albert Paul Malvino and Donald P. Leach, Digital Principles and Applications, (Fourth Edition) Tata Graw Hill Publishing Company Ltd, New Delhi.
2. Advanced microprocessors and peripherals – A. K. Ray & K. M. Bhurchandani, TMH, 2nd Edition 2006
3. Digital logic and state machine design – David. J Comer Oxford University Press
4. R.P. Jain, Modern Digital Electronic, Tata Mc Graw Hill Publishing Company Ltd. New Delhi.

Reference books:

- Salivahanan and S. Arivazhagan, Digital Circuits and Design, Vikas Publishing House Pvt. Ltd.
- Adel S. Se&a, and Kanneth C. Smith, Microelectronic Circuits, Oxford University Press
- Microprocessor and interfacing – N. Senthil Kumar, M. Saravanan S. Jeevananthan s. K.
- Shah Oxford University Press
- Digital Design. M. Morris Mano.

Learning Outcomes:

This course provides the foundation education in digital electronic circuit analysis and design. Through lecture, laboratory, and out-of-class assignments, students are provided learning experiences that enable them to:

- Design, simulate and implement basic combinational and sequential logic circuits.
- Become proficient with computer skills (eg., OrCAD Pspice, Logisim) for the analysis nd design of circuits.
- Develop technical writing skills important for effective communication.
- Acquire teamwork skills for working effectively in groups

ELECTRONICS LAB

Course/Paper: 03BCS-202
BCS Semester III

Pre requisite : Analog Circuit Analysis

Lab objectives:

This course provides substantial knowledge of the function pn junctions, pn- and zener diodes, bipolar and field-effect devices, basic analog and digital circuits. The course will discuss the device parameters of diodes, BJT's, field effect transistors, single stage amplifiers using common emitter, common base and common collector configurations as well as all configurations of FET amplifiers including CMOS and multi stage amplifiers, differential amplifier and output stages.

EXPERIMENTS

1. Study the following devices:
   (a) Analog & digital multi meters
   (b) Function/Signal generators
   (c) Regulated D.C. power supplies (constant, voltage and constant current operations).
2. Study of analog CRO measurement of time period, amplitude, frequency and phase angle using issajous figures.
3. Application of diode as clipper and clamper.
4. Plot V-I characteristic of zener diode & study zener diode as voltage, reverse Saturation current and static & dynamic resistances.
5. Plot V-I characteristic of zener diode & study zener diode as voltage regulator. Observe the effect of load changes and determine load limits of the voltage regulator.
6. Plot frequency response curve for audio amplifier and to determine gain bandwidth product.
7. Plot drain current-drain voltage and drain current-gate bias characteristics of field effect transistor and measurement of I & VP.
8. Plot gain: frequency characteristic of two stages RC coupled amplifier and calculate its bandwidth and compare it with theoretical value.
9. Plot gain: frequency characteristic of two stages RC coupled amplifier and calculate its bandwidth and compare it with theoretical value.
10. Plot input and output characteristics of BJT in CB, CC and CE configurations. Find their h-parameters.
11. Study half wave rectifier and effect of filter network on D.C. voltage output and ripple factor.
12. Study bridge rectifier and measure the effect of filter network on D.C. voltage output and ripple factor.

TEXT BOOKS

1. Johnson, DE, Johnson, JR and Hilburn, JL: Electric Circuit Analysis. Prentice Hall, 2nd edition
2. R Boylestad and L Nashelsky: Electronic Devices and Circuit Theory, Prentice Hall
3. Thomas L. Floyd: Digital Fundamentals, 8th Edition, Prentice Hall
4. David F. Hoeschele: Analog-to-Digital and Digital-to-Analog Conversion Techniques, 2nd Edition, John Wiley & Sons

References: Books

- J. Millman & C.C. Halkias: Integrated Electronics, Tata Mc-Graw Hill Publishing Ltd., New Delhi Mc-Graw Hill.
- Millman Grabel –Micro electronica, Mc-Graw Hill.
- Robert Boylestand & L.Nashelsky Electronic devices & circui theory.
- Sedra Smith- Microelectronic Circuits, Oxford Press, India.

Learning Outcome

Gain an intuitive understanding of the role of power flow and energy storage in electronic circuits; Develop the capability to analyze and design simple circuits containing non-linear elements such as transistors using the concepts of load lines, operating points and incremental analysis; Learn how the primitives of Boolean algebra are used to describe the

processing of binary signals and to use electronic components such as MOSFET's as building blocks in electronically implementing binary functions; Learn how the concept of noise margin is used to provide noise immunity in digital circuits;

DATA STRUCTURE LAB

Course/Paper: 03BCS-203
BCS Semester III

Prerequisites:

Students should have basic knowledge of C programming constructs and should be able to write basic C programs.

Lab objectives:

To provide a comprehensive working knowledge on the object oriented language C++ and to provide implementation experience on abstract data types, linear and nonlinear data structures for problem solving. To provide a working knowledge on generic programming based on over loading concepts, inheritance and virtuality. To inculcate ability to grasp the behaviour of data structures such as stacks, queues, trees, hash tables, search trees, graphs and their representation and to apply them in problem solving. To provide an application oriented working knowledge on searching and sorting techniques and to write programs to solve problems on arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.

Experiments:

1. Program on array searching, sorting (Bubble sort, Quick sort, Marge sort etc.)
2. Program to insert element at desire position, replacing element, deletion in array.
3. Various matrices operations.
4. Various strings programs.
5. Implementation of stack and queue using array
6. Implementation of stack and queue using link lists
7. Implementation of circular queue using link lists.
8. Polynomial addition, multiplication.
9. Two-way link lists programs.
10. Infix to postfix/prefix conversion.
11. BST implementation (addition, deletion, searching).
12. Graph traversal (BF

Text books:

1. Data structures a pseudo code approach with c++, indian edition, r.f.gilberg and b.a.forouzan cengage learning.
2. Programming prinicples and practice using c++, b.stroustrup, addition-wiesly (pearson education)
3. Data structures and stl,w.j.collins,mc graw hill,internationational edition.
4. Data structures and algorithms with oodesign patterns in c++,b.r.priess,john wiley &sons.

Learning outcomes:

1. Understanding of fundamental concepts of abstract data types and general standard data structures.

2. Ability to design linear data structures stacks, queues and linked lists.
3. Ability to design nonlinear data structures, trees and graphs, and to implement their operations.
4. Ability to implement different searching and sorting techniques.
5. Ability to apply different searching and sorting techniques for real world problems..

HUMANITIES AND SOCIAL SCIENCES

Course/Paper: 03BCS-204
BCS Semester III

Prerequisites: None

Lab objectives:
Designed for students to aware them with political affairs of country. . This Humanities Lab engages students in collaborative and experiential learning

Experiments:
1. Form of Government: Democracy, Dictatorship
2. India: Brief history of Indian Constitution, History of Indian National Movement, After Independence, Socio-economic growth.
3. Society: Social groups-concept and types socialization: concept and types, theory social control concept and types means. Social problem: concept and types.
4. The Fundamentals of Economics: The logic of economics fundamentals definitions of economics, basic terminology.
5. Micro Economics: Consumer's behavior, utility, demand, supply, elasticity of demand and supply. Theory of production, production function, factors of production.
6. Macro Economics: National income, business cycles, aggregate term, inflation, economic growth, international Trade, exchange rates.
7. Indian Economy: Basic features, infrastructure, occupation, natural and human resources, unemployment (Industrial Sector, India and Globalization).

Learning outcome :
After completion of this lab exercises student can learn the history of our Indian constitution, socialization and basic concept of economics

IV Semester

PRINCIPLES OF PROGRAMMING LANGUAGE

Course/Paper: 04BCS-101
BCS Semester IV

Prerequisite: Functional programming.

Course objectives:

- compare programming languages;
- describe the main principles of imperative, functional, object oriented and logic oriented programming languages;
- recite the high points of programming language history; and
- read the central formalisms used in the description of programming languages.
- assess programming languages critically and in a scientific manner;
- analyze the principles of an imperative, functional, object oriented or logic oriented programming language; and
- use a formalism to describe a programming language.

Unit	Content
I	Programming Language: Definition, History, Features. Issue in Language Design: Structure and Operation of computer, Language Paradigms. Efficiency, Regularity. Issues in Language Translation: Syntax, Semantics, Stages analysis and synthesis, Parse Tree, CFG and BNF grammar.
II	Specification and Implementation of Elementary and Structured Data Types. Type equivalence, checking and conversion. Array, List, Structure, Union.
III	Sequence control with Expressions, Conditional Statements, Loops, Exception handling. Subprogram definition and activation, simple and recursive subprogram, subprogram environment. Parameter passing mechanism.
IV	Abstract Data type, information hiding, encapsulation, type definition. Static and Stack-Based Storage management. Fixed and Variable size heap storage management. Garbage Collection
V	Parallel Programming: Introduction, parallel processing and programming language, Threads, semaphore, monitor, message passing.

Text Books:

1. Principles of Programming Languages by Dowek, Gilles
2. Principles of Programming Languages Version 1.0.1by Mike Grant & Zachary Palmer
3. A Text Book Of Principles Of Programming Languages 2nd Edition (English, Pradnya Kashikar, Nilesh Magar)

References:

- "A Formal Semantics of Programming Languages: An Introduction" by Glynn Winskel.
- "Foundations for Programming Languages" by John C. Mitchell
- "Types and Programming Languages" by Benjamin C. Pierce
- "Essentials of Programming Languages" by Daniel P. Friedman, Mitchell Wand and Christopher T. Haynes

Learning Outcomes:
After taking this course, students will be able to:

- define the semantics of a programming language using a definitional interpreter.
- investigate semantic issues in programming languages by studying implementations in an interpreter
- solve problems using a range of programming paradigms and assess the effectiveness of each paradigm for a particular problem.

MICROPROCESSOR AND INTERFACES

Course/Paper: 04BCS-102
BCS Semester IV

Prerequisites: The students should have good background on digital circuits (should have attended the course Switching Circuits and Logic design

Course Objectives:
The students should have good background on digital circuits (should have attended the course Switching circuits and Logic Design

Unit	Content
I	Introduction to Micro Computer Systems: Microprocessors, microcontroller and microcomputer devices, Machine and assembly language, Bus concept. Architecture & Pinout of 8085A.
II	Assembly Language and Programming in 8085: Instruction set, Program structures (sequential, conditional, iterative), Macros and subroutines, Stack, Counter and timing delay, interrupt structure and its programming.
III	Peripherals and their interfacing with 8085-I: Memory Interfacing, Interfacing I/O ports, Data transfer schemes (Synchronous, asynchronous, interrupt driven), Architecture & interfacing of PPI 8255, Data Converters and Timer 8254.
IV	Peripherals and their interfacing with 8085-II: Architecture & interfacing of- DMA controller 8257, interrupt Controller 8259A, USART 8251, Level Converters MC 1488 and MC 1489, Current loop, RS 232 C and RS 422 A.
V	Comparative study of 8085 A, 8086 and 8088 (Pinout, internal architecture, timing diagrams), Instruction format and addressing modes – Data and Branch related. Features of Pentium processor, MMX and Dual core processor.

Text Books:

1. Ramesh S.Gaonkar, "Microprocessor - Architecture, Programming and Applications with the 8085", Penram International publishing private limited, 2002.
2. The 8051 Microcontroller Architecture, Programming & applications II Edition Kenneth J Ayala ,2005.

References:

- Gaonkar: Microprocessor and its Applications.
- Liu,Y,Gibson, G.A. : Microcomputer systems: The 8086/8088 family, Prentice-Hall, 2nd Edn. 1986.

- Parson, AJ. : Microprocessors: Essential, Components and System, Galogtla Publ. Pvt. Ltd.
- INTEL-Microcontroller handbook.
- Ayle- 8051 Micro controller, penram press.

Course Outcome:

1. The student will learn the internal organization of some popular microprocessors/microcontrollers.
2. The student will learn hardware and software interaction and integration.
3. The students will learn the design of microprocessors/microcontrollers-based systems.

OBJECT ORIENTED PROGRAMMING

Course/Paper: 04BCS-103
BCS Semester IV

Prerequisites:

This topic is only suitable for those with a solid programming background who wish to learn more about Object Oriented Programming and modular program design. If you decide to take this option, please note that you should undertake preparatory reading before taking the course; see the reading list for details. People without a solid programming background should not take this course.

This is NOT a course about Java: students are expected to be already familiar with Java, the basic concepts of object orientation such as classes, interfaces, members, and methods, and the principles of imperative programming languages. Also, this is NOT a course about the usage of Java libraries: students are expected to be able to learn about various API calls available in Java's Software Developer Kit by themselves.

Course Objectives:

This course aims to prepare undergraduates for the programming work they will undertake during their time in Oxford and subsequently, including subsequent programming-heavy courses such as Compilers and Database Systems Implementation, in addition to the group project and later individual project work. The course contains few topics that have not been mentioned in previous courses, but the defining aim in this course is to illustrate those programming techniques put to work in a sequence of case studies of carefully chosen size, each of them big enough to have significant internal interfaces, but not so large as to be overwhelming. The course will introduce standard tools and techniques for software development: use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests, and profiling tools for studying performance. Participants will be able to choose between an IDE and a traditional editor/compiler setup.

Unit	Content
I	Introduction to programming paradigm. Aspect-oriented programming, Dynamic programming, Functional programming, Logic programming, Object-oriented programming, Parallel computing, Event Driven Programming.

II	Overview of C++ (A): Abstraction, Polymorphism, Inheritance, Classes, Objects, Methods. Constructor, destructor
III	Overview of C++ (B) :Overloading (function and operator), references, friend function, overriding, virtual function, virtual classes, templates, Namespace, Nested and inner classes, Exception handling, Run time type casting, STL (List, Map, Algorithm)
IV	Overview of Java(A) : Java Byte code and virtual machine, data types, operators, arrays, Objects, constructors, returning and passing objects as parameter, Single and Multilevel inheritance, Extended Classes, Access Control, Usage of super, overloading and overriding methods, Abstract classes, Using final with inheritance.
V	Overview of Java (B): Package and interfaces, String Handling, String constructors, special string operations, character extraction searching and comparing strings, string Buffer class. Applet Fundamentals, Using paint method and drawing polygons.

Text Books:

1. Elegant Objects (Volume 2) by Yegor Bugayenko
2. Design Patterns: Elements of Reusable Object-Oriented Software (Hardcover) by Erich Gamma
3. Growing Object-Oriented Software, Guided by Tests (Paperback) by Steve Freeman
4. Practical Object Oriented Design in Ruby (Paperback) by Sandi Metz

References:

- E Balaguruswamy: Object Oriented Programming with c++; Tata Mc Graw Hill.
- Margaret Ellis; Bjarne Strousstrup : The annotated c++ reference manual.
- Chirlian PM. : Programming inc++; Merril pub. Co.,1990.
- James Rumbaugh etal, “Object Oriented Modeling and Design”, PHI
- Herbert Schieldt, “The Complete Reference: Java”, TMH.

Learning Outcome:

By attending this course, students satisfying the prerequisites are expected to

- understand the fundamental principles of object-oriented design,
- be able to apply the object-oriented design principles such as separation of concerns, responsibility analysis, and design by contract, and
- be conversant with the most important design patterns.

SYSTEM SOFTWARE

Course/Paper: 04BCS-104
BCS Semester IV

Prerequisites:
Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Course Objectives:
To view some of the major tasks of the system software of a computer system, focusing on internal working of the hardware and software interface of a typical system.

Unit	Content
I	Overview: Comparison of machine language, assembly language and high level languages, External and internal representation of instructions and data. Data allocation structures, search structures and addressing modes. Activities and system software for program generation, translation and execution. Editors for source code and object code/executable code files.
II	Assemblers: Assembly language specification. Machine dependent and independent features of assembler. Classification of assemblers. Pass structure of assemblers (problem and associated for IBM-PC.
III	Loader and Linkers: Functions and classification. Machine dependent and independent features of loaders, Design of bootstrap, absolute and relocatable loaders, Design of linker. Case study of MS-DOS linker.
IV	Macro processors: Macro definition, call and expansion. Macro processor algorithm and data structure. Machine independent features (parameters, unique labels, conditional expansion, nesting and recursion).Pass structure and design of microprocessor and macro assembler, Case study of MASM macro processor.
V	High level language processor: HLL specification: Grammars and parse trees, expression and precedence. Lexical analysis: Classification of tokens, scanning methods, character recognition, lexical ambiguity. Syntactic analysis: Operator precedence parsing, recursive descent parsing. Symbol Table Management: Data structure for symbol table, basing functions for symbols, overflow technique, block structure in symbol table.

Text Books:
1. Basic Computing Using Windows
2. Bourne Shell Scripting
3. Creating a Simple 3D Game with XNA
4. GENtle
5. Knowing Knoppix

References:

- Andrew P. Sage and James D. Palmes: system software engineering
- Leland L. Beck: an intro to system programming 3rd edition by hardcover
- Nick Rozanski: software system architecture by kindle
- Neetu Sharma and Amit Sharma : system software engineering
- Vikas Thadda : system software engineering

Learning Outcomes:
The graduate has reliably demonstrated the ability to: Analyze and resolve information technology problems through the application of systematic approaches and diagnostic tools. Analyze, plan, design, and implement computer systems.

STATISTICS AND PROBABILITY THEORY

Course/Paper: 04BCS-105
BCS Semester IV

Prerequisites:
- The basics of classic probability require, for the most part, ability to do arithmetic and/or access to a good arithmetic calculator — I find that the Python interactive development environment, IDLE, is quite sufficient.
- Beyond that, you also need to be able to analyze how events depend on each other, as well as understanding the use of numbers logically and systematically… and for that reason, some basic algebra is extremely helpful.
- Finally, it also helps greatly to be able to apply the Binomial Theorem, which, again, is much easier to understand and apply if you have some background in algebra.

Course Objectives:
By the end of the course you will be able to do the following:
Mathematical:
- to create, simulate, and analyze elementary probabililty models and
- to explain the limitations of the statistical inferences made therefrom.

Epistemological:
- to better assess your understanding of mathematical concepts and their importance.
- to write math in a clear, concise way that emphasizes what's important

Unit	Content
I	Introduction: Sample space, Events, Algebra of events, Bayes' Rule, Bernoulli Trials.Probability Distribution and Probability Densities: Bernoulli, Binomial, Poisson, Normal, rectangular and exponential distributions and their PDFs. Moments and MGFs for above distributions.
II	Discrete Random Variables: Random Variables and their event space, probability mass function. Distribution Functions. Probability Generating Function. Expectations: Moments, Computation of mean Time to failure. Bernoulli & Poisson Processes.
III	Queuing Theory: Pure birth, Pure Death and Birth-Death Processes, mathematical Models for M/M/I, M/M/N, M/M/S and M/M/S/N/ queues.
IV	Discrete Parameter Markov Chains: M/G/I Queuing Model, Discrete Parameter Birth-Death Process.
V	Network of queues: Open Queuing Networks. Correlation & Regression: Linear regression, Method of least squares, Normal regression and correlation Analysis.

Text Books:

1. Probability and Mathematical Statisticsby Prasanna Sahoo - University of Louisville , 2013
2. Introduction to Probability, Statistics, and Random Processes by Hossein Pishro-Nik - Kappa Research, LLC , 2014
3. Advanced Data Analysis from an Elementary Point of View by Cosma Rohilla Shalizi - Cambridge University Press , 2013

References:

- Dr. Jain, Dr. Mangal Maheshwari, Mr. Praveen Deora "Statistic and Probability Theory". Dhanpat Rai .
- Jain and Rawat "SPT", CBC
- Y.N. Gaur "Statistic and Probability Theory", Genius Pub.

