Course Title: Discrete Structures Full Marks: 60+15

Course No: CSc 511 Pass Marks: 30+7.5

Credit Hours = 3

Nature of the course: Theory+Tutorial

Course description: Mathematical foundation of computer science, logic, sets and relations. Graphs, algebraic structures, Boolean algebra and combinatorics.

Course Objectives:

- Introduce discrete mathematics concepts necessary to understand basic foundation of computer science.
- Introduce mathematical logic applied to circuits, automata and algorithm analysis.

Course Contents:

Unit1:

1.1 Set operations, Cartesian products, relations and functions. 5hrs

1.2 Lattice, posets. 5hrs

1.3 Data structures for representing trees and graphs. 3hrs

Unit2:

2.1 Network applications of trees and graphs. 4hrs

2.2 Propositional calculus, truth tables, mathematical induction. 5hrs

2.3 Elementary combinatorics. 4hrs

Unit3:

3.1 First order logic, Boolean algebra, lattices. 6hrs

3.2 Proof techniques: induction, contradiction, reverse induction, construction, etc. 4hrs

Unit4:

4.1 Application of Boolean algebra for digital circuit design 5hrs

4.2 Inclusion/exclusion principle, pigeon-hole principle and applications 4hrs

Recommended Books:

1. Keneth Rosen: Discrete mathematical structures with Application to computer Science, WCB/McGraw-Hill, 1999.

2. Joe L. Mott, Abrahan Kandel, and Theodore P. Baker: Discrete Mathematics for Computer Scientists and Mathematicians, Second Ed. Prentice-Hall of India.

Course Title: Advanced Data Structures

Course Code: CSc 512

Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Description: Review of Fundamental data structures: lists, arrays, tree, graphs and recursive structures. Develop tools for analysis of data structure and simple to complex algorithms, searching, sorting, hash and applications. Introduction of advanced data structures: union/ find, red/black tree, B-trees, tries, splay trees, binomial heaps, etc.

Course objectives:

1. Introduce basic and advanced data structures dealing with algorithm development and problem solving.
2. Introduce techniques for the design of efficient algorithms by using advanced data structures.

Course Contents:

Unit 1: Lists, Stacks and Queues:

1.1 Review of mathematical tools used for the development and analysis of data structures. Big-oh notation, space and time complexity analysis. **2hrs**

1.2 Linear data structures: arrays, lists, stacks and queues. Magic square filling problem, Infix to pre/postfix conversion and evaluation algorithms, CLL, DLL, & DCLL, Queue as CLL, DEqueue in DLL, Josephus problem, generalized list. Data structures for manipulating sparse matrices and polynomial arithmetic. **7hrs**

Unit 2: Trees and Disjoint Sets:

2.1 Review of recursive algorithms, TOH problem, Pre to Post translation algorithm. **2 hrs**

2.2 Review of tree structures,binary trees,traversals and search trees . Tree balancing techniques: AVL trees,Red/black trees. **4hrs**

2.3 B- trees, B+ trees, Interval trees, Splay trees. **4hrs**

2.4 Data structures for set manipulation, union/find algorithms, weighted union, path compression. **3 hrs**

Unit 3: Sorting

3.1 Application of data structures for developing efficient sorting algorithms- insertion sort ,selection sort, radix sort, quick-sort, heap sort and merge sort. Efficiency of sorting algorithms. **8 hrs**

Unit 4: Searching

4.1 Efficiency of searching algorithms, search structures: optimal Binary search tree, B- trees, tries. **3 hrs**

4.2 Hashing with application in systems programming, Extendible hashing. **3 hrs**

Unit 5: Heaps

5.1 Heaps and priority queues, min-max heaps, deaps, leftist trees, binomial heaps, Fibonacci heaps, Amortized analysis, spilt, melt, delete, lazy- delete operations. **5 hrs**

Unit 6: Graph

6.1 Data structures for representation and manipulation graphs and networks use of adjacency matrix and adjacency list. Topological sorting MST problem, Dijkstra's

algorithm. **4 hrs**

Lab work:

Extensive programming exercises in C/C ++ that demonstrates the use of alternative data structures for design, development and implementation of problems in computer applications.