Learning Outcomes:

1. Students who successfully complete this course should be able to demonstrate understanding of :
2. basic probability axioms and rules and the moments of discrete and continous random variables as well as be familiar with common named discrete and continous random variables.
3. how to derive the probability density function of transformations of random variables and use these techniques to generate data from various distributions.
4. how to calculate probabilities, and derive the marginal and conditional distributions of bivariate random variables.
5. discrete time Markov chains and methods of finding the equilibrium probability distributions.
6. how to calculate probabilities of absorption and expected hitting times for discrete time Markov chains with
7. how to translate real-world problems into probability models.
8. how to read and annotate an outline of a proof and be able to write a logical proof of a statement.

ELECTIVE:--

OPEN SOURCE TECHNOLOGY

Course/Paper: 04BCS-106.1
BCS Semester IV

Prerequisites:

Current standing as a junior, senior, or graduate student in computer science. This class is not for the faint of heart. It will be difficult and demand a lot of time, but you will also learn a lot in this class. My hope is that this class prepares you to enter the work place and provides a meaningful software engineering experience. You should expect 10+ hours of homework each week and exposure to many technologies and software development paradigms that you may not be familiar with. You should be comfortable learning new things, and this class should give you experience learning new things.

Course Objectives:

The objectives of this course are to introduce students to open source software. Students will study common open source software licenses, open source project structure, distributed team software development, and current events in the open source world. Students will also work on an open source project and will be expected to make a significant contribution.

Unit	Content
I	OST overview: Evolution & development of OST and contemporary technologies, Factors leading to its growth. Open Source Initiative (OSI), Free Software Foundation and the GNU Project, principle and methodologies. Contexts OST (India & international). Applications of open source (open source teaching and open source media) Risk Factors. Myths regarding open source.
II	Philosophy of Software Freedom, Free Software, OSS, Closed software, Public Domain Software, Shared software, Shared source. Detail of few OSS like Open Audio, Video, 2d & 3d graphics software, system tools, office tools, Networking & internet, Security, Educational tools and Games.
III	Open Source Development Model, Starting and Maintaining an Open Source Project, Open Source Hardware, Open Source Design, Ongoing OS Projects (i.e. examples of few good upcoming software projects.) Case Study: - Linux, Wikipedia.
IV	Licenses and Patents: What Is A License, How to create your own Licenses? Important FOSS Licenses (Apache, BSD, GPL, LGPL), copyrights and copy lefts, Patents
V	Social and Financial impacts of open source technology, Economics of FOSS: Zero Marginal Cost, Income generation opportunities, Problems with traditional commercial software, Internationalization, Open Source as a Business Strategy.

References:

1. Open Source: Technology and Policy by Fadi P. Deek and James A. M. McHugh
2. Open Source Approaches in Spatial Data Handling (Advances in Geographic Information Science) by G. Brent Hall and Michael G. Leahy
3. Open Sources 2 by Chris Dibona, Danese Cooper, and Mark Stone
4. Understanding Open Source and Free Software Licensing by Andrew M. St. Laurent
5. Agile Technologies in Open Source Development (Premier Reference Source) by Barbara Russo, Marco Scotto, Alberto Sillitti, and Giancarlo Succi

Course Outcome:

After completing this course, the student should be able to

- explain common open source licenses and the impact of choosing a license
- explain open source project structure and how to successfully setup a project
- be competent with distributed software engineering tools and processes such as test-driven development, issues tracking, unit testing, code review, distributed version control, and continuous integration

E- COMMERCE

Course/Paper: 04BCS-106.2
BCS Semester IV

Prerequisites:
At present, E-Commerce is the most vast and popular business sector. Its an online buying and selling process so its needs are as simple as the business means. Customer reaching, High sales, Products quality and on time delivery are some of the needs of any E-commerce business. And when we talk about its objectives so basically its objectives are for fulfilling its needs.

Course Objectives:

- **High reachability** - The main objective and at the same time need is traction on your web store. Of,course if you are selling products online what you require are customers. If you are getting good reachability then your business will definitely grow. Therefore one of the objective is high reachability.
- **High Conversions**- if people are coming on your web store and purchasing something then it will calculate as conversions and from the number of people who are buying stuff from your web store we can calculate the conversion rate.
- **Customer satisfaction** - Customer is the main part of any E-commerce business so its very important to make your customer happy and satisfied. By providing quality and desirable products, on time delivery, 24*7 customer support, and timely sale & best deal offers you can make your customer happy. It is one of the main objectives of E-commerce.
- **Social popularity** - Unless and until you are not famous and popular among people you cannot establish your brand. social presence with Omni channel & Digital marketing is essential for any E-commerce business.

Unit	Content
I	Business Strategy in an Electronic Age: Value Chain-supply chains, Proter's value chain, model and Inter- Organizational value chains. Competitive Advantage-Competitive strategy, Proter's Model, First Mover advantage and competitive advantage using e-commerce Business strategy Introduction to Business Strategy, Strategic Implications of IT technology e-commerce Implementation and evaluation.
II	Business to Business Electronic Commerce: Inter-organizational Transactions, The credit Transaction Trade cycle. A variety of transactions, Electronic markets-markets and electronic markets, usage of electronic markets, Advantages and disadvantages of electronic markets.
III	Electronic Data Interchange (EDI): Definition and benefits of EDI. EDI technology, standards, communications, implementation, agreements and securities. EDI trading patterns and transactions.
IV	Building an E-Commerce Site: Introduction to object behavior, components, active scripting. Object models, Infrastructure objects, service object and data objects, choosing the objects. Building a scalable application, Addition the configure method, connecting to the database, Accessing and versioning the database. Building the catalog object with example. Creating shopping basket-Holding state, creating the tables for a shopping basket, modifying the object model and making the basket

	accessible.
V	J2EE Architecture Overview: Enterprise components, Information technology in the enterprises, introduction to enterprise objects and enterprise component model. The J2EE model features, J2EE components-container architecture. Enterprises Java and J2EE architecture.

References:

- R David Whitely; Electronic Commerce Strategy, technology, Applications Tate McGraw Hill
- Soka: From EDI to E-Commerce; McGraw Hill, 1995
- Harley Hahn-The Internet Complete Reference Tate McGraw Hill
- Concept of E-Commerce, Adash K.Pandey, S.K. Kataria & Sons
- E-Commerce, J.S. Pilaniya, Genius Publications

Learning Outcomes:

Students will be able to:

- Design and implement an e-commerce application with a shopping cart.
- Integrate the waterfall model in the development of e-commerce applications.
- Integrate user-centered design guidelines in developing user-friendly websites.
- Evaluate the bullwhip effect in a supply chain, analyze the causes, and recommend possible solutions.
- Analyze different types of portal technologies and deployment methodologies commonly used in the industry.
- Analyze the effectiveness of network computing and cloud computing policies in a multi-location organization.
- Analyze real business cases regarding their e-business strategies and transformation processes and choices.
- Integrate theoretical frameworks with business strategies.

ANALOG AND DIGITAL COMMUNICATION

Course/Paper: 04BCS-106.3
BCS Semester IV

Prerequisites: None

Course Objectives:

- To understand the building blocks of digital communication system.
- To prepare mathematical background for communication signal analysis.
- Tounderstand and analyze the signal flow in a digital communication system.
- To analyze error performance of a digital communication system in presence of noise and other interferences.
- To understand concept of spread spectrum communication system

Unit	Content
I	Modulation of Signals: Principles of Analog modulation technique like AM, FM, PM, SSB, Generation & detection. Frequency division multiplexer. Pulse modulation: Pulse transmission over band-limited signals, sampling theory, pulse amplitude modulation.
II	Digital Communication: PCM, DCSM, DM, ADM, comparison of above systems on the basis of performance criteria such as bit transmission, signaling rate, error probability, S/N ratio, bandwidth requirement. ISI & Eye diagram.
III	Digital Modulation technique: Data transmission using techniques such as PSK, FSK, QFSK (QAM), MSK Inter system comparison.
IV	Introduction to communication channel: Transmission line-primary and secondary line constant, telephone lines and cable, public switch telephone network (Electronic exchange). Introduction of fiber optic communication- Principle of light communication in fiber, losses in fiber, dispersion, light source and photo detector, connector and splicer.
V	Introduction to coding technique: Information theory, channel capacity, Shannon's theorem, source coding, error control coding, error detection and correction, block code, cycle code, line code channel throughput and efficiency.

References

- H. TAFF & D.L SCHILLING- Principle of Communication System, TMH
- G. Kennedy- Electronic Communication System, TMH
- B.P. LATHI- Communication System, John Wiles
- Sanjay Sharma- Analog and Digital Communication

Course Outcomes:

After successfully completing the course students will be able to

– Analyze the performance of a baseband and pass band digital communication system in terms of error rate and spectral efficiency.
– Perform the time and frequency domain analysis of the signals in a digital communication system.
– Select the blocks in a design of digital communication system.
– Analyze Performance of spread spectrum communication system.

Laboratories:--

COMMUNICATION LAB

Course/Paper: 04BCS-201
BCS Semester IV

Prerequisites: None

Course Objectives:

This course has experiments provides the foundation education in communication engineering lab analysis and design. Through lecture, laboratory, and out-of-class assignments, students are provided learning experiences that enable them to Analyze and deign basic electronic circuits, to carry out AM and FM modulation experiment s using

discrete electronic components and Become proficient with computer skills (eg., OrCAD Pspice and MATLAB) for the analysis and design of circuits .

Experiments:

1. Harmonic analysis of a square wave of modulated waveform
2. Observe the amplitude modulated waveform and measures modulation index. Demodulation of the AM signal
3. To modulate a high frequency carrier with sinusoidal signal to obtain FM signal. Demodulation of the FM signal
4. To observe the following in a transmission line demonstrator kit :
   i. The propagation of pulse in non-reflecting Transmission line.
   ii. The effect of losses in Transmission line.
   iii. The resonance characteristics of al half wavelength long x-mission line.
5. To study and observe the operation of a super heterodyne receiver
6. To modulate a pulse carrier with sinusoidal signal to obtain PWM signal and demodulate it.
7. To modulate a pulse carrier with sinusoidal signal to obtain PPM signal and demodulate it.
8. To observe pulse amplitude modulated waveform and its demodulation.
9. To observe the operation of a PCM encoder and decoder .To considers reason for using digital signal x-missions of analog signals.
10. Produce ASK signals, with and without carrier suppression, Examine the different processes required for demodulation in the two cases.
11. To observe the FSK wave forms and demodulate the FSK signals based on the properties of (a) tuned circuits (b) on PI.L.

Learning Outcome:

On successful completion of the course students will be able to:

- Understand basic elements of a communication system
- Conduct analysis of baseband signals in time domain and in frequency domain
- Demonstrate understanding of various analog and digital modulation and demodulation techniques
- Analyse the performance of modulation and demodulation techniques in various transmission environments
- Appreciate the importance of synchronisation in communication systems

MICROPROCESSORS LAB

Course/Paper: 04BCS-202
BCS Semester IV

Prerequisites:

The students should have good background on digital circuits (should have attended the course Switching Circuits and Logic Design)

Course Objectives:

Microprocessors: Historical background; Organization & Architectural Features of Microprocessor & Micro Controllers; The Instruction Set: Instruction format, addressing

modes; Assembly language programming of 8085 and 8051; Interfacing of memory devices; Data transfer techniques and I/O ports; Interfacing of keyboard and display devices; Programmable Interrupt and DMA controllers; Interfacing of sensors, transducers, actuators, A/D & D/A Converters, Analog Signal Conditioning Circuits, Data acquisition systems; Standard Interfaces – RS232, USB; Development aids and troubleshooting techniques; Application examples; Advanced microprocessors and microcontrollers.

Course Objectives:

1. Study of hardware, functions, memory, and operations of 8085 kit.
2. Program to perform integer addition (two and three numbers 8 bit)
3. Program to perform multiplication (two 8 bit numbers).
4. Program to perform division (two 8 bit numbers).
5. Transfer of a block data in memory to another place in memory in forward and reverse order.
6. Swapping of two block data in memory.
7. Addition of 10 numbers using array.
8. Searching a number in an array.
9. Sorting of array (ascending, descending order).
10. Print Fibonacci sequence. (15 elements)
11. To insert a number at correct place in a sorted array.
12. Interfacing seven segment display using 8255.

Text Books:

1. Gaonkar R., Microprocessor Architecture, Programming, and Applications with the 8085, Penram
2. Pal A., Microprocessors: Principles and Applications, TMH

Reference Books:

- Ayala K. J., The 8051 MicrocontrollerArchitecture, Programming & Applications, Penram
- Mazidi and Mazidi, Microcontroller and Embedded Systems, Pearson Education
- Kapadia, R., 8051 Microcontroller and Embedded Systems, Jai

Learning Outcome:

1. Provide practical hands-on experience with microprocessor applications and interfacing techniques.
2. Understand 8085 microprocessor kit, knowledge of 8085 instruction set and ability to utilize it in assembly language programming.
3. Understand real mode Memory addressing and ability to interface various devices to the microprocessor.

OBJECT ORIENTED PROGRAMMING LAB

Course/Paper: 04BCS-203
BCS Semester IV

Prerequisites:
This topic is only suitable for those with a solid programming background who wish to learn more about Object Oriented Programming and modular program design. If you decide to take this option, please note that you should undertake preparatory reading before taking the course; see the reading list for details. People without a solid programming background should not take this course.
This is NOT a course about Java: students are expected to be already familiar with Java, the basic concepts of object orientation such as classes, interfaces, members, and methods, and the principles of imperative programming languages. Also, this is NOT a course about the usage of Java libraries: students are expected to be able to learn about various API calls available in Java's Software Developer Kit by themselves.

Course Objectives:
This course aims to prepare undergraduates for the programming work they will undertake during their time in Oxford and subsequently, including subsequent programming-heavy courses such as Compilers and Database Systems Implementation, in addition to the group project and later individual project work. The course contains few topics that have not been mentioned in previous courses, but the defining aim in this course is to illustrate those programming techniques put to work in a sequence of case studies of carefully chosen size, each of them big enough to have significant internal interfaces, but not so large as to be overwhelming. The course will introduce standard tools and techniques for software development: use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests, and profiling tools for studying performance. Participants will be able to choose between an IDE and a traditional editor/compiler setup.

Experiments:
C++ Programs
1. Programs based on inheritance property.
2. Programs of operator overloading (complex number arithmetic, polar coordinates).
3. Programs using friend functions.
4. Programs on various matrix operations.
5. Stack operations using OOPs concepts.
6. To implement Tower of Hanoi problem.

JAVA Programs:-
7. To implement spell checker using dictionary.
8. To implement color selector from a given set of colors.
9. To implement shape selector from a given set of shapes.
10. To implement a calculator with its functionality.
11. To show movement of a car.

Text Books:
5. Elegant Objects (Volume 2) by Yegor Bugayenko
6. Design Patterns: Elements of Reusable Object-Oriented Software (Hardcover) by Erich Gamma

7. Growing Object-Oriented Software, Guided by Tests (Paperback) by Steve Freeman
8. Practical Object Oriented Design in Ruby (Paperback) by Sandi Metz

References:

- E Balaguruswamy: Object Oriented Programming with c++; Tata Mc Graw Hill.
- Margaret Ellis; Bjarne Strousstrup : The annotated c++ reference manual.
- Chirlian PM. : Programming inc++; Merril pub. Co.,1990.
- James Rumbaugh etal, "Object Oriented Modeling and Design", PHI
- Herbert Schieldt, "The Complete Reference: Java", TMH.

Learning Outcome:

By attending this course, students satisfying the prerequisites are expected to

- understand the fundamental principles of object-oriented design,
- be able to apply the object-oriented design principles such as separation of concerns, responsibility analysis, and design by contract, and
- be conversant with the most important design patterns.

SYSTEM SOFTWARE LAB

Course/Paper: 04BCS-204
BCS Semester IV

Prerequisites:

There are no pre-requisites other than the fact that you are a CS major. You should have done CS 101. This course is a second year CS core course. It is not open to students from other departments. Audit students are not allowed.

Course Objectives:

To develop skills to construct reliable software of high quality that is reasonably easy to understand, modify and maintain

Experiments:

In this lab we will practice how source code is processed by compiler/ assembler/ pre-processor.

All programs have to be written in C++

1. Write a class for file handling, having functions to open/ read/ write/ close/ reset.

(2-5) Develop a program which take input a file of C language

a. Print Lines of Codes and print signature of all function (including main)
b. Print number of variables in every function (with type)
c. Generate a new file without the comments. (/* */ and //)
d. Process all #define (i.e. #define MAX 100, than replace every occurrence of MAX with 100). (Macro value 100 can be an expression also.)

6. Write a program to create a symbol table.
7. Write a program which can parse a given C file and store all variables and functions in symbol table.

(8-10). Write a program to convert given C program into RTL code.

Assumption

a. input C file will have only main function,

b. only two type of statements, either variable declaration statements (int sub1=23;) OR mathematical expression (sub1=sub2-sub3;).
c. system have 16 registers (R1 to R16)
d. RTL opcode available are: ADD, LOAD, MOVE, SUB, MULTIPLY, DIVIDE
e. No control-flow (i.e. if-else, loop, jump etc.) expression is there in input code e.g.

```
int main()
{
int sub1=72, sub2=85, sub3=63;
float per;
per=(sub1+sub2+sub3)/(100+100+100);
}
```

Learning Outcome:

- Learn the software life cycle phases (project management, requirements engineering, software design, prototyping and testing). Create and specify a software design based on the requirement specification that the software can be implemented based on the design.
- Get familiar with UML (modeling language for analysis and design). Make a testing plan for the software.

V Semester

SOFTWARE ENGINEERING

Course/Paper: 05BCS-101
BCS Semester V

Prerequisites:
Analytical, problem-solving, teamwork and communication skills as well as creativity and attention to detail; familiarity with database management, development environment, program testing and other related software

Course Objectives:
This course introduces the concepts and methods required for the construction of large software intensive systems. It aims to develop a broad understanding of the discipline of software engineering. It aims to set these techniques in an appropriate engineering and management context.

Unit	Content
I	System Analysis: Characteristics, Problems in system Development, System Level project Planning, System Development Life cycle (SDLC), computer system engineering system analysis, modeling the architecture, system specification.
II	Software Project Management: Objectives, Resources and their estimation, LOC and FP estimation, effort estimation, COCOMO estimation model, risk analysis, software project scheduling. Software Development : Life Cycle (SWDLC), SWDLC models software engineering approaches
III	Requirement Analysis: Requirement analysis tasks, Analysis principles. Software prototyping and specification data dictionary finite state machine (FSM) models. Structured Analysis: Data and control flow diagrams, control and process specification behavioral modeling, extension for data intensive applications.
IV	Software Design: Design fundamentals, Effective modular design: Data architectural and procedural design, design documentation.
V	Object Oriented Analysis: Object oriented Analysis Modeling, Data modeling. Object Oriented Design: OOD concepts and methods class and object definitions, refining operations. Class and object relationships, object modularization. Introduction to Unified Modeling Language

Textbooks:
1. Shooman, M. : Software Engineering, McGraw-Hill
2. Shere: Software Engineering & Management, Prectice-Hall.

References:
- Pressman Roger: Software Engineering -A Practitioner's Approach; Tata McGraw Hill, N.Delhi,-1991
- Jalote Pankaj: An Integrated Approach to Software Engineering; Narosa, New Delhi, 1991.
- Fairley, R.E. : Software Engineering Concepts, McGraw-Hill,

Learning Outcomes:
Students will be able to: Model the structure and behavior a *software* system the UML class diagrams and state diagrams. Design a solution to a given problem using one or more design patterns and implement the design in a programming language

COMPUTER ARCHITECTURE

Course/Paper: 05BCS-102
BCS Semester V

Prerequisites:
- Understanding of basic computer organization, including familiarity with such components as CPU, ALU, multiplexors, registers, main memory, caches, and buses
- familiarity with the roles of compilers, assemblers, and operating systems
- some familiarity with assembly language
- ability to understand simple C programs and to run programs in a UNIX environment, and
- familiarity with the representation of numbers in digital computers

Course Objectives:
This course presents key principles underlying the design of modern digital computers. The course introduces quantitative techniques used to guide the design process. It describes CPU performance issues and introduces instruction set architectures. The course then uses a hypothetical computer design, with a simple RISC architecture, to show how modern digital computers are implemented, first using a simple non-pipelined implementation, followed by a higher-performance pipelined implementation. The major hazards introduced by pipelining, including structural hazards, data hazards, and control hazards are discussed and techniques for overcoming them are described. Additional topics covered in this course include the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

Unit	Content
I	REGISTER TRANSFER LANGUAGE: Data movement around registers. Data movement from/to memory, arithmetic and logic micro operations. Concept of bus and timing in register transfer.
II	CPU ORGANISATION: Addressing Modes, Instruction Format. CPU organization with large registers, stacks and handling of interrupts & subroutines Instruction pipelining
III	ARITHMETIC ALGORITHM: Array multiplier, Booth's algorithm. Addition subtraction for signed unsigned numbers and 2's complement numbers.
IV	MICROPROGRAMMED CONTROL UNIT : Basic organization of micro-programmed controller, Horizontal & Vertical formats, Address sequencer
V	MEMORY ORGANISATION: Concept of RAM/ROM, basic cell of RAM, Associative memory, Cache memory organization, Vertical memory organization. I/O ORGANISATION: Introduction to Peripherals & their interfacing. Strobe based and handshake-based communication, DMA based data transfer, I/O processor.

Textbooks:

1. D.Sima, T. Fountain, P. Kacsuk. "Advance computer Architectures: A Design Space Approach", Addision Wesley, 1997.
2. M.J. Flynn, "Computer Architecture: Pipelined and Parallel Processor Design", Narosa Publishing House/Jones

References:

- M. Mano, "Computer System Architecture", PHI India Pvt. Ltd.
- Tannenbaum, " Structured Computer Organization", PHI India Pvt. Ltd.
- Stallings, "Computer Organization", PHI India Pvt. Ltd.

Learning Outcomes:

Students gain an understanding of the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

DATABASE MANAGEMENT SYSTEMS

Course/Paper: 05BCS-103
BCS Semester V

Prerequisites:

You are required to have the background from a data structures course (e.g., lists, hash tables, arrays, search trees) and strong programming experience. Programming skills: The projects are in Java and C++. The knowledge of C++ is assumed. You will be given a brief overview of Java and expected to be able to pick up the language on you own, if you haven't used it before.

Course Objective:

- Knowledge of DBMS, both in terms of use and implementation/design
- Experience with SQL
- Increased proficiency with the programming language C++
- Experience working as part of team
- Experience with analysis and design of (DB) software

Unit	Content
I	INTRODUCTION TO DATABASE SYSTEMS: Overview and History of DBMS. File System vs DBMS .Advantage of DBMS Describing and Storing Data in a DBMS. Queries in DBMS. Transaction management and Structure of a DBMS.
II	ENTITY RELATIONSHIP MODEL: Overview of Data Design Entities, Attributes and Entity Sets, Relationship and Relationship Sets. Features of the ER Model-Key Constraints, Participation Constraints, Weak Entities, Class Hierarchies, Aggregation, Conceptual Data Base, Design with ER Model-Entity vs Attribute, Entity vs Relationship Binary vs Ternary Relationship and Aggregation vs ternary Relationship Conceptual Design for a Large Enterprise.

III	RELATIONSHIP ALGEBRA AND CALCULUS**:** Relationship Algebra Selection and Projection, Set Operations, Renaming, Joints, Division, Relation Calculus, Expressive Power of Algebra and Calculus.
IV	SQL QUERIES PROGRAMMING AND TRIGGERS: The Forms of a Basic SQL Query, Union, Intersection and Except, Nested Queries ,Correlated Nested Queries, Set-Comparison Operations, Aggregate Operators, Null Values and Embedded SQL, Dynamic SQL, ODBC and JDBC, Triggers and Active Databases.
V	SCHEMA REFINEMENT AND NORMAL FORMS: Introductions to Schema Refinement, Functional Dependencies, Boyce-Codd Normal Forms, Third Normal Form, Normalization-Decomposition into BCNF Decomposition into 3-NF.

Text Books:
Database Management Systems , Ramakrishnan & Gehrke

References:
- Date C J, "An Introduction To Database System", Addision Wesley
- Korth, Silbertz, Sudarshan, "Database Concepts", Mc Graw Hill
- Bipin C. Desai, "An introduction to Database Systems", Galgotia
- F. H. Lochousky, DC Tsichritzis"DBMS" NewYork Academic Press
- N. Goodman, V. Hadzilacos "Concurrency Control and Recovery in Data Base System" Addison Wesley.

Learning Outcomes:
- Understand database concepts and structures.
- Explain terms related to database design and management.
- Understand the objectives of data and information management.
- Understand data modeling and database development process.
- Construct and normalize conceptual data models.
- Develop logical data models.
- Implement a relational database into a database management system.
- Use database management systems such as Microsoft Access and Oracle SQL Plus.
- Become proficient in using database query language, i.e., SQL.
- Understand the issues related to database performance.

COMPUTER GRAPHICS

Course/Paper: 05BCS-104
BCS Semester V

Prerequisites:
- Must be fluent in C/C++ and familiar with modern development tools such as Visual Studio, XCode, or the GNU toolchain
- We will assume knowledge of the following mathematical topics
    - Vectors, vector operations, and vector spaces
    - Matrices
    - Basic linear algebra such as solving a system of linear equations
    - Polynomials
    - Elementary signal processing (Fourier transform and filtering)

Unit	Content
I	Introduction to Raster scan displays, Storage tube displays, refreshing, flicking, interlacing, color monitors, display processors resolution, working principle of dot matrix, inkjet laser printers, working principles of keyboard, mouse scanner, digitizing camera, track ball , tablets and joysticks, graphical input techniques, positioning techniques, rubber band techniques, dragging etc.
II	Scan conversion techniques, image representation, line drawing, simple DDA, Bresenham's Algorithm, Circle drawing, general method, symmetric DDA, Bresenham's Algorithm, curves, parametric function, Beizier Method, Bsp- line Method.
III	2D & 3D Co-ordinate system, Translation, Rotation, Scaling, Reflection Inverse transformation, Composite transformation, world coordinate system, screen coordinate system, parallel and perspective projection, Representation of 3D object on 2D screen.
IV	Point Clipping. Line Clipping Algorithms, Polygon Clipping algorithms, Introduction to Hidden Surface elimination, Basic illumination model, diffuse reflection, specular reflection, phong shading, Gourand shading ray tracing, color models like RGB, YIQ, CMY, HSV etc.
V	Multimedia components, Multimedia Hardware, SCSI, IDE, MCI, Multimedia data and file formats, RTF, TIFF, MIDI, JPEG, DIB, MPEG, Multimedia Tools, Presentation tools, Authoring tools, presentation.

Text Books:

1. Introduction to Computer Graphics Version 1.1, David J. Eck
2. Fundamentals of Computer Graphics, Dr John Collomosse
3. Computer Graphics David M. Mount, David M. Mount
4. Lecture Notes Fundamentals of Computer Graphics, Prof. Michael Langer

References:

- Donald Hearn, M. Pauline Baker, "Computer Graphics", Pearson
- Rogers, "Procedural Elements of Computer Graphics", TMH
- Asthana, Sinha, "Computer Graphics", Addison Wesley
- Steven Harrington, "Computer Graphics", A Programming Approach
- Udit Agarwal, "Computer Graphics", S.K. Katariya.

Learning Outcomes:

- Demonstrate proficiency in a range of computer graphics technology, including bitmap image editing, vector graphics, page layout, web design, video editing and effects.
- Demonstrate understanding of different printing technologies and pre-press processes.
- Differentiate between vector and bitmap graphics, and effectively use each to the best effect.
- Demonstrate critical thinking by coordinating work between functions and disciplines.
- Demonstrate an appreciation of the cultural, social, political, environmental, and historical aspects of design.
- Demonstrate knowledge of color theory and how to apply it.
- Understand terminology used in the graphic design industry.

TELECOMMUNICATION FUNDAMENTALS

Course/Paper: 05BCS-105
BCS Semester V

Prerequisites:
No special technical skills are required to attend. If you are interested in telecom and data networking and want to get up to speed, this class is for you. We've found over the years that all sorts of folks can become telecom and LAN/WAN, wireless experts. There is really no special academic background or special knowledge needed to get started, just an interest in this exciting and growing field.

Course Objectives:
Upon completion of the course, participants will be able to:

- Discuss the differences among various communications technologies used in advanced transportation management information systems
- Describe the Open Systems Interconnect (OSI) model and how it is used to define telecommunications technologies and protocols
- Explain the fundamentals of data, voice, and video communications

Unit	Content
I	Electromagnetic Spectrum, Frequency Spectrum-Bandwidth-Allocation, Time domain and Frequency domain analysis, Transmission media, , Twisted pair, UTP cables, Coaxial and optical fiber cables, wireless, microwave and satellite transmission, Transmission impairments. Serial and parallel transmission, Simplex, half duplex or full duplex transmission mode. Network, LAN, MAN, WAN, Internet, Intranet, Extranet, Network Topology, Protocols, Layered Architecture, OSI and TCP/P protocol Architecture.
II	Physical Layer : Convention and terminology (bit rate, channel capacity, bandwidth, Signal strength, SNR) Physical transmission media interface(Mechanical, Electrical and Radio interface specification) Modulation (ASK, FSK and PSK, PCM, PAM, Delta Modulations), Line coding (NRZ-L, NRZ–I , Bipolar AMI, Manchester and differential Manchester), Multiplexing (FDM, Synchronous and Statistical TDM) Brief Introduction to Ethernet, SONET/SDH.
III	Data Link Layer: Channel allocation problem, pure and slotted ALOHA Protocols, Persisted And Non-Persisted CSMA, Collision Free Protocols, Digital Cellular Radio and CDMA. Logical Link Sub Layer, MAC Sub layer. Brief Introduction: Frame Relay, PPP.
IV	Switching Networks: Circuit switching Networks, Space and Time division switching, Routing circuit switched networks, control signaling packet switching principles, fixed, flooding and adaptive routing strategies, Brief Introduction: Broadband and Narrowband ISDN, ADSL.
V	Network Devices: Gateway, Router, Bridge, Switch, Hub, Repeater, Multilayer Switch, Protocol Converter, Router, Proxy, Firewall, Multiplexer, Network Card, Modem. Network Technology: DSL, GSM, Bluetooth, Infrared. Brief Introduction to Servers : File Server, Print Server, Mail Server, Proxy Server, Remote Access Server (RAS), Application Server, Web Server, Backup Server

Textbooks:

1. R.P.Yadav , "Telecommunication Engg Fundamentals" , Genius Publications.
2. T. Vishwanathan, "Telecommunication switching system & Network", PHI

References:

- William Stallings, "Data and Computer communication"(PHI,5edi)
- James Martin, "Telecommunication and the computer"-(PHI ,3edi)
- A.S.Tanenbaum, "computer networks"-(PHI 3 edi)

Learning Outcomes:

1. explain basic physical and technical principles of modern digital telecommunications,
2. describe basic principles of operation in modern digital telecommunication equipment and systems,
3. demonstrate measurements and experiments in laboratory on actual components, devices, equipment and systems in telecommunications,
4. describe development and implementation methods of telecommunication systems,
5. examine communication equipment for the technical functionality.

ELECTIVE:--

LOGICAL AND FUNCTIONAL PROGRAMMING

Course/Paper: 05BCS-106.1
BCS Semester V

Prerequisites:

Prerequisites are courses it is suggested you understand before you attempt this course. If you're having a hard time understanding the material in this course, make sure you understand these prerequisites first.

- Introduction to Computer Science
- Introductory Discrete Mathematics for Computer Science

Course Objectives:

Obtaining a basic knowledge and practical experience in functional and logic programming. Introduction into formal concepts used as a theoretical basis for both paradigms.

Unit	Content
I	PROPOSITIONS: Fully parenthesized propositions, Evaluation of constant propositions, Evaluation of proposition in a state. Precedence rules for operators, Tautologies, Propositions a sets of states and Transforming English to prepositional form.
II	REASONING USING EQUIVALENCE TRANSFORMATIONS: The laws of equivalence, rules of substitution and transitivity, formal system of axioms and Inference rules. NATURAL DEDUCTION SYSTEM: Introduction to deductive proofs, Inference rules, proofs and sub-proofs, adding flexibility to the natural deduction system and developing natural deduction system proofs.

III	PREDICATES: Extending the range of a state, Quantification, Free and Bound Identifiers, Textual substitution, Quantification over other ranges and some theorems about textual substitution and states.
IV	LOGIC PROGRAMMING: Introduction to prepositional and predicate calculus, First-order predicate calculus, Format logical systems, PROLOG programming-Facts, Rules and queries, Implementations, Applications, Strengths and Weaknesses.
V	FUNCTIONAL PROGRAMMING: Introduction to lambda calculus-Syntax and semantics, Computability and correctness. Features of Functional Languages-Composition of functions, Functions as first-class Objects, no side effects and clean semantics, LISP Programming-Data types and structures, Scheme dialect, primitive functions, functions for constructing functions and functional forms. Applications of functional languages and comparison of functional and imperative languages.

Textbooks:

1. Thompson, S.: Haskell, The Craft of Functional Programming, ADDISON-WESLEY, 1999, ISBN 0-201-34275-8
2. Nilsson, U., Maluszynski, J.: Logic, Programming and Prolog (2ed), John Wiley & Sons Ltd., 1995
3. Hill, P., Lloyd, J.: The Gödel Programming Language, MIT Press, 1994, ISBN 0-262-08229-2
4. Jones, S.P.: Haskell 98 Language and Libraries, Cambridge University Press, 2003, p. 272, ISBN 0521826144

Reference:

- Foundations of Logic and Functional Programming: Workshop, Trento, Italy, December 15-19, 1986. Proceedings (Lecture Notes in Computer Science) by Mauro Boscarol, Luigia Carlucci Aiello, and Giorgio Levi
- Second Fuji International Workshop on Functional and Logic Programming: Shonan Village Center, Japan Nov 1996 by Tetsuo Ida, Atsushi Ohori, and Masato Takeichi
- Functional and Logic Programming: 8th International Symposium, FLOPS 2006, Fuji-Susono, Japan, April 24-26, 2006, Proceedings (Lecture Notes in Computer Science / Programming and Software Engineering) by Masami Hagiya and Philip Wadler

Learning Outcomes:

- Know the fundamental concepts and terms of the Logic and Functional programming paradigms and how these contrast to the Procedural view of computation.
- Use basic concepts and terms of Logic Programming
- Perform Logic Program computation, minimal model computation and recursive predicate definition for list processing.
- Use basic concepts and terms of Functional Programming.
- Develop programs in ML – simple and recursive functions, list processing, function composition.
- Apply functional programming computation and develop higher order functions.

INFORMATION THEORY & CODING

Course/Paper: 05BCS-106.2
BCS Semester V

Prerequisite:
Probability and Random Processes, Digital Communications

Course Objectives:

Unit	Content
I	Elements Of Information Theory: Measure of information, average information, entropy, information rate. Communication channel, discrete and continuous channel
II	Shannon-Hartley theorem and its implications. Channel capacity, Gaussian channel and bandwidth-S/N tradeoff.
III	Introduction of Coding: types of efforts, types of codes, error control coding, methods of controlling errors
IV	Linear Block and Binary Cyclic Codes: matrix decryption of linear block codes, error detection and error correction capabilities of linear block codes. Hamming codes, structure of cyclic codes, encoding using an (n-k) bit shift register syndrome calculation, its error detection & correction, special classes of cyclic codes bch.
V	Burst and Convolution Codes: burst and random error correcting codes, encoders for convolution codes. Decoders for convolution codes

Text Books :
1. Elements of Information Theory by Thomas Cover, Joy Thomas
2. Channel Codes: Classical and Modern by William Ryan, Shu Lin

Reference:
- Coding and Information Theory (Graduate Texts in Mathematics) by Steven Roman
- Information and Coding Theory (Springer Undergraduate Mathematics Series) by Gareth A. Jones and J. Mary Jones
- Information Theory and Network Coding (Information Technology: Transmission, Processing and Storage) by Raymond W. Yeung
- Fundamentals of Information Theory and Coding Design (Discrete Mathematics and Its Applications) by Roberto Togneri and Christopher J.S deSilva
- Anoop Singh Poonia, "Information Theory of Coding", Genius Pub.

Learning Outcomes:
The participants must at the end of the course be able to: Understand and explain the basic concepts of information theory, source coding, channel and channel capacity, channel coding and relation among them. Describe the real life applications based on the fundamental theory.

ADVANCED DATA STRUCTURES

Course/Paper: 05BCS-106.3
BCS Semester V

Prerequisites:
The recommended prerequisite is 6.854 Advanced Algorithms. This is the entry-level graduate course in Theory/Algorithms, and it should be taken before jumping into any deeper graduate courses. However, we recognize that some highly qualified students have not yet taken 6.854 for objective reasons. Therefore, we will try to accommodate students who have only taken 6.046, and we will not rely on 6.854 material. Nonetheless, in order to use this option, you must have an excellent understanding of algorithms at the undergraduate level; such a level of understanding can be reached through an A+ in 6.046, relevant UROP, involvement in computer competitions etc.

Course Objectives
1. To teach efficient storage mechanisms of data for an easy access.
2. To design and implementation of various basic and advanced data structures.
3. To introduce various techniques for representation of the data in the real world.
4. To develop application using data structures.
5. To teach the concept of protection and management of data.
6. To improve the logical ability

Unit	Content
I	ADVANCED TREES: Definitions Operations on Weight Balanced Trees (Huffman Trees), 2-3 Trees and Red- Black Trees. Augmenting Red-Black Trees to Dynamic Order Statistics and Interval Tree Applications. Operations on Disjoint sets and its union-find problem Implementing Sets. Dictionaries, Priority Queues and Concatenable Queues using 2-3 Trees.
II	MERGEABLE HEAPS: Merge able Heap Operations, Binomial Trees Implementing Binomial Heaps and its Operations, 2-3-4. Trees and 2-3-4 Heaps. Amortization analysis and Potential Function of Fibonacci Heap Implementing Fibonacci Heap. SORTING NETWORK**:** Comparison network, zero-one principle, bitonic sorting and merging network sorter.
III	GRAPH THEORY DEFINITIONS: Definitions of Isomorphic Components. Circuits, Fundamental Circuits, Cut-sets. Cut-Vertices Planer and Dual graphs, Spanning Trees, Kuratovski's two Graphs.
IV	GRAPH THEORY ALGORITHMS**:** Algorithms for Connectedness, Finding all Spanning Trees in a Weighted Graph and Planarity Testing, Breadth First and Depth First Search, Topological Sort, Strongly Connected Components and Articulation Point. Single Min-Cut Max-Flow theorem of Network Flows. Ford-Fulkerson Max Flow Algorithms
V	NUMBER THEORITIC ALGORITHM: Number theoretic notation, Division theorem, GCD recursion, Modular arithmetic, Solving Linear equation, Chinese remainder theorem, power of an element, RSA public key Crypto system, primality Testing and Integer Factorization.