References:

1. Weiss. M.a. Data structures and algorithms analysis , Adddtion Wesley 1999
2. Horowitz, E.et al Fundamental of Data Structures in C++, ISBN 0-7167- 8292-8
3. Cormen,T.H,et.al Introduction to Algorithms. MIT press/ McGraw – Hill 1990.
4. Kingston, J.H. Algorithms and Data structures, Addition Wesley.

Course Title: Operating System Full Marks: 60+15

Course No: CSc 513 Pass Marks: 30+7.5

Credit Hours = 3

Nature of the course: Theory +Lab

Course Description: Fundamental components of operating systems, process management, memory management, scheduling, I/O devices and drivers, critical section, deadlocks and overview of a few well known operating systems.

Course Objectives:

- Introduce the underlying principles of an operating system.
- Exposure of multi-programming, virtual memory and resource management concepts.
- Case study of public and commercially available operating systems.

Course Contents:

Unit1:

1.1 Overview of operating systems concepts: batch systems, multiprogramming, time-sharing, real-time. Storage hierarchy, system calls, virtual machines, hardware protection. 3hrs

1.2 Processors management: scheduling, co-operating processes, CPU as a critical resource and its sharing, multiple processors. 4hrs

1.3 Critical section problems, semaphores, deadlocks/detection, avoidance and prevention monitors. 3hrs

Unit2:

2.1 Memory management/logical and physical view of memory, contiguous allocation and swapping. 3hrs

2.2 Segmentation, Paging, segmentation with paging, virtual memory, demand Paging. 4hrs

2.3 Page replacement algorithms, allocation of frames, thrashing, demand segmentation. 3hrs

2.4 File systems interface, directory structures, access methods, protection. 3hrs

Unit3:

3.1 Input/Output system, I/O hardware, transforming I/O requests to hardware operations. 4hrs

3.2 Disk structure, disk scheduling, disk reliability, stable storage management, operating system jobs. 3hrs

Unit 4:

4.1 Programming assignment: each student is required to design and implement a program dealing with scheduling, memory management, and other resource management. 5hrs

4.2 Case study: overview of UNIX, Linux, and Windows NT. 3hrs

Recommended Books:

1. Silberschatz, A. and P.B. Galvin. Operating System Concepts, Fifth Edition. John-Wiley 1998.

Course Title: Algorithmic Mathematics

Course Code: CSc 514

Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives: To solve the technical and scientific problems by using the theory of numerical, statistical, and optimal computational procedures along with practical applications.

Course Contents:

Unit 1: Solution of Algebraic and Transcendental Equations (3 Hrs)

Newton Raphson Method, Secant Method, Solution of systems of Nonlinear Equations (Newton Raphson Method)

Unit 2: Interpolation (5 Hrs)

Errors in polynomial interpolation, Finite Differences, Differences of a polynomial, Newton's formulae for Interpolation, Bessel's Formula, Everett's Formula, Relation between Bessel's and Everett's Formulae, Lagrange's Interpolation Formula

Unit3: Curve Fitting, B-Splines and Approximation (4 Hrs)

Least- squares Curve Fitting Procedures (Linear, Quadratic and Exponential), B-splines, Approximation of Functions

Unit 4: Numerical Differentiation and integration (4 Hrs)

Numerical Differentiation, Trapezoidal Rule, Simpson's 1/3 –Rule, Simpson's 3/8 –Rule, Volume calculation, Newton-Cotes Integration Formulae, General Quadrature Formula, Gaussian Integration

Unit 5: Matrices and Linear Systems of Equations (3 Hrs)

Solution of Linear Systems- Direct Methods (Gauss Jordan), Solution of Linear Systems- Iterative Method (Gauss-Seidel), Eigen value Problem (Eigen Value, Eigen Vector)

Unit 6: Numerical Solution of Ordinary Differential Equations (4 Hrs)

Solution of Taylor's Series, Euler's Method (Modified Method), Predictor-Corrector Methods, Simultaneous and Higher Order Equations (4^{th} order Runge Kutta Method), Boundary Value Problems (Finite Difference Method)

Unit 7: Numerical Solution of the Partial Differential Equations (3 Hrs)

Finite- Difference Approximations to derivatives, Laplace's Equation, Parabolic Equations, Hyperbolic Equations, Iterative Methods for solution of Equations

Unit 8: Introduction and Descriptive Statistics (3 Hrs)