Textbooks:

1. Mehlhorn "Data Structures and Algorithms: 1, Searching and Sorting", Springer Verlag EATCP
2. Monograph on Theoretical Computer Science

References:

- **Coreman, Rivest, Lisserson**, : "Algorithm", **PHI.**
- Motwani and Raghavan "Randomized Algorithms", Cambridge University Press
- Preparata and Shamos "Computational Geometry", Springer Verlag

Learning outcomes:

1. Students develop knowledge of basic data structures for storage and retrieval of ordered or unordered data. Data structures include: arrays, linked lists, binary trees, heaps, and hash tables.
2. Students develop knowledge of applications of data structures including the ability to implement algorithms for the creation, insertion, deletion, searching, and sorting of each data structure.
3. Students learn to analyze and compare algorithms for efficiency using Big-O notation. Students implement projects requiring the implementation of the above data structures.

Laboratories:--

SOFTWARE ENGINEERING LAB

Course/Paper: 05BCS-201
BCS Semester V

Prerequisites:
Analytical, problem-solving, teamwork and communication skills as well as creativity and attention to detail; familiarity with database management, development environment, program testing and other related software

Course Objectives:
This course introduces the concepts and methods required for the construction of large software intensive systems. It aims to develop a broad understanding of the discipline of software engineering. It aims to set these techniques in an appropriate engineering and management context.

Experiments:
In this lab first 8 experiments are to practice software engineering techniques. Use any open source CASE tool. Many of them are available at www.sourceforge.net. You can choose any other CASE tool, as per choice.
Language : C++ / JAVA
Design Approach : Object Oriented
These designing can be done on any automation system e.g. library management system, billing system, payroll system, bus reservation

system, gas agency management system, book-shop management system, students management system.

1. Do a feasibility study
2. Document all the requirements as specified by customer in Software Requirement Specification
3. Design sequence diagrams for project
4. Design Collaboration diagram
5. Design Data Flow Diagram for the project
6. Design Entity Relation Diagram for the project
7. Design Class diagram
8. Design at least 10 test cases for each module.
9. -10: Code and test the project, which you have designed in last 8 labs.

Textbooks:

1. Shooman, M. : Software Engineering, McGraw-Hill
2. Shere: Software Engineering & Management, Prectice-Hall.

References:

- Pressman Roger: Software Engineering -A Practitioner's Approach; Tata McGraw Hill, N.Delhi,-1991
- Jalote Pankaj: An Integrated Approach to Software Engineering; Narosa, New Delhi, 1991.
- Fairley, R.E. : Software Engineering Concepts, McGraw-Hill,

Learning Outcomes:

Students will be able to: Model the structure and behavior a *software* system the UML class diagrams and state diagrams. Design a solution to a given problem using one or more design patterns and implement the design in a programming language

COMPUTER ARCHITECTURE LAB

Course/Paper: 05BCS-202
BCS Semester V

Prerequisites:

- Understanding of basic computer organization, including familiarity with such components as CPU, ALU, multiplexors, registers, main memory, caches, and buses
- familiarity with the roles of compilers, assemblers, and operating systems
- some familiarity with assembly language
- ability to understand simple C programs and to run programs in a UNIX environment, and
- familiarity with the representation of numbers in digital computers

Course Objectives:

This course presents key principles underlying the design of modern digital computers. The course introduces quantitative techniques used to guide the design process. It describes CPU performance issues and introduces instruction set architectures. The course then uses a hypothetical computer design, with a simple RISC architecture, to show how modern digital

computers are implemented, first using a simple non-pipelined implementation, followed by a higher-performance pipelined implementation. The major hazards introduced by pipelining, including structural hazards, data hazards, and control hazards are discussed and techniques for overcoming them are described. Additional topics covered in this course include the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

Experiments:

This lab will be based on assembly programming on of RISC processor simulator SPIM.
SPIM simulator is available at site http://pages.cs.wisc.edu/~larus/spim.html.

SPIM exercises

1. Read an integer from the keyboard and print it out if (n => n_min AND n <= n_max).
2. Read an integer from the keyboard and print out the following as per switch-case statement

```
Switch (n)
{ n <= 10 print "not a lot"
n == 12 print "a dozen"
n == 13 print "a baker's dozen"
n == 20 print "a score"
n >= 100 print "lots and lots"
n != 42 print "integer"
Otherwise print "you have the answer!"
```

3. Read a string from the keyboard and count the number of letters. Use the equivalent of following for loop
to count number of chars. **for (s1=0; str[s1] != '\n'; ++s1)**
4. Print out a line of characters using simple procedure call.
5. Print out a triangle of characters using recursive procedure call.
6. Print factorial of a number using recursion.
7. Print reverse string after reading from keyboard.
8. Print a string after swapping case of each letter.
9. Print an integer in binary and hex.
10. Implement bubble sort algorithm.
11. Print Pascal Triangle of base size 12.
12. Evaluate and print Ackerman function.

Textbooks:

1. D.Sima, T. Fountain, P. Kacsuk. “Advance computer Architectures: A Design Space Approach”, Addision Wesley, 1997.
2. M.J. Flynn, “Computer Architecture: Pipelined and Parallel Processor Design”, Narosa Publishing House/Jones

References:

- M. Mano, “Computer System Architecture”, PHI India Pvt. Ltd.
- Tannenbaum, “ Structured Computer Organization”, PHI India Pvt. Ltd.
- Stallings, “Computer Organization”, PHI India Pvt. Ltd.

Learning Outcomes:

Students gain an understanding of the design of the memory hierarchy in modern digital computers, caching and virtual storage techniques, multiprocessor systems, and distributed shared memory systems.

DATABASE MANAGEMENT LAB

Course/Paper: 05BCS-203
BCS Semester V

Prerequisites:
You are required to have the background from a data structures course (e.g., lists, hash tables, arrays, search trees) and strong programming experience. Programming skills: The projects are in Java and C++. The knowledge of C++ is assumed. You will be given a brief overview of Java and expected to be able to pick up the language on you own, if you haven't used it before.

Course Objective:
- Knowledge of DBMS, both in terms of use and implementation/design
- Experience with SQL
- Increased proficiency with the programming language C++
- Experience working as part of team
- Experience with analysis and design of (DB) software

Experimients:
Student can use My Sql (preferred open source DBMS) or any other Commercial DBMS tool (MS-Access / ORACLE) at backend and C++ (preferred) VB/JAVA at front end.
1. (a) Write a C++ program to store students records (roll no, name, father name) of a class using file handling.
(Using C++ and File handling).
(b) Re-write program 1, using any DBMS and any compatible language.(C++/MySQL) (VB and MS-Access)
2. Database creation/ deletion, table creation/ deletion.
(a) Write a program to take a string as input from user. Create a database of same name. Now ask user to input two more string, create two tables of these names in above database.
(b) Write a program, which ask user to enter database name and table name to delete. If database exist and table exist then delete that table.
3. Write a program, which ask user to enter a valid SQL query and display the result of that query.
4. Write a program in C++ to parse the user entered query and check the validity of query. (Only SELECT query with WHERE clause)
5 - 6. Create a database db1, having two tables t1 (id, name, age) and t2 (id, subject, marks).
(a) Write a query to display name and age of given id (id should be asked as input).
(b) Write a query to display average age of all students.
(c) Write a query to display mark-sheet of any student (whose id is given as input).
(d) Display list of all students sorted by the total marks in all subjects.
7 - 8. Design a Loan Approval and Repayment System to handle Customer's Application for Loan and handle loan repayments by depositing installments and reducing balances.
9 -10. Design a Video Library Management System for managing issue and return of Video tapes/CD and manage customer's queries.

Text Books:
Database Management Systems , Ramakrishnan & Gehrke

References:

- Date C J, “An Introduction To Database System”, Addision Wesley
- Korth, Silbertz, Sudarshan, “Database Concepts”, Mc Graw Hill
- Bipin C. Desai, “An introduction to Database Systems”, Galgotia
- F. H. Lochousky, DC Tsichritzis"DBMS" NewYork Academic Press
- N. Goodman, V. Hadzilacos "Concurrency Control and Recovery in Data Base System" Addison Wesley.

Learning Outcomes:

- Understand database concepts and structures.
- Explain terms related to database design and management.
- Understand the objectives of data and information management.
- Understand data modeling and database development process.
- Construct and normalize conceptual data models.
- Develop logical data models.
- Implement a relational database into a database management system.
- Use database management systems such as Microsoft Access and Oracle SQL Plus.
- Become proficient in using database query language, i.e., SQL.
- Understand the issues related to database performance.

COMPUTER GRAPHICS LAB

Course/Paper: 05BCS-204
BCS Semester V

Prerequisites:

- Must be fluent in C/C++ and familiar with modern development tools such as Visual Studio, XCode, or the GNU toolchain
- We will assume knowledge of the following mathematical topics
    - Vectors, vector operations, and vector spaces
    - Matrices
    - Basic linear algebra such as solving a system of linear equations
    - Polynomials
    - Elementary signal processing (Fourier transform and filtering)

Experiments:

1. Implementation of line generation using slope’s method, DDA and Bresenham’s algorithms.
2. Implementation of circle generation using Mid-point method and Bresenham’s algorithm.
3. Implementation of ellipse generation using Mid-point method.
4. Implementation of polygon filling using Flood-fill, Boundary-fill and Scan-line algorithms.
5. Implementation of 2D transformation: Translation, Scaling, Rotation, Mirror Reflection and Shearing (write a menu driven program).
6. Implementation of Line Clipping using Cohen-Sutherland algorithm and Bisection Method.
7. Implementation of Polygon Clipping using Sutherland-Hodgman algorithm.

8. Implementation of 3D geometric transformations: Translation, Scalind and rotation.
9. Implementation of Curve generation using Interpolation methods.
10. Implementation of Curve generation using B-spline and Bezier curves.
11. Implementation of any one of Back face removal algorithms such as Depth-Buffer algorithm, Painter's algorithm, Warnock's algorithm, Scan-line algorithm)

Text Books:

1. Introduction to Computer Graphics Version 1.1, David J. Eck
2. Fundamentals of Computer Graphics, Dr John Collomosse
3. Computer Graphics David M. Mount, David M. Mount
4. Lecture Notes Fundamentals of Computer Graphics, Prof. Michael Langer

References:

- Donald Hearn, M. Pauline Baker, "Computer Graphics", Pearson
- Rogers, "Procedural Elements of Computer Graphics", TMH
- Asthana, Sinha, "Computer Graphics", Addison Wesley
- Steven Harrington, "Computer Graphics", A Programming Approach
- Udit Agarwal, "Computer Graphics", S.K. Katariya.

Learning Outcomes:

- Demonstrate proficiency in a range of computer graphics technology, including bitmap image editing, vector graphics, page layout, web design, video editing and effects.
- Demonstrate understanding of different printing technologies and pre-press processes.
- Differentiate between vector and bitmap graphics, and effectively use each to the best effect.
- Demonstrate critical thinking by coordinating work between functions and disciplines.
- Demonstrate an appreciation of the cultural, social, political, environmental, and historical aspects of design.
- Demonstrate knowledge of color theory and how to apply it.
- Understand terminology used in the graphic design industry.

VI semester

OPERATING SYSTEMS

Course/Paper: 06BCS-101
BCS Semester VI

Prerequisite:
Familiar with, basic hardware and software aspects of computer systems organization

Course objectives:
To provide an introduction of operating system concepts as reference to real systems. To give exposure to the professional responsibilities that are part of operating system design and development.

Unit	Content
I	Introduction to Operating Systems, Operating system services, multiprogramming, time-sharing system, storage structures, system calls, multiprocessor system. Basic concepts of CPU scheduling, Scheduling criteria, Scheduling algorithms, algorithm evaluation, multiple processor scheduling, real time scheduling I/0 devices organization, I/0 devices organization, I/0 devices organization, I/0 buffering.
II	Process concept, process scheduling, operations on processes, threads, inter-process communication, precedence graphs, critical section problem, semaphores, classical problems of synchronization. Deadlock problem, deadlock characterization, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock, Methods for deadlock handling.
III	Concepts of memory management, logical and physical address space, swapping, contiguous and non-contiguous allocation, paging, segmentation, and paging combined with segmentation.
IV	Concepts of virtual memory, demand paging, page replacement algorithms, allocation of frames, thrashing, demand segmentation. Security threads protection intruders-Viruses-trusted system.
V	Disk scheduling, file concepts, file access methods, allocation methods, directory systems, file protection, introduction to distributed systems and parallel processing case study.

Text Books:
1. Operating System Concepts- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Operating systems- A Concept based Approach-D.M.Dhamdhere, 2nd Edition, TMH

Reference Books :
- Operating Systems' – Internal and Design Principles Stallings, Fifth Edition–2005, Pearson education/PHI
- Operating System A Design Approach-Crowley, TMH.
- Modern Operating Systems, Andrew S Tanenbaum 2nd edition Pearson/PHI.

Learning Outcomes:

1. Understand functional architecture of an operating system
2. Develop algorithms for subsystem components
3. Design device drivers and multi threading libraries for a tiny OS
4. Develop application programs using UNIX system calls
5. Design and solve synchronization problems
6. Understand standard UNIX and FAT file systems

COMPUTER NETWORKS

Course/Paper: 06BCS-102
BCS Semester VI

Pre Requisite:

Students need an introductory course in probability, a strong understanding of bits and bytes, and knowledge of how computers lay out data in memory.

Course Objectives:

To expose the students to the basic principles of the technology of data communications and networking. Upon completion of this course, the students will have a good working knowledge :

- To understand the concept of data communication and modulation techniques.
- To comprehend the use of different types of transmission media and network devices.
- To understand the error detection and correction in transmission of data.
- To understand the concept of flow control, error control and LAN protocols.
- To understand the functions performed by Network Management System
- To understand security issues.

Unit	Content
I	Network, Network Protocols, Edge, Access Networks and Physical Media, Protocol Layers and their services models, Internet Backbones, NAP's and ISPs.
II	Application Layer: Protocol and Service Provided by application layer, transport protocols. The world wide web. HTTP, Message formats, User Server Interaction and Web caches. FTP commands and replies. Electronic Mail, SMTP, Mail Message Formats and MIME and Mail Access Protocols DNS The internet's directory service DNS records and Message.
III	Transport Layer: Transport Layer Service and Principles, Multiplexing and De multiplexing applications, connectionless Transport. UDP Segment structure and UDP Checksum. Principles of Reliable Data Transfer-Go back to N and Selective Repeat. Connection Oriented Transport TCP Connection and Segment Structure, Sequence Numbers and acknowledgement numbers, Telnet, Round trip time and timeout. TCP connection management.
IV	Network Layer and Routing: Network service model, Routing principles. Link State routing Algorithm, A distant Vector routing & OSPF algorithm. Router Components; Input Prot, Switching fabric and output port. IPV6 Packet format. Point To Point Protocol (PPP), transition States, PPP Layers-Physical Layer and Data Link Layer, Link Control Protocols. LCP Packets and options. Authentication PAP and CHAP, Network Control Protocol (NCP).

V	Sonet/SDH: Synchronous Transport Signals. Physical configuration-SONET Devices, Sections, Lines and Paths. SONET Layers-Photonic Layer, section layer, line layer, path layer and device layer relationship. Sonet Frame format. Section overhead, Line overhead and path overhead. Virtual Tributaries and types of VTs.

Text Books:

1. Data Communications and Networking, Fourth Edition by Behrouza A.Forouzan,TMH.
2. Introduction to Data communications and Networking,W.Tomasi,Pearson education.
3. Data and Computer Communications,G.S.Hura and M.Singhal,CRC Press,Taylor and Francis Group.

Reference Books:

- Computer Networks,A.S.Tanenbaum,4th edition,Pearson education
- An Engineering Approach to Computer NetworksS. Keshav,2nd Edition,Pearson Education.
- Understanding communications and Networks, 3rd Edition, W.A.Shay,Cengage Learning.

Learning Outcomes:

1. A strong foundation in core Computer science and engineering, both theoretical and applie concepts.
2. An ability to apply knowledge of mathematics, science and engineering to real world problems.
3. An ability to communicate effectively, both in writing and oral.
4. The broad education necessary to understand the impact of computer science and engineering solutions in the scientific, societal and human context.
5. A recognition of the need for, an ability to engage in life-long learning

DESIGN & ANALYSIS OF ALGORITHMS

Course/Paper: 06BCS-103
BCS Semester VI

Pre requisite:
Programming fundamentals and elementary data structures.

Course Objective:
The objective of the course is to teach techniques for effective problem solving in computing. The use of different paradigms of problem solving will be used to illustrate clever and efficient ways to solve a given problem. In each case emphasis will be placed on rigorously proving correctness of the algorithm. In addition, the analysis of the algorithm will be used to show the efficiency of the algorithm over the naive techniques.

Unit	Content
I	BACKGROUND: Review of Algorithm Complexity and Order Notations and Sorting Methods. DIVIDE AND CONQUER METHOD: Binary Search, Merge Sort, Quick sort and strassen's matrix multiplication algorithms. GREEDY METHOD: Knapsack Problem, Job Sequencing, Optimal Merge Patterns and Minimal Spanning Trees.
II	DYNAMIC PROGRAMMING: Matrix Chain Multiplication. Longest Common Subsequence and 0/1 Knapsack Problem. BRANCH AND BOUND: Traveling Salesman Problem and Lower Bound Theory. Backtracking Algorithms and queens problem.
III	PATTERN MATCHING ALGORITHMS: Naïve and Rabin Karp string matching algorithms, KMP Matcher and Boyer Moore Algorithms. ASSIGNMENT PROBLEMS: Formulation of Assignment and Quadratic Assignment Problem.
IV	RANDOMIZED ALGORITHMS. Las Vegas algorithms, Monte Carlo algorithms, randomized algorithm for Min-Cut, randomized algorithm for 2-SAT. Problem definition of Multi commodity flow, Flow shop scheduling and Network capacity assignment problems.
V	PROBLEM CLASSES NP, NP-HARD AND NP-COMPLETE: Definitions of P, NP-Hard and NP-Complete Problems. Decision Problems. Cook's Theorem. Proving NP-Complete Problems - Satisfiability problem and Vertex Cover Problem. Approximation Algorithms for Vertex Cover and Set Cover Problem.

Textbooks

- Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, MIT Press, Third Edition, 2009.
- Basse, "Computer Algorithms: Introduction to Design & Analysis", Addision Wesley.
- Horowitz & Sahani, "Fundamental of Computer Algorithm", Galgotia.

References:

1. Coreman, Rivest, Lisserson, : “Algorithm", PHI.
2. Motwani and Raghavan "Randomized Algorithms", Cambridge University Press
3. Preparata and Shamos "Computational Geometry", Springer Verlag
4. Mehlhorn "Data Structures and Algorithms: 1, Searching and Sorting", Springer Verlag EATCP

Course Learning Outcomes

On successful completion of this course you should be able to:

- Compare, contrast, and apply the key algorithmic design paradigms: brute force, divide and conquer, decrease and conquer, transform and conquer, greedy, dynamic programming and iterative improvement;
- Compare, contrast, and apply key data structures: trees, lists, stacks, queues, hash tables and graph representations;

- Define, compare, analyse, and solve general algorithmic problem types: sorting, searching, graphs and geometric;
- Theoretically compare and analyse the time complexities of algorithms and data structures; and
- Implement, empirically compare, and apply fundamental algorithms and data structures to real-world problems.

EMBEDDED SYSTEMS

Course/Paper: 06BCS-104
BCS Semester VI

Prerequisite:
- Sequential Logic Concepts
- Basic Computer Organization: CPU, memory, I/O, buses, interrupts
- Digital Systems test procedures

Course Objectives:
1. To introduce the difference between embedded systems and general purpose systems.
2. To optimize hardware designs of custom single-purpose processors.
3. To compare different approaches in optimizing general-purpose processors.
4. To introduce different peripheral interfaces to embedded systems.
5. To understand the design tradeoffs made by different models of embedded systems.
6. To apply knowledge gained in software-hardware integration in team-based projects.

Unit	Content
I	Overview of Embedded System: Embedded System, Categories and Requirements of Embedded Systems, Challenges and Issues in Embedded Software Development, Applications of Embedded Systems in Consumer Electronics, Control System, Biomedical Systems, Handheld computers, Communication devices.
II	Embedded Hardware & Software Development Environment: Hardware Architecture, Micro- Controller Architecture, Communication Interface Standards, Embedded System Development Process, Embedded Operating systems, Types of Embedded Operating systems.
III	Design quality and Microcontroller: Quality matrix, software and hardware, Estimation , 8 Bit microcontrollers Architecture, on chip peripherals, instruction set/programming of Intel MCS51 family (8 bit) Inter facing of 8051 with LCD, ADC, sensors, stepper motor, key board, DAC, memory .
IV	Real Time & Database Applications: Real- Time Embedded Software Development, Sending a Message over a Serial Link, Simulation of a Process Control System, Controlling an Appliance from the RTLinux System, Embedded Database Applications using examples like Salary Survey, Energy Meter Readings.

V	Programming Languages for Embedded Systems: Tools for building embedded systems - with case studies. Microchip PIC16 family PIC16F873 processor features architecture memory organization register file map I/O ports PORTA - PORTB PORTC Data EEPROM and flash program memory Asynchronous serial port SPI mode I2C mode.

Textbook:

1. Jean Labrosse, MicroC/OS-II: The Real-Time Kernel, 2nd Edition, CMP Books, 2002
2. Embedded Systems: Architecture, Programming and Design by Raj Kamal.
3. Embedded System Design: Modeling, Synthesis and Verification by Daniel D. Gajski, Samar Abdi, Andreas Gerstlauer - 2009

References:

- R. Gupta, "Co-synthesis of Hardware and Software for Embedded Systems", Kluwer

Learning Outcomes:

At the end of this course, each passing student should be able to perform the following tasks Compare embedded system design models using different processor technologies (single purpose, general-purpose, application specific processors).

THEORY OF COMPUTATION

Course/Paper: 06BCS-105
BCS Semester VI

Prerequiste :

Set algebra, elementary formal logic, constructing proofs, recurrence relations.

Course Objectives :

The Learning Objective of Theory of Computation are such that the student will

- Study various types of Finite Automata.
- Appreciate to prove equivalence of language described by Automata.
- Understand the grammar and PDA for a given language.
- Grasp the comprehensive knowledge of Turing Machine.
- Acquire awareness about the concepts of tractability, decidability, NP completeness.
- Understand the challenge for Theoretical Computer Science and its contribution.

Unit	Content
I	Finite Automata & Regular Expression: Basic Concepts of finite state system, Deterministic and non-deterministic finite automation and designing regular expressions, relationship between regular expression & Finite automata minimization of finite automation mealy & Moore Machines.
II	Regular Sets of Regular Grammars**:** Basic Definition of Formal Language and Grammars. Regular Sets and Regular Grammars, closure proportion of regular sets, Pumping lemma for regular sets, decision Algorithms for regular sets, Myhell_Nerod Theory & Organization of Finite Automata.

III	Context Free Languages& Pushdown Automata: Context Free Grammars – Derivations and Languages – Relationship between derivation and derivation trees – ambiguity – simplification of CEG – Greiback Normal form – Chomsky normal forms – Problems related to CNF and GNF Pushdown Automata: Definitions – Moves – Instantaneous descriptions – Deterministic pushdown automata – Pushdown automata and CFL - pumping lemma for CFL - Applications of pumping Lemma.
IV	Turing Machines: Turing machines – Computable Languages and functions – Turing Machine constructions – Storage in finite control – multiple tracks – checking of symbols – subroutines – two way infinite tape. Undecidability: Properties of recursive and Recursively enumerable languages – Universal Turing Machines as an undecidable problem – Universal Languages – Rice's Theorems.
V	Linear bounded Automata Context Sensitive Language: Chomsky Hierarchy of Languages and automata, Basic Definition& descriptions of Theory & Organization of Linear bounded Automata Properties of context-sensitive languages.

Text Books

1. Introduction to the Theory of Computation, by Michael Sipser, latest edition, PWS Publishing Company.
2. Joha E. Hopcroft, Jeffery Ullman,"Introduction to Automata theory, Langauges& computation" ,Narosa Publishers.
3. K.L.P. Mishra & N.Chandrasekaran,"Theory of Computer Science", PHI Learning.
4. Michael Sipsev,"Theory of Computation",Cenage Learning.

Reference Books

- **Hopcroft, Ullman**, "Introduction to Automata Theory, Language and Computation",Nerosa Publishing House
- **K.L.P. Mishra and N.Chandrasekaran**, "Theory of Computer Science (Automata, Languages and Computation)", **PHI**
- **Martin J. C**., "Introduction to Languages and Theory of Computations", **TMH**

Learning Outcomes:

1. Be familiar with regular languages and finite automata.
2. Be familiar with context-free languages, push-down automata, and Turing recognizable languages.
3. Be exposed to a broad overview of the theoretical foundations of computer science.
4. Be familiar with thinking analytically and intuitively for problem-solving situations in related areas of theory in computer science.

ELECTIVE:--

DIGITAL SIGNAL PROCESSING

Course/Paper: 06BCS-106.1
BCS Semester VI

Prerequisite :
Although the course has no formal pre-requisites, strong knowledge of linear algebra and experience in MATLAB is necessary.

Course Objective:
This course is designed to provide students the fundamentals of discrete-time signals, signal transforms, and digital filter design. Through this course, students are expected to achieve a basic understanding of digital signal processing. Ultimately, it is hoped that through learning this course students will be equipped with a clear picture of DSP as well as a necessary foundation for further study of advanced DSP topics in the future.

Unit	Content
I	Flow Graph and Matrix Representation of Digital Filters: Signal flow graph representation of digital network, matrix representation, basic network structures for IIR and FIR systems, Telligen's theorem for digital filters and its applications.
II	Digital filter Design Techniques: Design of IIR and FIR digital filters, computer aided design of IIR and FIR filters, comparison of IIR and FIR digital filters.
III	Computation of the Discrete Fourier Transform: Goertzel algorithm, FT algorithms, decimation in time and frequency, FFFT algorithm for N a composite number, Chirp Z transforms (CZT).
IV	Discrete Random Signals: Discrete time random process ,averages spectrum representations of infinite energy signals, response of linear system to random signals
V	Power Spectrum Estimation: Basic principles of spectrum estimation, estimates of the auto covariance, power spectrum, cross covariance and cross spectrum.

Text Books:

1. Required: J. G. Proakis, D. G. Manolakis, Ã¢â‚¬Å“Digital Signal Processing: Principles, Algorithms and Applications,Ã¢â‚¬Â • Prentice Hall, Fourth Edition.
2. Optional: J. G. Proakis, V. K. Ingle, Ã¢â‚¬Å“Student Manual for Digital Signal Processing with MATLAB,Ã¢â‚¬Â • Prentice Hall;
3. Vinay K. Ingle, John G. Proakis, "Digital Signal Processing Using Matlab," Thomson Learning, First or Second Edition.

Reference Books s:

- Oppenheim Alan V, Schafer Ronald W and Buck John R, Discrete-Time Signal Processing, 3rd Edition, Pearson Education, 2009.

- Prandoni Paolo and Vetterli Martin, Signal Processing for Communication, 1st Edition, EPFL Press. (TK5102.9.P899)
- Mitra Sanjit K, Digital Signal Processing : A Computer Based Approach, 4th Edition, McGraw-Hill, 2011. (TK5102.9.M684 201

Learning outcomes

Upon successfully completing the course, students should have an understanding of Digital Signal Processing, as well as knowledge of some of its applications. Students will also understand signals and transforms, filters, random variables and statistical signal processing, and time-frequency analysis among other topics.

ADVANCED SOFTWARE ENGINEERING

Course/Paper: 06BCS-106.2
BCS Semester VI

Prerequisite:

- Student should have prior basic knowledge on Software attributes, Process models
- Student should have some basic knowledge on Testing, Maintenance.

Objectives

- To understand the importance of software engineering lifecycle models in the development of software
- To understand the various design principles in modelling a software
- To develop a software which adheres to the standard benchmarks
- To undergo the technical know in the process of software testing

Unit	Content
I	SOFTWARE CONFIGURATION MANAGEMENT: SCM Process, Objects in Software configuration, Version control, Change control, Configuration audit, Status reporting, SCM standards .SOFTWARE QUALITY ASSURANCE: Quality Concepts, Quality Movement, SQA Activities and Formal Approaches to SQA.
II	SOFTWARE TESTING AND DEBUGGING: Software Testing Fundamentals .Text Case Design ,White –Box Testing, Basis Path testing, Control Structure Testing, Black Box Testing and Testing for Specialized Environments, Architectures and Applications. Program Error, Debugging Process (Information Gathering, Fault Isolation, Fault Confirmation, Documentation, Fixing fault, Testing) Debugging Example.
III	MANAGING TEAM: Understanding behavior and selecting right person for the job, Motivation, working in groups, decision making, leadership and organizational structures. INTERNATIONAL STANDARDS: Importance and defining software quality, ISO 9126, BS 6079 planning steps, ISO 12207 approach to software lifecycle data.

IV	WEB ENGINEERING**:** Attributes of Web-Based Applications. Process, Modeling activity, Analysis modeling for WebApps, Design- functional, information & interaction, testing WebApps- content, navigation, configuration, and performance testing.
V	PROJECT MANAGEMENT FOR SPECIAL CLASSES OF SOFTWARE PROJECTS: Using CASE tools, CBSE, Re-engineering, forward engineering, client/server software engineering, outsourcing, Software project management standards. Change and Content Management of Web Engineering.

Text Book

1. R. S. Pressman, "Software Engineering - A practitioners approach", III Edition, McGraw Hill International editions, 1992.
2. Pankaj Jalote, "An Integrated Approach to software Engineering", Springer Verlag, 1997.
3. James F. Peters and Witold Pedryez, "Software Engineering – An Engineering Approach", John Wiley and Sons, New Delhi.

Reference Books

- **Carlo Ghezzi, M. Jarayeri, D. Manodrioli**, Fundamentals of Software
- Engineering, **PHI**
- **Slini puri** "Advance Software Engineering", **Genius Publication.**
- 5.**Archna Jain/Nikita Jain/Prinka Panjabi** "Advance Software Engineering", **Ashirwad Publication.**

Learning Outcomes

1. Ability to develop software projects using software practices
2. Knowledge of software cost estimation and software quality metrics

MICROWAVE AND SATELLITE COMMUNICATION

Course/Paper: 06BCS-106.3
BCS Semester VI

Prerequisites:
Students should have knowledge of C And C++.

Lab Objective:
1. To introduce Basic Unix general purpose Commands
2. To learn network Unix commands.
3. To learn C programming in Unix editor environment.
4. To learn shell script and sed concepts.
5. To learn file management and permission advance commands.
6. To learn awk, grap, perl scripts.

Unit	Content
I	Microwave Transmission System: General representation of E M field in terms of TEM, TE and TM components, Uniform guide structures, rectangular wave guides, Circular Wave guides, Solution in terms of various modes, Properties of propagating and evanescent modes, Dominant modes, Normalized model voltages and currents, Power flow and energy storage in modes frequency range of operation for single mode working, effect of higher order modes, Strip line and micro strip lines-general properties, Comparison of coaxial, Micro strip and rectangular wave guides in terms of band width, power handling capacity, economical consideration etc.
II	Origin and brief history of satellite communication; Elements of a satellite communication link; Current status of satellite communication. Orbital Mechanism and Launching of Satellite***:*** Equation of orbit, Describing the orbit, Location the satellite in the orbit, Locating the satellite with respect to earth, Orbital elements, Look angle determination, Elevation and Azimuth calculation, Geostationary and other orbits, Orbital perturbations, Orbit determination, Mechanics of launching a synchronous satellite, Selecting a launch vehicle.
III	Space Craft: Satellite subsystems, Altitude and Orbit Control (AOCS), Telemetry, Tracking and Command (TT&C). Communication subsystems, Transponders, Spacecraft antennas, Frequency re-use antennas.
IV	Satellite Channel and Link Design: Basic transmission theory, Noise temperature, Calculation of system noise temperature, Noise figure, G/T ratio of earth stations, Design of down links and uplinks using C/N ratio, FM factor for multi-channel signals, Link Design for FDM/FM, TV signals and Digital Signals.
V	Earth Station Technology: Earth station design, Basic antenna theory, antenna noise temperature; Tracking; Design of small earth station antennas, Low noise amplifiers, High power amplifiers, FDM and TDM systems.

Text Books :

1. UNIX – Concepts and Applications, Sumitabha Das, 4th Edition, Tata McGraw Hill, 2006
2. Unix, concepts and applications by Sumitabha Das, McGraw-Hill
3. Mastering Shell Scripting, Randal. K. Michael , Second Edition, Wiley Publication

.

Reference Books :

- UNIX and Shell Programming, Behrouz A. Forouzan and Richard F.Gilberg, Thomson, 2005. 2. UNIX & Shell Programming, M.G. Venkateshmurthy, Pearson Education, 2005.
- Unix Shell Programming by Yashwant Kanetkar.
- Unix shell programming by forozun.

Learning Outcome

1. Ability to understand the Unix Operating System and the working of the built in commands available in unix.
2. Analyze the working of the user defined commands and will be able change the permissions associated with files.
3. Understanding the concept of Shell and the different usage of the commands in shell. CO 4: Ability to program in AWK language
4. Identify and analyze various perl programs and administrator privileges

Laboratories:--

SHELL PROGRAMMING LAB

Course/Paper: 06BCS-201
BCS Semester VI

Prerequisites:
Students should have knowledge of C And C++.

Lab Objective:
To introduce Basic Unix general purpose Commands

1. To learn network Unix commands.
2. To learn C programming in Unix editor environment.
3. To learn shell script and sed concepts.
4. To learn file management and permission advance commands.
5. To learn awk, grap, perl scripts.

Experiments:

1. Practice commands: cp, mv, rm, ln, ls, who, echo, cat, mkdir, rmdir. Wildcards (?, *) ,
    - I/O redirection (<, >, >>), pipelines (|)
2. Practice commands: xargs, alias, set-unset, setenv-unsetenv, export, source, ps, job, kill.
3. Practice commands: head, tail, cut, paste, sed, grep, sort, uniq, find , locate, chmod.
4. Writing a simple shell script to echo who is logged in.
5. Write a shell script to display only executable files in a given directory.
6. Write a shell script to sort a list of file either in alphabetic order or largest file first according to user response.

7. Write a shell script to count the lines. Words and characters in its input (Note : Don't use wc).
8. Write a shell script to print end of a glossary file in reverse order using array. (Hint: use awk tail).
9. Modify cal command to accept more than one month (e.g. $cal Oct, Nov,)(Hint : use alias too)
10. Write a shell script to check whether Ram logged in, continue checking every 60 seconds until success.

Text Books

1. UNIX – Concepts and Applications, Sumitabha Das, 4th Edition, Tata McGraw Hill, 2006
2. Unix, concepts and applications by Sumitabha Das, McGraw-Hill
3. Mastering Shell Scripting, Randal. K. Michael , Second Edition, Wiley Publication

.

Reference Books

- UNIX and Shell Programming, Behrouz A. Forouzan and Richard F.Gilberg, Thomson, 2005. 2. UNIX & Shell Programming, M.G. Venkateshmurthy, Pearson Education, 2005.
- Unix Shell Programming by Yashwant Kanetkar.
- Unix shell programming by forozun.

Learning Outcome

1. Ability to understand the Unix Operating System and the working of the built in commands available in unix.
2. Analyze the working of the user defined commands and will be able change the permissions associated with files.
3. Understanding the concept of Shell and the different usage of the commands in shell. CO 4: Ability to program in AWK language
4. Identify and analyze various perl programs and administrator privileges

NETWORK LAB

Course/Paper: 06BCS-202
BCS Semester VI

Prerequisite: C programming Language

Lab Objectives: Students will try:

1. To get familiar with the basic network administration commands.
2. To install and configure network simulator and learn basics of TCL scripting.
3. To understand the network simulator environment and visualize a network topology and observe its performance
4. To analyze the traffic flow and the contents of protocol frames.
5. To implement client-server socket programs. 6. To design and configure a network for an organization.

Experiments:

1. The lab is to be conducted in Perl programming language, Perl works on all platforms (including windows)
2. Write few basic programs of Perl.

a. A Hello World Program
b. Write a program to add to 10 numbers.
c. Write a program of reading input from the keyboard and displaying them on monitor.
d. Write a program to take two strings as input and compare them
3. To understand advance constructs of Perl
e. Write a program to create a list of your course (all theory courses in current semester) using array and print them.
f. Write a program to accept ten number, store it into a hash table (Perl have itself) and when asked by user tell him that
number exists or not. (do not store duplicate numbers)
g. Write a program to compute the number of lines in a file.

4. Find the IP address of a host or turn an IP address into a name.
5. Connect to an FTP server and get or put files. Automate the one-time transfer of many files to download the file everyday,
which have changed since yesterday. (use Net:FTP)
6. Write a program to send mail. The programs should monitor system resources like disk space and notify admin by mail when
disk space becomes dangerously low. (use Net:mail)
7. Fetch mail from a POP3 server (use Net:pop3)
8. Find out who owns a domain (use Net:whois , Whois is a service provided by domain name registration authorities to identify
owners of domain names)
9. Test whether a machine is alive. machine can be specified using IP address or domain name of machine.
10. You have a URL that fetch its content from a Perl script, convert it to ASCII text (by stripping html tags) and display it.
11. Writing a TCP Client, Writing a TCP Server and Communicate some data over TCP

Text Books:

1. Computer Network: Top Down approach, Behrouz Forouzan, Firoz Mossharraf. MGH
2. Packet analysis with Wire shark, Anish Nath, PACKT publishing
3. Introduction to Network Simulator NS2, 2nd Edition, Teerawat Issariyakul, Ekram Hossain, Springer

Reference Book

NS2.34 Manual

Lab Outcomes: Students will able to:

1. Student will execute and evaluate network administration commands and demonstrate their use in different network scenarios
2. Students will demonstrate the installation and configuration of network simulator.
3. Students will demonstrate and asses different network scenarios and their performance behavior.
4. Students will analyze the contents the packet contents of different protocols.
5. Student will implement the socket programming for client server architecture.
6. Student will design and setup a organization network using packet tracer.

WEB PROGRAMING LAB

Course/Paper: 06BCS-203
BCS Semester VI

Prequisite : Basic understanding of programming fundamentals

Lab Objective

The course introduces you to the basic concepts of the World Wide Web, and the principles and tools that are used to develop Web applications. The course will provide an overview of Internet technology and will introduce you to current Web protocols, client side and server side programming, communication and design.