An overview of probability and statistics, Pictorial and tabular methods in descriptive statistics, Measures of central tendency, dispersion, and direction, Joint and conditional probabilities, Central limit theorem

Unit 9: Discrete Random Variables and Probability Distributions (3 Hrs)

Random variables, Probability distributions for random variables, Expected values of discrete random variables, The binomial probability distribution, Hypothesis testing using the binomial distribution, The Poisson probability distribution

Unit 10: Hypothesis Testing Procedures (4 Hrs)

Tests about the mean of a normal population, The t-test, Test procedures for a population variance, Z-tests for differences between two populations means, The two-sample t-test, A confidence interval for the mean of a normal population

Unit 11: Optimization Techniques (4 Hrs)

The simplex method, objective function and constraint conditions, changing inequalities to equalities, the conical form, of solution, optimal values of variables, Integer programming, Dynamic programming

Unit 12: Transformation (5 Hrs)

Laplace transform, Fourier transform, Discrete Fourier transform, Fast Fourier transform, Z transform and their inverse transform

References:

1 P. Thangaraj, "*Computer Oriented Numerical Methods*", Prentice Hall of India Private Limited, 2008
2. Jay L. Devore, "Probability and Statistics for Engineering and the Sciences", Brooks/Cole Publishing Company, Monterey, California, 1982.
3. Introductory methods of Numerical analysis, S.S. Sastry
4. An Introduction to numerical computations, S. Yakowitz and F. Szidarovszky
5. Numerical Methods, Dr. V.N. Vedamurthy, Dr. N. Ch. S.N . Iyengar
6. Numerical Methods , E. Balagurusamy

Course Title: Advanced Database Design
Course Code: CSc 515
Credit Hours = 3
Nature of the course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course objectives: To study the further advanced database techniques beyond the fundamental database techniques which were covered in the graduate level course (B.Sc.), and thus to acquaint the students with some relatively advanced issues. At the end of the course students should be able to: gain an awareness of the basic issues in objected oriented data models, learn about the Web-DBMS integration technology and XML for Internet database applications, familiarize with the data-warehousing and data-mining techniques and other advanced topics, and apply the knowledge acquired to solve simple problems.

Course Contents:

Unit 1: The Extended Entity Relationship Model and Object Model (5 hrs)
The ER model revisited, Motivation for complex data types, User defined abstract data types and structured types, Subclasses, Super classes, Inheritance, Specialization and Generalization, Constraints and characteristics of specialization and Generalization, Relationship types of degree higher than two.

Unit 2: Object-Oriented Databases (8 hrs)
Overview of Object-Oriented concepts, Object identity, Object structure, and type constructors, Encapsulation of operations, Methods, and Persistence, Type hierarchies and Inheritance, Type extents and queries, Complex objects; Database schema design for OODBMS; OQL, Persistent programming languages; OODBMS architecture and storage issues; Transactions and Concurrency control, Example of ODBMS

Unit 3: Object Relational and Extended Relational Databases (8 hrs)
Database design for an ORDBMS - Nested relations and collections; Storage and access methods, Query processing and Optimization; an overview of SQL3, Implementation issues for extended type; Systems comparison of RDBMS, OODBMS, ORDBMS

Unit 4: Parallel and Distributed Databases and Client-Server Architecture (8 hrs)
Architectures for parallel databases, Parallel query evaluation; Parallelizing individual operations, Sorting, Joins; Distributed database concepts, Data fragmentation, Replication, and allocation techniques for distributed database design; Query processing in distributed databases; Concurrency control and Recovery in distributed databases. An overview of Client-Server architecture

Unit 5: Databases on the Web and Semi Structured Data (8 hrs)

Web interfaces to the Web, Overview of XML; Structure of XML data, Document schema, Querying XML data; Storage of XML data, XML applications; the semi structured data model, Implementation issues, Indexes for text data

Unit 6: Enhanced Data Models for Advanced Applications (8 hrs)
Active database concepts; Temporal database concepts; Spatial databases Concepts and architecture; Deductive databases and Query processing; Mobile databases, Geographic information systems.