Experiments:

1. Develop a static html page using style sheet to show your own profile.
   Add a page to show 5 photos and
   Add a page to show your academics in a table
   Add a page containing 5 links to your favorite website
   Add navigational links to all above pages (add menu).
2. Update your homepage, by creating few html file (e.g. header, footer, left-sidebar, right), in these file you will put all html code to be shown on every page.
3. Use Cascading Style Sheets to format your all pages in a common format.
4. Basic Php programs:
   Write a simple "hello word" program using php.
   Write a program to accept two strings (name and age) from user. Print welcome statement e.g. " Hi Ram, yourage is 24."
   Write a program to create a calculator, which can support add, subtraction and multiply and division operation.
   Write a program to take input parameters for a table (no. of rows and no. of columns), and create the desired
   table.
   Create a "Contact Me" page -
   Ask user to enter his name, email ID,
   Use Java-Script to verify entered email address.
   Store submitted value in a MySql database.
   Display latest 5 submitted records in contact me page.
   Display above record with navigation support. e.g. (next, previous, first, last).

Text Books :

1. Web Technologies, Black Book, dreamtech Press
2. HTML 5, Black Book, dreamtech Press
3. Learning PHP, MySQL, JavaScript, CSS and HTML 5, Robin Nixon, O'Reilly publication

Reference Books:

- Developing Web Applications in PHP and AJAX, Harwani, McGrawHill
- Professional PHP Programming, Jesus Caspagnetto, Etal. Wrox Publication.
- Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel, Pearson
- Developing Web Applications, Ralph Moseley and M. T. Savaliya, Wiley-India

Learning Outcomes

Upon successful completion of this course you should be able to:

- Define modern protocols and systems used on the Web (such as HTML, HTTP, URLs, CSS, XML
- explain the functions of clients and servers on the Web, and describe the strengths and weaknesses of the client-server internet approaches to web design and implementation
- program, access, and manipulate data through the adoption of accepted standards, mark-up languages, client-side programming, and server-side programming
- design and implement an interactive web site(s) with regard to issues of usability, accessibility and internationalization
- design and implement a client-server internet application that accommodates specific requirements and constraints, based on analysis, modelling or requirements specification
- justify and explain particular internet application concepts, relevant alternatives and decision recommendations, including design considerations for internet security

MICROCONTROLLER LAB

Course/Paper: 06BCS-204
BCS Semester VI

Prerequisite:

The students should have good background on digital circuits (should have attended the course Switching Circuits and Logic Design.

Lab Objectives:

By the end of the course,

1. Students will understand difference between microprocessor and microcontroller and basics of embedded System.
2. Students will be able to apply the principles of logic design in understanding architecture and memory organization
3. Students will be able to understand assembly language and high level language programming concepts which will give an ability to analyze a problem, critical thinking, and identify and define the computing requirements appropriate to its solution.
4. Students will understand different peripherals and their interfacing concepts with microcontroller
5. Students will be are able to positively and appropriately apply knowledge in doing open ended project

Experiments:

1. Write a program to add two 2-byte numbers with a 3-byte sum.
2. Write a program to add an array of 8 numbers using loop.
3. Write a program to convert temperature from Fahrenheit to Centigrade.
4. Implement a sequencer traffic light controller.
1. 5-6. Implement real time interrupt.
2. 7-8. Interface microcontroller with stepper motor and move motor by given steps.
3. 9-10. Interface, test and control LED display with Microcontroller.
4. 11-12. Implement a watchdog timer and test the same to check infinite loop.

Text Books:

1. Gaonkar R., Microprocessor Architecture, Programming, and Applications with the 8085, Penram
2. Pal A., Microprocessors: Principles and Applications, TMH
3. Ayala K. J., The 8051 MicrocontrollerArchitecture, Programming & Applications, Penram.

Reference Books

- Mazidi and Mazidi, Microcontroller and Embedded Systems, Pearson Education
- Kapadia, R., 8051 Microcontroller and Embedded Systems, Jaico

Learning Outcomes

1. Ability to apply knowledge of mathematics, engineering to understand concepts in microcontroller based system
2. Ability to analyze a problem and formulate appropriate computing solution for microcontroller based applications.
3. An ability to design experiments in microcontrollers l analyze computer based process to meet desired needs
4. Ability to work, document and present as an individual and as a team-member to design formulate and implement experiments using modern tools.

VII SEMESTER

COMPILER CONSTRUCTION

Course/Paper: 07BCS-101
BCS Semester VII

Prerequisite:
Basic understanding of machine architecture, linkers, and OS calls at the ABI level. A willingness to get down and code. Good programming skills and ability to reason well. A deep understanding of the language you want to compile. Getting over the idea that *parsing* is all there is to compilers.

Course Objective:
The course is intended to teach the students the basic techniques that underlie the practice of Compiler Construction. The course will introduce the theory and tools that can be standarly employed in order to perform syntax-directed translation of a high-level programming language into an executable code.

Unit	Content
I	Compiler, Translator, Interpreter definition, Phase of compiler introduction to one pass & Multipass compilers, Bootstrapping, Review of Finite automata lexical analyzer, Input, buffering, Recognition of tokens, Idea about LEX: A lexical analyzer generator, Error handling.
II	Review of CFG Ambiguity of grammars, Introduction to parsing. Bottom up parsing Top down parsing techniques, Shift reduce parsing, Operator precedence parsing, Recursive descent parsing predictive parsers. LL grammars & passers error handling of LL parser. LR parsers, Construction of SLR, Conical LR & LALR parsing tables, parsing with ambiguous grammar. Introduction of automatic parser generator: YACC error handling in LR parsers.
III	Syntax directed definitions; Construction of syntax trees, L-attributed definitions, Top down translation. Specification of a type checker, Intermediate code forms using postfix notation and three address code, Representing TAC using triples and quadruples, Translation of assignment statement. Boolean expression and control structures.
IV	Storage organization, Storage allocation, Strategies, Activation records, Accessing local and non local names in a block structured language, Parameters passing, Symbol table organization, Data structures used in symbol tables.
V	Definition of basic block control flow graphs, DAG representation of basic block, Advantages of DAG, Sources of optimization, Loop optimization, Idea about global data flow analysis, Loop invariant computation, Peephole optimization, Issues in design of code generator, A simple code generator, Code generation from DAG.

Textbooks:
1. **Neetu choudhary/Deepika Sharma** "Compiler Design", Genius publication.
2. **Neelam Sharma/Yetendra Sharma** "Compiler Design", Ashirwad publication

References:

- **Aho, Sethi & Ullman**, "Compiler Design", **Addision Wesley**.
- **Adesh kumar pandey "**compiler concept", **KATSON.**
- .**A.A. Puntanbekar** "Compiler Design",Technical Publication Pune.

Learning Outcome:

Students have a sound mental model of how an abstract high level language is systematically transformed into a low-level machine language or byte code. Students are able to use that understanding to write better application code. Students are able to synthesis algorithms and techniques from compiler constructor to other problem domains.

DATA MININIG AND WAREHOUSING

Course/Paper: 07BCS-102
BCS Semester VII

Prerequisite: Knowledge of DBMS

Course Objective:

Students will be enabled to understand and implement classical models and algorithms in data warehousing and data mining. They will learn how to analyze the data, identify the problems, and choose the relevant models and algorithms to apply. They will further be able to assess the strengths and weaknesses of various methods and algorithms and to analyze their behavior.

Unit	Content
I	Overview, Motivation(for Data Mining),Data Mining-Definition & Functionalities, Data Processing, Form of Data Preprocessing, Data Cleaning: Missing Values, Noisy Data, (Binning, Clustering, Regression, Computer and Human inspection), Inconsistent Data, Data Integration and Transformation. Data Reduction:-Data Cube Aggregation, Dimensionality reduction, Data Compression, Numerosity Reduction, Clustering, Discretization and Concept hierarchy generation.
II	Concept Description:- Definition, Data Generalization, Analytical Characterization, Analysis of attribute relevance, Mining Class comparisons, Statistical measures in large Databases. Measuring Central Tendency, Measuring Dispersion of Data, Graph Displays of Basic Statistical class Description, Mining Association Rules in Large Databases, Association rule mining, mining Single-Dimensional Boolean Association rules from Transactional Databases– Apriori Algorithm, Mining Multilevel Association rules from Transaction Databases and Mining Multi- Dimensional Association rules from Relational Databases.
III	What is Classification & Prediction, Issues regarding Classification and prediction, Decision tree, Bayesian Classification, Classification by Back propagation, Multilayer feed-forward Neural Network, Back propagation Algorithm, Classification methods K-nearest neighbor classifiers, Genetic Algorithm. Cluster Analysis: Data types cluster analysis, Categories of clustering methods, Partitioning methods. Hierarchical Clustering- CURE and Chameleon. Density Based Methods-DBSCAN, OPTICS. Grid Based Methods- STING, CLIQUE.

	Model Based Method –Statistical Approach, Neural Network approach, Outlier Analysis
IV	Data Warehousing: Overview, Definition, Delivery Process, Difference between Database System and Data Warehouse, Multi Dimensional Data Model, Data Cubes, Stars, Snow Flakes, Fact Constellations, Concept hierarchy, Process Architecture, 3 Tier Architecture, Data Marting.
V	Aggregation, Historical information, Query Facility, OLAP function and Tools. OLAP Servers, ROLAP, MOLAP, HOLAP, Data Mining interface, Security, Backup and Recovery, Tuning Data Warehouse, Testing Data Warehouse.

Textbooks:

1. **Mallach,**"Data Warehousing System", **TMH**
2. **Gajendra Sharma,** Data Mining And Warehousing, **Katson**
3. M. Golfarelli, S. Rizzi. Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill, 2009.

References:

- **M.H. Dunham**,"Data Mining:Introductory and Advanced Topics" **Pearson Education**
- **Jiawei Han, Micheline Kamber**, "Data Mining Concepts & Techniques" **Elsevier**
- **Sam Anahory, Dennis Murray**, "Data Warehousing in the Real World : A Practical Guide for Building Decision Support Systems, 1/e " **Pearson Education**

Learning outcome:

Evaluate systematically supervised & unsupervised models and algorithms With respect to their accuracy. Develop practical work of Data Management techniques & design hypothesis based on the analysis to conceptualize a Data Management solution to a practical problem.

LOGIC SYNTHESIS

Course/Paper: 07BCS-103
BCS Semester VII

Prerequisite:

Binary logic, synthesis and analysis of combinational circuits. concept and methods of digital computer arithmetic. The fundamental principles of sequential circuits & acquaintance to novel design techniques and implementation technologies of digital systems.

Course Objective:

The goal of this course is to provide the student with a detailed knowledge of different methods for logic representation, manipulation, and optimization, for both combinational and sequential logic. The course views logic synthesis in the context of the implementation styles that are popularly used in industry today. At the end of the course the student should be able to perform the design of digital systems from a sound theoretical and practical perspective and have access to and an understanding of a suite of powerful tools that can be applied to a wide variety of CAD for VLSI problems.

Unit	Content
I	Introduction to VLSI, circuits Asics and Moore's Law. Microelectronic Design, Styles, four phases in creating Microelectronics chips computer Aided Synthesis and Optimization. Algorithms Review of Graph Definitions and Notations Decision and Optimization Problems, Shortest and Longest Path Problems, Vertex Cover, Graph, Coloring, Clique covering and partitioning Algorithms Boolean Algebra and Representation of Boolean Functions, binary Decision diagrams. Satisfiability and cover problems.
II	Hardware Modeling: Introduction to Hardware Modeling Language, State Diagrams. Data flow and Sequencing Graphs. Compilation and Behavioral Optimization Techniques. Circuits Specifications for Architectural Synthesis Resources and constraints. Fundamental Architectural Synthesis Problems Temporal Domain Scheduling Spatial Domain Binding Hierarchical Models and Synchronization Problem. Area and performance estimation-Resource Dominated circuits and General Circuits.
III	Scheduling Algorithms: Model for Scheduling Problems, Scheduling without Resource, Constraints-Unconstrained Scheduling ASAP Scheduling Algorithms Latency. Constrained Scheduling. ALAP scheduling. Under Timing Constraints and Relative Scheduling with Resource Constraints Integer Linear Programming Model, Multiprocessor Scheduling, Heuristic Scheduling Algorithms (List Scheduling). Force Directed Scheduling.
IV	Two Level Combination Logic Optimization: Logic Optimization Principles-Definitions, Exact Logic Minimization, Heuristic, Logic Minimization, and Testability Properties Operations on Two level logic Cover-positional Cube Notation, Functions with Multivolume inputs and list oriented manipulation. Algorithms for logic minimization.
V	Sequential logic optimization: Introduction, Sequential circuit optimization using state based models- state minimization, state encoding. Sequential circuit optimization using network models. Implicit finite state machine traversal methods. Testability consideration for synchronous circuits.

Text books:

1. MORRIS MANO, M., CILETTI,
2. BROWN, ST.,VRANESIC, Z., Fun L Design, 3/e, McGraw-Hill Higher Education
3. KYRIAKIS-BITZAROS, E. D., Logic C oratory Manual M., Digital Design, 4/e, Prentice Hall. damentals of Digital Logic with VHD ircuit Design, Lab Recommended Books

Reference Books:

- MORRIS MANO, M., and KIME, C.R., Logic and Computer Design Fundamentals, Pearson Education, 4/e, 2008.
- GAJSKI D.D., Principles of Digital Design, Prentice Hall; 1/e, 1996.

Learning Outcome:

The students will be able to understand the functionality and applications of logic circuits. Design and simulate logic circuits using software CAD tools. Recognize and define the hardware required for synthesis and implementation of simple combinational and sequential circuits in terms of standard integrated circuits. Analyze, design and synthesize logic circuits for low complexity applications.

ARTIFICIAL INTELLIGENCE

Course/Paper: 07BCS-104
BCS Semester VII

Prerequisite:
Analysis and Design of Algorithms (an important skill to make efficient algorithms). Programming in C++ so that you understand the basics of structural program flow and object oriented programming. C++ is essentially used in the market even today because it is very fast and processor friendly.

Course Objectives:
The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches. Develop a basic understanding of the building blocks of AI as presented in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning. Students will implement a small AI system in a team environment. The knowledge of artificial intelligence plays a considerable role in some applications students develop for courses in the program.

Unit	Content
I	Meaning and definition of artificial intelligence, Various types of production systems, Characteristics of production systems, Study and comparison of breadth first search and depth first search. Techniques, other Search Techniques like hill Climbing, Best first Search. A* algorithm, AO* algorithms etc, and various types of control strategies.
II	Knowledge Representation, Problems in representing knowledge, knowledge representation using propositional and predicate logic, comparison of propositional and predicate logic, Resolution, refutation, deduction, theorem proving, inferencing, monotonic and non-monotonic reasoning.
III	Probabilistic reasoning, Baye's theorem, semantic networks scripts schemas, frames, conceptual dependency and fuzzy logic, forward and backward reasoning.
IV	Game playing techniques like minimax procedure, alpha-beta cut-offs etc, planning, Study of the block world problem in robotics, Introduction to understanding and natural languages processing.
V	Introduction to learning, Various techniques used in learning, introduction to neural networks, applications of neural networks, common sense, reasoning, some example of expert systems.

Textbooks:
1. Introduction to Artificial Intelligence, Rajendra Akerkar; Prentice Hall of India, 2005.
2. **Archana Jain, Mukesh Verma**, Artificial Intelligence & Expert System, **Ashirwad**
3. **Amit Konar**, Artificial Intelligence and Soft Computing, **CRC**

References:
- **Charnick** "Introduction to A.I.", **Addision Wesley**
- **Rich & Knight**, "Artificial Intelligence"
- **Elamie,** "Artificial Intelligence", **Academic Press**

Learning Outcome:
Given a real world supervised learning problem, choose and implement appropriate learning algorithms such as decision trees, support vector machines, and boosting. Implement and execute by hand alpha-beta search. Design good evaluation functions and strategies for game playing.

MULTIMEDIA SYSTEMS

Course/Paper: 07BCS-105
BCS Semester VII

Prerequisite:
Basic knowledge of Adobe CS4, Adobe Fireworks, Adobe Fireworks, Gimp, Google Sketchup, Microsoft Frontpage, Apple Quicktime, Photoshop Pro, Microsoft Powerpoint, Adobe Flash Player, Adobe Shockwave.

Course Objective:
The objective of this course is to provide students with a basic understanding of multimedia systems. This course focuses on topics in multimedia information representation and relevant signal processing aspects, multimedia networking and communications, and multimedia standards especially on the audio, image and video compression. All of these topics are important in multimedia industries.

Unit	Content
I	Introduction to Multimedia, Multimedia Information, Multimedia Objects, Multimedia in business and work. Convergence of Computer, Communication and Entertainment products and Stages of Multimedia Projects, Multimedia hardware, Memory & storage devices, Communication devices, Multimedia software's, presentation tools, tools for object generations, video, sound, image capturing, authoring tools, card and page based authoring tools.
II	Multimedia Building Blocks Text, Sound MIDI, Digital Audio, audio file formats, MIDI under windows environment Audio & Video Capture.
III	Data Compression Huffman Coding, Shannon Fano Algorithm, Huffman Algorithms, Adaptive Coding, Arithmetic Coding Higher Order Modeling. Finite Context Modeling, Dictionary based Compression, Sliding Window Compression, LZ77, LZW compression, Compression, Compression ratio loss less & lossy compression.
IV	Speech Compression & Synthesis Digital Audio concepts, Sampling Variables, Loss less compression of sound, loss compression & silence compression.
V	Images: Multiple monitors, bitmaps, Vector drawing, lossy graphic compression, image file formatic animations Images standards, JPEG Compression, Zig Zag Coding, Multimedia Database. Content based retrieval for text and images, Video: Video representation, Colors, Video Compression, MPEG standards, MHEG Standard Video Streaming on net, Video Conferencing, Multimedia Broadcast Services, Indexing and retrieval of Video Database, recent development in Multimedia

Textbooks:

1. **Sleinreitz** "Multimedia System" **Addison Wesley**.
2. Li Ze-Nian and Drew Mark S, Fundamentals of Multimedia, Pearson Prentice-Hall, 2004.

References:

- Tay Vaughan "Multimedia, Making IT Work" Osborne TMH.
- Buford "Multimedia Systems" Addison Wesley.
- Aagrawal & Tiwari "Multimedia Systems" Excel.

Learning Outcome:

Through this course, students are expected to achieve a basic understanding of multimedia systems. With such background equipment, students would be able to evaluate more advanced or future multimedia systems. This course will also arouse students' interest in the course and further motivate them towards developing their career in the area of multimedia and internet applications.

ELECTIVE:--

SERVICE ORIENTED ARCHITECHURE

Course/Paper: 07BCS-106.1
BCS Semester VII

Prerequisite:

Foundations of Web Design and Programming

Course Objectives:

The objective of this course is to offer an introduction to service-oriented architecture (SOA) addressing the essential concepts of SOA and to teach how to use Business Process Management (BPM) practices to meet the demands of a constantly changing environment.

Unit	Content
I	SOA Fundamentals: Defining SOA, Business Value of SOA, Evolution of SOA, SOA characteristics, concept of a service in SOA, misperceptions about SOA, Basic SOA architecture, infrastructure services, Enterprise Service Bus (ESB), SOA Enterprise Software models, IBM On Demand operating environment.
II	Web services Technologies: XML technologies – XML, DTD, XSD, XSLT, XQuery, XPath Web services technologies - Web services and SOA, WSDL, SOAP, UDDI WS Standards (WS-*) - Web services and Service oriented enterprise (SOE), WS-Coordination and WS-Transaction, Business Process Execution Language for Web Services (BPEL4WS), WS-Security and the Web services security specifications, WS-Reliable Messaging, WS Policy, WS-Attachments.
III	SOA Planning and Analysis: Stages of the SOA lifecycle, SOA Delivery Strategies, service-oriented analysis, Capture and assess business and IT issues and drivers, determining non-functional requirements (e.g., technical constraints, business constraints, runtime qualities, non-runtime qualities), business centric SOA and its benefits, Service modeling, Basic modeling building blocks, service models for legacy application integration and enterprise integration, Enterprise solution

	assets(ESA) .
IV	SOA Design and implementation: service-oriented design process, design activities, determine services and tasks based on business process model, choosing appropriate standards, articulate architecture, mapping business processes to technology, designing service integration environment (e.g., ESB, registry), Tools available for appropriate designing, implementing SOA, security implementation, implementation of integration patterns, services enablement, quality assurance.
V	Managing SOA Environment: Distributing service management and monitoring concepts, operational management challenges, Service-level agreement considerations, SOA governance (SLA, roles and responsibilities, policies, critical success factors, and metrics), QoS compliance in SOA governance, role of ESB in SOA governance, impact of changes to services in the SOA lifecycle.

Text Books

1. 1.Sang Shin, SOA Programming, http://www.javapassion.com/soaprogramming/
2. SOA Principles of Service Design by Thomas Erl
3. Service-Oriented Architecture (SOA): Concepts, Technology, and Design by Thomas Erl

Reference Books

- 1.SOA Design Patterns (The Prentice Hall Service-Oriented Computing Series from Thomas Erl) by Thomas Erl
- 2.SOA in Practice: The Art of Distributed System Design (Theory in Practice) by Nicolai M. Josuttis
- 3.Service Oriented Architecture (SOA) For Dummies, 2nd Edition (For Dummies (Computer/Tech)) by Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Fern Halper

Learning Outcome :

The students who succeeded in this course;

- Implement and operate modern Service-Oriented Architectures (SOA)
- Apply SOA technology to reduce application development time and improve business agility
- Describe the business processes, process models and how SOA supports process improvement
- Automate complex business processes using workflow visualisation and service orchestration
- Apply proven SOA standards to achieve platform interoperability and integration of legacy systems

OPTICAL COMMUNICATION

Course/Paper: 07BCS-106.2
BCS Semester VII
Prerequisite: NIL

Course Objectives:

1. To learn the basic elements of optical fiber transmission link, fiber modes configurations and Structures.
2. To understand the different kind of losses, signal distortion in optical wave guides and other signal degradation factors.
3. To learn the fiber optical receivers such as PIN APD diodes, noise performance in photo detector, receiver operation and configuration.
4. To learn the fiber optical network components, variety of networking aspects, FDDI, SONET/SDH and operational principles WDM.

Unit	Content
I	Introduction to optical communication principles of light transmission optical fiber modes and configurations, Mode theory for circular wave-guides, Single-mode fibers, Multimode fibers, Numerical aperture, Mode field diameter, Vnumber, fiber materials, Fiber fabrication techniques.
II	Optical sources, LED'S, LASER diodes, Model reflection noise, Power launching and coupling, population inversion, fiber splicing, optical connectors, Photo-detectors, PIN, Avalanche detector, Response time, Avalanche multiplication noise.
III	Signal degradation in optical fibers, Attenuation losses, Signal distortion in optical wave guides, Material dispersion, Wave guide dispersion, Chromatic dispersion, Inter-modal distortion, Pulse broadening in Graded index fibers, Mode coupling, Advance fiber designs: dispersion shifted, Dispersion flattened, Dispersion compensating fibers, Design optimization of single mode fibers.
IV	Coherent optical fiber communication, Modulation techniques for Homodyne and Heterodyne systems, Optical filter link design. Rise time budget and link power budget, Long haul systems bit error rate, line coding, NRZ, RZ, Block Codes eye pattern.
V	Advance system and techniques, wavelength division multiplexing, optical amplifiers semiconductor amplifier, EDFA, Comparison between semiconductor and optical amplifier, Gain band width, Photonic switching, Optical Networks. Optical fiber bus, Ring topology, Star architectures, FDDI, SON-ET.

Text books:

1. Gerd Keiser, "Optical Fiber Communication" McGraw -Hill International, Singapore, 3rd edition, 2000
2. Rajiv Ramaswami, Kumar N. Sivaranjan, "Optical Networks A practical perspective", 2nd edition, Elsevier, 2004.
3. J.Gowar, "Optical Communication System", 2nd edition, Prentice Hall of India, 2001.
4. Optical Fiber Communications: Principles and Practice (3rd Edition) by John Senior

Reference Books :

- Optical Communication Networks by Biswanath Mukherjee
- Introduction to Optical Communication, Lightwave Technology, Fiber Transmission, and Optical Networks by Lawrence Harte and David Eckard
- OFDM for Optical Communications by William Shieh and Ivan Djordjevic
- Optical Communications (Wiley Series in Telecommunications and Signal Processing) by Robert M. Gagliardi and Sherman Karp.

Learning Outcome:

1. Graduate will demonstrate the ability to design conduct experiment analyze and interpret data.
2. Graduate will demonstrate the ability to design a system, component or process as per needs and specification.
3. Graduate will develop confidence for self education and ability for life long learning.

REAL TIME SYSTEMS

Course/Paper: 07BCS-106.3
BCS Semester VII

Prerequisits: Knowledge of operating system (OS) basics: overview of OS architectures, OS classification, UNIX kernel knowledge, OS service call principles, shell. Context switching, multitasking. File systems, processes, virtual memory. Basic skills of programming in C.

Course Objective:
To deliver real-time-system theory. Real-time scheduling theory. To have solid experiences on real-time scheduler implementation. Working of system under (explicit or implicit) timing constraints is a real-time system.

Unit	Content
I	Introduction: Definition, Typical Real Time Applications: Digital Control, High Level Controls, Signal Processing etc., Release Times, Deadlines, and Timing Constraints, Hard Real Time Systems and Soft Real Time Systems, Reference Models for Real Time Systems: Processors and Resources, Temporal Parameters of Real Time Workload, Periodic Task Model, Precedence Constraints and Data Dependency.
II	Real Time Scheduling: Common Approaches to Real Time Scheduling: Clock Driven Approach, Weighted Round Robin Approach, Priority Driven Approach, Dynamic Versus Static Systems, Optimality of Effective-Deadline-First (EDF) and Least-Slack-Time-First (LST) Algorithms, Offline Versus Online Scheduling, Scheduling Aperiodic and Sporadic jobs in Priority Driven and Clock Driven Systems.

III	Resources Access Control: Effect of Resource Contention and Resource Access Control (RAC), Non-preemptive Critical Sections, Basic Priority-Inheritance and Priority-Ceiling Protocols, Stack Based Priority-Ceiling Protocol, Use of Priority-Ceiling Protocol in Dynamic Priority Systems, Preemption Ceiling Protocol, Access Control in Multiple-Unit Resources, Controlling Concurrent Accesses to Data Objects.
IV	Multiprocessor System Environment: Multiprocessor and Distributed System Model, Multiprocessor Priority-Ceiling Protocol, Schedulability of Fixed-Priority End-to-End Periodic Tasks, Scheduling Algorithms for End-to-End Periodic Tasks, End-to-End Tasks in Heterogeneous Systems, Predictability and Validation of Dynamic Multiprocessor Systems, Scheduling of Tasks with Temporal Distance Constraints.
V	Real Time Communication: Model of Real Time Communication, Priority-Based Service and Weighted Round- Robin Service Disciplines for Switched Networks, Medium Access Control Protocols for Broadcast Networks, Internet and Resource Reservation Protocols, Real Time Protocols, Communication in Multicomputer System, An Overview of Real Time Operating Systems.

Textbooks:

1. Abbott, D.: Linux for Embedded and Real-Time Applications. Newnes, 2002
2. Cheng, A. M. K.: Real-Time Systems: Scheduling, Analysis, and Verification. Wiley, 2002
3. Cottet, F., Delacroix, J., Kaiser, C., Mammeri, Z.: Scheduling in Real-Time Systems. John Wiley & Sons, 2002
4. Krishna, C. M., Shin, K. G.: Real-Time Systems. McGraw-Hill, 1997

References:

- Real Time Systems by **Jane W. S. Liu, Pearson Education Publication**.
- Real-Time Systems: Scheduling, Analysis, and Verification by **Prof. Albert M. K.Cheng, John Wiley and Sons Publications.**
- Real Time System, **Poonam Singh, Dhanpat Rai**

Learning Outcome:

Theoretical background (specification/verification) and practical knowledge of real-time operating systems. After completing the course students will appreciate the use of multitasking techniques in real-time systems, understand the fundamental concepts of real-time operating systems, understand the features and structures of practical implementations, appreciate how application areas (e.g. safety-critical, desktop, etc.) impact on real-time operating system facilities.

Laboratories:--

COMPILER DESIGN LAB

Course/Paper: 07BCS-201
BCS Semester VII

Prequisites:
For CS 432 F the prerequisite is: CS 232 F: Programming Languages. For CS 728: A Programming Languages course similar to CS 232 F. Other desired background: knowledge of automata theory, context free languages, computer architecture, data structures and simple graph algorithms, logic or algebra.

Lab Objective:
Develop an Understanding of Intermediate code form: Three address code, Polish notation. Develop an Understanding of Allocation data structure & Heaps. Develop an Understanding of Scanning by using concept of Finite state automaton.

Experiments:
1. Develop a lexical analyzer to recognize a few patterns in PASCAL and C.
    a. (ex: identifiers, constants, comments, operators etc.)
2. Write a program to parse using Brute force technique of Top down parsing.
3. Develop on LL (1) parser (Construct parse table also).
4. Develop an operator precedence parser (Construct parse table also)
5. Develop a recursive descent parser.
6. Write a program for generating for various intermediate code forms
    a. i) Three address code ii) Polish notation
7. Write a program to simulate Heap storage allocation strategy
8. Generate Lexical analyzer using LEX
9. Generate YACC specification for a few syntactic categories.
10. Given any intermediate code form implement code optimization techniques

Textbooks:

1. modern compiler implementation in ML by A. W. Appel. Cambridge University Press, 1997
2. Compilers: Principles, Techniques and Tools by A. V. Aho, R. Sethi, J. D. Ullman. Addison-Wesley, 1986.

Learning Objective:
Implement a fully functional compiler for a subset of the Java language that targets an embedded microprocess. Apply patterns and common data structures to compiler constructrion including tree traversal, graph algorithms, and proof systems.

DATA MINING AND WAREHOUSING LAB

Course/Paper: 07BCS-202
BCS Semester VII

Perquisites: knowledge of DBMS

Lab objective: Data warehousing programs teach students how to stage, integrate, and access information. Programs expose students to information applications for specific industries, such as tracking inventory and working reservations in hotel and tourism fields, managing the supply chain in manufacturing industries, and identifying market variables in global economies.

Experiments:
The objective of the lab exercises is to use data mining techniques to use standard databases available to understand DM processes using any DM tool)

- **Gain insight for running pre- defined decision trees and explore results using MS OLAP Analytics.**
- **Using IBM OLAP Miner – Understand the use of data mining for evaluating the content of multidimensional cubes.**
- **Using Teradata Warehouse Miner – Create mining models that are executed in SQL. (** Portal work : The objective of this lab exercises is to integrate pre-built reports into a portal application)
- **Publish and analyze a business intelligence portal.** Metadata & ETL Lab: The objective of this lab exercises is to implement metadata import agents to pull metadata from leading business intelligence tools and populate a metadata repository. To understand ETL processes
- **Import metadata from specific business intelligence tools and populate a meta data repository.**
- **Publish metadata stored in the repository.**
- **Load data from heterogeneous sources including text files into a pre-defined warehouse schema.**
- **Case study**
- Design a data mart from scratch to store the credit history of customers of a bank. Use this credit profiling to process
- future loan applications.
- Design and build a Data Warehouse using bottom up approach titled 'Citizen Information System'.

Textbooks:

1. **Mallach,**"Data Warehousing System", **TMH**
2. **Gajendra Sharma,** Data Mining And Warehousing, **Katson**
3. M. Golfarelli, S. Rizzi. Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill, 2009.

Learning Outcome:
At the end of this course, you will understand the main concepts, principles and techniques of data ware housing and data mining. For practical work you will be using a popular data mining package to analyse data of various forms, including transaction data, relational data and textual data and up-to-date conceptual and practical knowledge on recent developments in data mining.

LOGIC SYNTHESIS LAB

Course/Paper: 07BCS-203
BCS Semester VII

Prerequisites:
Lab Objective:

1. Write a program which reads simple digital circuit (of size up to 10 gates) in blif / Boolean equation and display
1. Schematic in graphics format.
2. Write a program to convert Blif format into Boolean equation.
3. Write a program that estimate area of circuit (specified as Blif or Boolean equation) using library binding technique of simple circuit (up to 10 gates).
4. Write a program to implement state machine up to 5 states.
5. Write a program to count 4-input lookup table in a simple circuit (up to 10 gates specified as Blif or Boolean equation).
6. Write a program to obtain sequencing graph for a given set of arithmetic expression (up to 10 nodes)
7. Write VHDL Codes for all gates with all Modeling.
8. Write VHDL Codes & Test bench for half adder and full adder

Textbooks:

1. W.H. ,loyner, Jr., "Logic Synthesis", Proceedings of Comp-Euro "87, Hamburg, Germany, 1987.
2. A.R. Newton, "Techniques for Logic Synthesis", 1985.
3. A.L. Sangiovanni-Vincentelli, "An Overview of Synthesis Systems", IEEE Custom Integrated Circuits Conference, Portland, Oregon, 1985.

Learning outcome:
The students will be able to understand the functionality and applications of logic circuits. Design and simulate logic circuits using software CAD tools. Recognize and define the hardware required for synthesis and implementation of simple combinational and sequential circuits in terms of standard integrated circuits. Analyze, design and synthesize logic circuits for low complexity applications

PROJECT STAGE-I

Course/Paper: 07BCS-204
BCS Semester VII

Here students have to submit synopsis and then a detailed analysis of their project with literature survey.

Objective

- Knowledge and application of research methods appropriate to the field of study
- Critical thinking and problem solving skills
- Application of knowledge and skills to plan and execute a substantial research project

- Communication skills to justify and interpret theoretical propositions, methodologies, conclusions and decisions to technical and non technical audiences
- Application of skills and knowledge with personal autonomy and accountability

This course is particularly important in the development of graduate attributes related to innovation, work readiness, global outlook, environmental awareness, social awareness and life long learning.

Learning Outcome

After completion of this course, student will be able to:

1. Conduct an independent research project under supervision
2. Adhere to responsible laboratory or field practice regarding data collection and recording, and laboratory/field safety
3. Demonstrate time and project management in the successful identification of a research project, development of an experimental design, collection of accurate and precise data, critical analysis and interpretation of results, retrieval of information, and critical reading of scientific literature
4. Present a seminar on the results of a research project.

PRACTICAL TRAINING SEMINAR

Course/Paper: 07BCS-205
BCS Semester VII

Here students have to submit training reports as well as they are required to give a brief presentation of there training in external and internal.

About Industrial Practical Training Program

Practical training is a mandatory course for ME (credited) and called Practical Training/ Summer Training course. This course has to be taken by the college of engineering students in order to graduate; Majors that falls under this group are:

Goals of Practical Training

- Expose the CoE students to the methods of engineering practice.
- Identify the strengths and weaknesses in current students through industry feedback and modify the curriculum according to the needs.
- Strengthen the relation between industrial institutions and Qatar University Benefits of the industrial training program.

To The Students

1. Familiarize the students with practical engineering work in various disciplines.
2. Improve the performance of the students in theory classes by introducing them to the in practical work.
3. Help to know the strengths and weaknesses in the students, so they can improve their skills and overcome their limitations by taking appropriate measures.
4. Expose students to real work situations and equip them with the necessary skills so that they would be ready for the job when they graduate.

VIII SEMESTER

INFORMATION SYSTEM AND SECURITIES

Course/Paper: 08BCS-101
BCS Semester VIII

Prerequisite:
Familiar with Unix/Windows, monitoring, and securities systems. Microsoft Office suite, network operating systems, SQL, LAN.

Course Objective:
The academic aim is to prepare students with the skills, knowledge, cutting-edge technologies, practices to be security architects capable of designing, implementing, analyzing and managing the security of real-life information systems.

Unit	Content
I	Introduction to security attacks, services and mechanism, introduction to cryptography. Conventional Encryption: Conventional encryption model, classical encryption techniques- substitution ciphers and transposition ciphers, cryptanalysis, stereography, stream and block ciphers. Modern Block Ciphers: Block ciphers principals, Shannon's theory of confusion and diffusion, fiestal structure, data encryption standard(DES), strength of DES, differential and linear crypt analysis of DES, block cipher modes of operations, triple DES, IDEA encryption and decryption, strength of IDEA, confidentiality using conventional encryption, traffic confidentiality, key distribution, random number generation.
II	Introduction to graph, ring and field, prime and relative prime numbers, modular arithmetic, Fermat's and Euler's theorem, primality testing, Euclid's Algorithm, Chinese Remainder theorem, discrete logarithms. Principals of public key crypto systems, RSA algorithm, security of RSA, key management, Diffle-Hellman key exchange algorithm, introductory idea of Elliptic curve cryptography, Elganel encryption.
III	Message Authentication and Hash Function: Authentication requirements, authentication functions, message authentication code, hash functions, birthday attacks, security of hash functions and MACS, MD5 message digest algorithm, Secure hash algorithm(SHA). Digital Signatures: Digital Signatures, authentication protocols, digital signature standards (DSS), proof of digital signature algorithm.
IV	Authentication Applications: Kerberos and X.509, directory authentication service, electronic mail security-pretty good privacy (PGP), S/MIME.
V	IP Security: Architecture, Authentication header, Encapsulating security payloads, combining security associations, key management. Web Security: Secure socket layer and transport layer security, secure electronic transaction (SET). System Security: Intruders, Viruses and related threads, firewall design principals, trusted systems.

Textbooks:

1. Security Policies Management of Information Security Third Ed. By: Michael E. Whitman Computer Ethics & Professional Responsibility
2. Web and Database Security by Jiping Xiong, Lifeng Xuan, Jian Zhao and Tao Huang

Reference book:

- **William Stallings**, "Cryptography and Network Security: Principals and Practice", **Prentice Hall, New Jersy.**
- **Johannes A. Buchmann,** "Introduction to Cryptography", **Springer-Verlag**.
- **Bruce Schiener**, "Applied Cryptography".

Learning Outcome:

The ability to analyze, design, program, implement, secure and maintain network applications. Design, implement, test and document object-oriented software systems. Install, configure, build, troubleshoot, secure, modify and maintain computer system architectures and networks to meet user requirements.

CAD FOR VLSI DESIGN

Course/Paper: 08BCS-102
BCS Semester VIII

Prerequisites:

Logic design, spanning combinational and sequential logic. Analog electronic circuits; transistor-level circuit design. Familiar yourself with scripting and using UNIX.

Course Objective:

The objective of this course is to help students develop in-depth analytical and design capabilities in digital CMOS circuits and chips. The development of VLSI chips requires an interdisciplinary team of architects, logic designers, circuit and layout designers, packaging engineers, test engineers, and process device engineers. Also essential are the computer aids for design automation and optimization.

Unit	Content
I	Modern digital systems, complexity and diversity of digital systems, productivity gap and need for CAD tools. Introduction to steps and CAD flow for designing with ASIC and FPGA.
II	Introduction to VHDL, background, VHDL requirement, Elements of VHDL, top down design, convention and syntax, basic concepts in VHDL i.e. characterizing H/W languages, objects, classes, and signal assignments.
III	Structural specification of H/W- Parts library, Wiring, modeling, binding alternatives, top down wiring. Design organization and parameterization. Type declaration, VHDL operators.
IV	VHDL subprogram parameters, overloading, predefined attributes, user defined attributes, packaging basic utilities. VHDL as a modeling language- bi-directional component modeling, multi mode component modeling,
V	Examples of VHDL synthesis subsets- combinational logic synthesis, sequential circuit synthesis, state machine synthesis. VHDL language grammar. Introduction to synthetic circuits and circuit repositories.