References:

1. Elmasri and Navathe, Fundamentals of Database Systems [4e], Pearson Education
2. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems [3e], McGraw-Hill
3. Korth, Silberchatz, Sudarshan , Database System Concepts, McGraw-Hill.
4. Peter Rob and Coronel, Database Systems, Design, Implementation and Management, Thomson Learning.
5. C.J.Date, Longman, Introduction to Database Systems, Pearson Education

Course Title: Advanced Computer Architecture

Course Code: CSC 521

Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course objectives: This course is designed to provide information of state-of-the-art high performance computer architectures. Topics include performance, ISA, instruction-level parallelism (ILP), thread-level parallelism (TLP), dynamic scheduling, out-of-order execution, register renaming, exception handling, static scheduling (VLIW/EPIC), cache/memory/DRAM/storage hierarchy design, speculation techniques, advanced branch predictor design, multiprocessor, coherency issues, storage systems, multicore processors, case studies including P6, Netburst, Core, Itanium, UltraSparc T2, New trends in architecture/microarchitecture development in face of physical design limit.

Course Prerequisite: Computer Organization and Operating System or the equivalent.

Course Contents:

Unit1: Fundamentals of Computer Design (10 Hrs)

Trends in Technology, Cost/Price Measuring and Reporting Performance, Benchmark suits, Principles of Quantitative Computer Design: Amdals Law, CPU Performance Equation, Instruction Set Architecture Design, Pipelining and its hazards, Case Study: Desktop-Server-Embedded Systems – Price-Performance

Unit 2: Instruction Level Parallism (12 Hrs)

Classification of types of hazard, Data hazards, Control hazards, Forwarding, interlocks, instruction scheduling, Branch instruction costs, Dynamic hardware prediction, Hardware Speculation, Limitations of Instruction Level Parallelism, Multithreading, Superscalar and VLIW architectures, Case study: Pentium

Unit 3: Multiprocessor and Thread Level Parallism (8 Hrs)

Multicore processors or Architectures: SIMD, MIMD, Shared Memory, Cache Coherence, Synchronization, Memory Consistency models, Case Study: Recent Research on Multiprocessors

Unit 4: Memory Hierarchy Design (10 Hrs)

Classification of cache organization, Quantifying Cache performance, Average Memory access time – processor performance, Miss penalty, Reducing miss penalty: Multi-level cache, Miss verses Write priority, Write buffer etc., Reducing Miss Rate: Block size, Cache size, Associativity, Compiler optimization, Reducing Hit Time: Cache simplicity, Address translation, Trace Caches, Main Memory Organization, Bandwidth, Memory interleaving, Memory Banks, Case Study: 21264 and/or Sony Playstation 2

Unit 5: Storage System (5 Hrs)

Disk Storages, Faults, Failures, I/O Performance, Reliability Measures and Benchmarks

References:

1. Hennessy J.L., Patterson D.A., Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
2. David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware / Software Interface.

Course Title: Automata Theory Full Marks: 60+15

Course No: CSc 522 Pass Marks: 30+7.5

Credit Hours = 3

Nature of the course: Theory +Tutorial

Course Description: Deterministic and non-deterministic finite state machine, regular expressions and their properties. Context free grammars, push-down automata and applications , undecidable and intractable problems, Gentle introduction to classes P and NP.

Course Objectives:

- Mathematical models of computers: DFA, NFA, CFG, pushdown automata, Turing machine.
- Notions of computability, and undecidability.

Course Contents:

Unit1:

1.1 Deterministic and non-deterministic finite states machines. Equivalence between NFA and DFA. 6hrs

1.2 Regular expressions and their properties, relationship between DFA and regular expressions, pumping lemma, algorithms for minimizing finite state machines.5hrs

Unit2:

2.1 Context free grammar(CFG), properties, closure properties, languages and automata, languages that are not context free, ambiguous languages. 6hrs

2.2 Push down automata, deterministic and non-deterministic PDA, relationship between CFG and PDA. 5hrs

2.3 Undecidability, post-correspondence problem, several examples undecidable

problems. 7hrs

Unit3:

3.1 Turing machines, computation by Turing machines, extensions of Turing machines,

Turing enumerable languages. 5hrs

3.2 The Church-Turing thesis, universal Turing machine, halting problems . 6hrs

3.3 Computational complexity , class P, class NP, interactable problems, NP complete

problems. 5hrs

Recommended Book:

1. Efim Kinber and Carl Smith , Theory of Computing: A Gentle Introduction

   Prentice Hall, ISBN 0-13-027961-7.