Textbooks:

1. Computer Aids for VLSI Design Second Edition *Steven M. Rubin*
2. Design Through Verilog HDL by T. R. Padmanabhan and B. Bala Tripura Sundari

3. "Digital VLSI Chip Design with Cadence and Synopsys CAD Tools" by Erik Brunvand

Reference books:

- "VLSI Physical Design: From Graph Partitioning to Timing Closure" by Andrew B. Kahng and Jens Lienig
- "Digital Timing Macromodeling for VLSI Design Verification (The Springer International Series in Engineering and Computer Science)" by Jeong-Taek Kong and David V. Overhauser
- "Evolutionary Algorithms for VLSI CAD" by Rolf Drechsler

Learning Outcome:
The students will be bale to understand the concept behind ASIC (Application Specific Integrated Circuits) design and the different implementation approaches used in industry. Implement a complete digital system on silicon using state-of-the art CAD tools.

ADVANCED COMPUTER ARCHITECTURES

Course/Paper: 08BCS-103
BCS Semester VIII

Prerequisite:
Basic notions of computer architecture including programs, instruction sets, simple processor design control unit, memories (cache & virtual) and input /output interface. Understanding of the interaction amongst architecture, applications and technology. Understanding of a framework for evaluating design decisions in terms of application requirements and performance measurements.

Course Objective:

1. To identify and analyze the attributes of computer architecture design with recent trend technology
2. To identify the techniques to improve the speed and performance of computers – Parallelism in Instruction level – Hardware approaches - pipelining,dynamic scheduling, superscalar processors, and multiple issue of instructions.
3. To implement the design aspects and categorize various issues , causes and hazards due to parallelisms
4. To examine and compare the performance with benchmark standards
5. Its important to understand and apply that design principles in real systems.

Unit	Content
I	INTRODUCTION**:** Parallel Computing, Parallel Computer Model, Program and Network Properties, Parallel Architectural Classification Schemes, Flynn's & Feng's Classification, Performance Metrics and Measures, Speedup Performance Laws: Multiprocessor System and Interconnection Networks; IEEE POSIX Threads: Creating and Exiting Threads, Simultaneous Execution of Threads, Thread Synchronization using Semaphore and Mutex, Canceling the Threads.
II	PIPELINING AND MEMORY HIERARCHY: Basic and Intermediate Concepts, Instruction Set Principle; ILP: Basics, Exploiting ILP, Limits on ILP;

	Linear and Nonlinear Pipeline Processors; Super Scalar and Super Pipeline Design; Memory Hierarchy Design: Advanced Optimization of Cache Performance, Memory Technology and Optimization, Cache Coherence and Synchronization Mechanisms.
III	THREAD AND PROCESS LEVEL PARALLEL ARCHITECTURE: Introduction to MIMD Architecture, Multithreaded Architectures, Distributed Memory MIMD Architectures, Shared Memory MIMD Architecture, Clustering, Instruction Level Data Parallel Architecture, SIMD Architecture, Fine Grained and Coarse Grained SIMD Architecture, Associative and Neural Architecture, Data Parallel Pipelined and Systolic Architectures, Vector Architectures.
IV	Parallel Algorithms: PRAM Algorithms: Parallel Reduction, Prefix Sums, Preorder Tree Traversal, Merging two Sorted lists; Matrix Multiplication: Row Column Oriented Algorithms, Block Oriented Algorithms; Parallel Quicksort, Hyper Quick sort; Solving Linear Systems: Gaussian Elimination, Jacobi Algorithm; Parallel Algorithm Design Strategies.
V	Developing Parallel Computing Applications: OpenMP Implementation in 'C': Execution Model, Memory Model; Directives: Conditional Compilation, Internal Control Variables, Parallel Construct, Work Sharing Constructs, Combined Parallel Work-Sharing Constructs, Master and Synchronization Constructs; Run-Time Library Routines: Execution Environment Routines, Lock Routines, Timing Routines; Simple Examples in 'C'. Basics of MPI.

Textbooks:

1. **ADVANCED COMPUTER ARCHITECTURES Kai** Hwang Tata McGraw-Hill Education
2. **Computer Architecture : A Quantitative Approach by Hennessy**

Reference books:

- **Kai Hwang**," Advance Computer Architecture", **TMH**
- **Hennessy and Patterson**," Computer Architecture: A Quantitative Approach", **Elsevier**
- **Dezso and Sima,** "Advanced Computer Architecture", **Pearson**
- M.J. Flynn, " Computer Architecture: Pipelined and Parallel Processor Design", Narosa Publishing House/Jones

Learning Outcome:

On successful completion of this course will be able to describe the principles of computer design and classify instructions set architecture. Describe the operations of performance such as pipelines, dynamic scheduling branch predictions, caches. Describe the operations of virtual memory. Describe the modern architecture such as RISC, Scalar, VLIW Multi core and multi CPU systems.

ELECTIVE:--

DISTRIBUTED SYSTEMS

Course/Paper: 08BCS-104.1
BCS Semester VIII

Prerequisite: Prerequisite knowledge:

- Java programming
- Operating systems concepts (threads, processes, mutual exclusion, deadlock)
- Computer networking concepts (Internet, protocols, sockets, network application programming)

Course Objective:
The subject aims to provide an understanding of the principles on which the Web, Email, DNS and other interesting distributed systems are based. Questions concerning distributed architecture, concepts and design; and how these meet the demands of contemporary distributed applications will be addressed.

Unit	Content
I	CHARACTERIZATION OF DISTRIBUTED SYSTEMS: Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges. System Models: Architectural models, Fundamental Models Theoretical Foundation for Distributed System: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport's & vectors logical clocks, Causal ordering of messages, global state, termination. Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non token based algorithms, performance metric for distributed mutual exclusion algorithms.
II	DISTRIBUTED DEADLOCK DETECTION: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms. Agreement Protocols: Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem, Atomic Commit in Distributed Database system.
III	DISTRIBUTED OBJECTS AND REMOTE INVOCATION: Communication between distributed objects, Remote procedure call, Events and notifications, Java RMI case study. SECURITY: Overview of security techniques, Cryptographic algorithms, Digital signatures Cryptography pragmatics, Case studies: Needham Schroeder, Kerberos, SSL & Millicent. DISTRIBUTED FILE SYSTEMS: File service architecture, Sun Network File System, The Andrew File System, Recent advances.
IV	TRANSACTIONS AND CONCURRENCY CONTROL: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control. DISTRIBUTED

	TRANSACTIONS: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault - tolerant services, highly available services, Transactions with replicated data.
V	DISTRIBUTED ALGORITHMS: Introduction to communication protocols, Balanced sliding window protocol, Routing algorithms, Destination based routing, APP problem, Deadlock free Packet switching, Introduction to Wave & traversal algorithms, Election algorithm. CORBA CASE STUDY: CORBA RMI, CORBA services.

Textbooks:

1. "Distributed Algorithms" by Nancy Lynch
2. "Distributed Systems" by Jie Wu

References:

- **Coulouris, Dollimore, Kindberg,** "Distributed System: Concepts and Design", **Pearson Ed.**
- **Gerald Tel**, "Distributed Algorithms", **Cambridge University Press**
- **William Stalling**, Distributed System, **Addision Wesley**

Learning outcome:

On completion of this subject the student is expected to: Have an understanding of the principles and paradigms underlying distributed software systems & Obtain experience developing distributed applications.

IMAGE PROCESSING

Course/Paper: 08BCS-104.2
BCS Semester VIII

Prerequisites:

Set theory, Partial Differential Equations (PDE), Probability and Statistics, Data Mining, Digital Signal Processing & Machine Learning and pattern recognition.

Course Objectives:

Introduction Digital Image Processing, Image Fundamentals and Human Visual Perception, Image Enhancement in Spatial Domain, Image Transforms, Image Enhancement in Frequency Domain & Image Restoration.

Unit	Content
I	Introduction and Fundamentals: Motivation and Perspective, Applications, Components of Image Processing System, Element of Visual Perception, A Simple Image Model, Sampling and Quantization. Image Enhancement Spatial Domain: Introduction; Basic Gray Level Functions – Piecewise-Linear Transformation Functions: Contrast Stretching; Histogram Specification; Histogram Equalization; Local Enhancement; Enhancement using Arithmetic/Logic Operations – Image Subtraction, Image Averaging; Basics of

	Spatial Filtering; Smoothing – Mean filter, Ordered Statistic Filter; Sharpening – The Laplacian.
II	Image Enhancement in Frequency Domain: Fourier Transform and the Frequency Domain, Basis of Filtering in Frequency Domain, Filters – Low-pass, High-pass; Correspondence Between Filtering in Spatial and Frequency Domain; Smoothing Frequency Domain Filters – Gaussian Low pass Filters; Sharpening Frequency Domain Filters – Gaussian High pass Filters; Homomorphic Filtering. Image Restoration: A Model of Restoration Process, Noise Models, Restoration in the presence of Noise only Spatial Filtering – Mean Filters: Arithmetic Mean filter, Geometric Mean Filter, Order Statistic Filters – Median Filter, Max and Min filters; Periodic Noise Reduction by Frequency Domain Filtering – Band pass Filters; Minimum Meansquare Error Restoration.
III	Color Image Processing: Color Fundamentals, Color Models, Converting Colors to different models, Color Transformation, Smoothing and Sharpening, Color Segmentation. Morphological Image Processing: Introduction, Logic Operations involving Binary Images, Dilation and Erosion, Opening and Closing, Morphological Algorithms – Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening.
IV	Registration: Introduction, Geometric Transformation – Plane to Plane transformation, Mapping, Stereo Imaging – Algorithms to Establish Correspondence, Algorithms to Recover Depth. Segmentation: Introduction, Region Extraction, Pixel-Based Approach, Multi-level Thresholding, Local Thresholding, Region-based Approach, Edge and Line Detection: Edge Detection, Edge Operators, Pattern Fitting Approach, Edge Linking and Edge Following, Edge Elements Extraction by Thresholding, Edge Detector Performance, Line Detection, Corner Detection.
V	Feature Extraction: Representation, Topological Attributes, Geometric Attributes. Description: Boundary-based Description, Region-based Description, Relationship. Object Recognition: Deterministic Methods, Clustering, Statistical Classification, Syntactic Recognition, Tree Search, Graph Matching.

Textboks:

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing, 2nd ed., (Prentice Hall 2002).

2. Digital Image Processing: An Algorithmic Introduction using Java by Wilhelm Burger and Mark James Burge

Reference:

- 1.Digital Image Processing (3rd Edition) by Rafael C. Gonzalez and Richard E. Woods
- 2.Practical Algorithms for Image Analysis with CD-ROM by Lawrence O'Gorman, Michael J. Sammon, and Michael Seul
- 3.The Image Processing Handbook by John C. Russ
- 4.Feature Extraction & Image Processing, Second Edition by Mark Nixon and Alberto S Aguado

Learning Outcome:
Understand image formation and the role human visual system plays in perception of gray and color image data. Get broad exposure to and understanding of various applications of image processing in industry, medicine, and defense. Learn the signal processing algorithms and techniques in image enhancement and image restoration.

NATURAL LANGUAGE PROCESSING

Course/Paper: 08BCS-104.3
BCS Semester VIII

Prerequisite:
Linear algebra, Artificial Intelligence and Neural Networks, Programming in any high level language, preferably python or Matlab (inbuilt libraries and functions available) & robability and Statistics

Course Objective:
To understand natural language processing and to learn how to apply basic algorithms in this field. To get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora. To conceive basics of knowledge representation, inference, and relations to the artificial intelligence.

Unit	Content
I	Introduction to Natural Language Understanding: The study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems, Linguistic Background: An outline of English syntax.
II	Introduction to semantics and knowledge representation, Some applications like machine translation, database interface.
III	Grammars and Parsing: Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Transition Network Grammars, Top-Down Chart Parsing. Feature Systems and Augmented Grammars: Basic Feature system for English, Morphological Analysis and the Lexicon, Parsing with Features, Augmented Transition Networks.
IV	Grammars for Natural Language: Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language, Handling questions in Context-Free Grammars. Human preferences in Parsing, Encoding uncertainty, Deterministic Parser.
V	Ambiguity Resolution: Statistical Methods, Probabilistic Language Processing, Estimating Probabilities, Part-of- Speech tagging, Obtaining Lexical Probabilities, Probabilistic Context-Free Grammars, Best First Parsing. Semantics and Logical Form, Word senses and Ambiguity, Encoding Ambiguity in Logical Form

Textbooks:
1. Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization by Peter Jackson and Isabelle Moulinier
2. Speech and Language Processing (2nd Edition) by Daniel Jurafsky and James H. Martin

Reference:

- Foundations of Statistical Natural Language Processing by Christopher D. Manning and Hinrich Schuetze
- Natural Language Processing with Python by Steve Bird, Ewan Klein, Edward Loper, and Bird Steven
- Natural Language Processing and Text Mining by Anne Kao and Steve R. Poteet

Learning Outcome:

The student will grow in confidence in their mathematical and statistical abilities. In particular, the student will understand the models, methods, and algorithms of statistical Natural Language Processing (NLP) for common NLP tasks, such as speech recognition, machine translation, spam filtering, text classification, and spell checking. The student will implement probabilistic models in code, estimate parameters for such models, and run meaningful experiments to validate such models.

Laboratories:--

INFORMATION SYSTEM AND SECURITIES LAB

Course/Paper: 08BCS-201
BCS Semester VIII
Prerequisites:

To be familiar with Familiar with Unix/Windows, monitoring, and securities systems. Microsoft Office suite, network operating systems, SQL, LAN.

Lab Objective:

To put the emphasis on helping each individual acquire the skills and knowledge necessary to maximize his or her potential as a researcher in information systems security.

List of Projects are as follows (Implement any one)

1. Shopping cart project using ADO.NET: This sample project has all basic features required for a shopping cart web site including Login, Registration, Add to Cart, Checkout etc. A good ASP.NET learning project using C#, ASP.NET, SQL Server.
2. Personal Assistant: This is a small project for managing personal details. Current version of this project support Address Book feature - Add, Edit and Manage contacts and addresses using VB.NET.
3. Address Book: This is a small project for managing contact details. This is a C# version of the 'Personal Assistant' project.
4. School Management System: This is a project for managing education institutes using C#.
5. Library Management System: This is an academic project for students using Java.
6. spider Alerts & Web services: This project communicates with web services and downloads Alerts from the web server using Java & XML.
7. Patient Information System: This software can be used to keep track of the patients' information and treatment details in a hospital or clinic. Some of the advanced features include patient consulting, lab information, billing etc using JSP, Servlet & JDBC.
8. Web based Address Book: This application can be used to keep track of your contacts/addresses. N Tier architecture is used to separate data layer, business layer and UI layers.

Textbooks:

1. Security Policies Management of Information Security Third Ed. By: Michael E. Whitman Computer Ethics & Professional Responsibility
2. **Web and Database Security** *by Jiping Xiong, Lifeng Xuan, Jian Zhao and Tao Huang*

References:

- **William Stallings**, "Cryptography and Network Security: Principals and Practice", **Prentice Hall, New Jersy.**
- **Johannes A. Buchmann,** "Introduction to Cryptography", **Springer-Verlag**.
- **Bruce Schiener**, "Applied Cryptography".

Learning Outcome:

Design, implement, test and document object-oriented software systems. Install, configure, build, troubleshoot, secure, modify and maintain computer system architectures and networks to meet user requirements

VLSI DESIGN LAB

Course/Paper: 08BCS-202
BCS Semester VIII

Prerequisites: A basic knowledge of digital logic design, RLC circuits.

Course Objectives*:*

At the completion of this course, a student is expected to be able to design and analyze digital circuits, understand transistor operations, circuit families, area-powerperformance analysis, layout design techniques, signal integrity analysis, memory design and clocking issues. Students are also expected to understand various design methodologies such as custom, semi-custom, standard cell, arrayed logic, sea-of-gates.

Simple Design exercises:

01 Half adder, Full adder, Subtractor Flip Flops, 4bit comparator.
02 Parity generator
03 Bit up/down counter with load able count
04 Decoder and encoder
05 8 bit shift register
06 8:1 multiplexer
07 Test bench for a full adder
08 Barrel shifter
09 N by m binary multiplier
10 RISC CPU (3bit opcode, 5bit address)

TOOLS :

Xilinx Tools/ Synopsis Tools/ Cadence Tools/ Model SIM/ Leonardo Spectrum Tools/VIS/SIS Tools to be used.

Text book(s) and/or required materials

1. J. Bhaskar, "VHDL Primer", 1st edition, BSP

2. Samir Palnitkar, "Verilog HDL - Guide to Digital Design and Synthesis", 3rd Edition, Pearson Education, 2003
3. Lab Manual

Learning Outcome :
This course provides the design of various digital circuits using different VLSI simulation software tools like Modelsim,Xilinx and Questa. The outcome of this course to learn VHDL and Verilog language and also learn the usage of different tools.

X-WINDOWS PROGRAMING LAB

Course/Paper: 08BCS-203
BCS Semester VIII

Prerequisite : Knowledge of C Language

Lab Objective:
understanding fundamentals of Windows operating systems, • preparing students theoretically and practically to apply acquired knowledge and skills in producing programs for Windows operating systems.

To understand x-windows, x-lib, x-toolkit and x network protocol and learn it's commend line argument.

Programs in C/C++ language.
2. Write a program to establish connection with x server and get the sender and protocol information.
3. Using X library of the server, write a program to create a new window of a given size, title, border, foreground and background
colors.
4-5 To implement keyboard event handling/marking using x library.
6-7 To implement mouse event handling/marking using x library and interface with windows managers and drawing applications.
8. To implement a multiple windows application.
9-10 To implement various drag and drop based GUI components in Visual Basic.
11-12 To implement various drag and drop based GUI components in Motif and Lesstif.

PROJECT STAGE-II

Course/Paper: 08BCS-204
BCS Semester VIII

Here students have to submit detailed project report with there implementation in external viva as well as internal viva.

Objective

- Knowledge and application of research methods appropriate to the field of study
- Critical thinking and problem solving skills
- Application of knowledge and skills to plan and execute a substantial research project
- Communication skills to justify and interpret theoretical propositions, methodologies, conclusions and decisions to technical and non technical audiences
- Application of skills and knowledge with personal autonomy and accountability

This course is particularly important in the development of graduate attributes related to innovation, work readiness, global outlook, environmental awareness, social awareness and life long learning.

Learning Outcome:

AFTER COMPLETION OF THIS COURSE, student WILL BE ABLE TO:

1. Conduct an independent research project under supervision
2. Adhere to responsible laboratory or field practice regarding data collection and recording, and laboratory/field safety
3. Demonstrate time and project management in the successful identification of a research project, development of an experimental design, collection of accurate and precise data, critical analysis and interpretation of results, retrieval of information, and critical reading of scientific literature

5. Present a seminar on the results of a research project.

SEMINAR

Course/Paper: 08BCS-205
BCS Semester VIII

Here students have to submit a synopsis of recent topic in research and on approval they have to submit detailed report and give viva both in external and internal examination.

Objective:

To increase awareness of professional, societal and ethical issues associated with the practice of engineering.

Outcome :

1. Consider career options and set career goals
2. Identify areas of legal concern in engineering
3. Understand the need to be knowledgeable of contemporary issues