2. Hopcroft and Ullman , Automata Theory, Addison-Wesley, 1979.

Course Title: Design and Analysis of Algorithm

Course Code: CSc 523

Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives: The main aim of this course is to introduce the concept of algorithmic approach for solving real-life problems that arise frequently in computer applications and to teach basic principles and techniques of computational complexity. This course helps the student to develop the habit of always responding to a new algorithm: with a problem at hand, helps exploring one or more approaches for solving it, then develop a new algorithm, analyze it, and modifies or rejects it until a satisfactory result is achieved.

Course Contents:

Unit 1: Principles of Analyzing Algorithms and Problems (4 Hrs)

Introduction, Algorithm Specification, Performance Analysis, Randomized Algorithms

Unit 2: Review of Abstract Data Types (5 Hrs)

Stacks, Queues, Binary Trees, Dictionaries, Priority Queues, Sets and Disjoint Set Union, Graphs

Unit 3: Sorting, Selection and Sequencing (9 Hrs)

Divide-and Conquer Method, Binary Search Technique, Finding the Minimum and Maximum, Merge Sort, Quick Sort, Selection, Strassen's Matrix Multiplication, General Method of Problem Solving – The Greedy Method, Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum Cost Spanning Tree, Optimal Storage Allocation on Magnetic Tapes, Optimal Merge Patterns.

Unit 4: Dynamic Programming (9 hrs)

Dynamic Programming as a General Method, Multistage Graphs, All Pair Shortest Paths, Single Source Shortest Path: General Weights, Optimal Binary Search Trees, String Editing, 0/1-Knapsack, Reliability Design, TSP

Unit 5: Graph Traversal and Search Techniques (9 hrs)

Binary Tree Traversal Techniques and Search, Graph Traversal Techniques and Search: BFS and DFS, Connected Components and Spanning Trees, Bi-connected Components and DFS.

Unit 6: Backtracking (9 hrs)

Introduction to Backtracking, The 8-Queens Problem, Sum of Sub-sets, Graph Coloring, Hamiltonian Cycle, Knapsack Problem

References:

1 Horowitz, E., Sahni, S., and Rajasekaran, S., *Computer Algorithms C++*, Galgotia Publications, 1999.
2 Horowitz, E., Sahni, S., and Rajasekaran, S., *Computer Algorithms*, Galgotia Publications, 1999.
3 Baase, Sara., and Gelder, Allen Van, *Computer Algorithms – Introduction to Design and Analysis*, Third Edition, Addison-Wesley, Pearson Education Asia, 2000, ISBN: 81-7808-171-7

Course Title: Optimization Techniques

Course Code: CSc 524

Credit Hours = 3

Nature of the course: Theory + Tutorial

Full Marks: 60 + 15

Pass Marks: 30 + 7.5

Course description: This course is about optimization techniques and complexity analysis in linear, integer and nonlinear programming and network flows. It is designed to provide the students with in-depth knowledge of theoretical as well as practical background and their implementations. The approach of the course will be mathematical modeling, numerical methods, algorithmic analysis and implementations.

This course in optimization techniques mainly focuses in theoretical computer science and scientific computing. It relates with operations research methods and numerical mathematics.

Course objectives: The main objectives of this course are enabling the students:

1. To broaden their knowledge towards modeling techniques, numerical methods and algorithms.
2. To realize the importance of various aspects of optimization techniques in every day experiences in the industry like computer, manufacturing and IT.
3. To implement the knowledge of optimization techniques in real life problems.
4. To be motivated and prepared for the study of more advanced field of optimization methods and complexity analysis.

Course Contents:

Unit 1: Linear Programming and Sensitivity Analysis (7 Hrs)

Two-variable LP-model, graphical and algebraic LP solutions, some LP applications, the Simplex Method and sensitivity analysis, primal-dual relationships and economic interpretation, dual simplex and generalized simplex algorithms and post-optimal analysis.

Unit 2: Transportation and Network Models (8 Hrs)

The transportation models and algorithm, the assignment and transshipment models, minimum spanning tree algorithm, shortest-route problem, maximum flow and min-cost models, critical path method and algorithms for matching.

Unit 3: Advanced Linear Programming and Applications (7 Hrs)

Simplex method fundamentals, revised simplex method and computational considerations, bounded variables algorithm, duality, parametric linear programming, goal programming formulations and algorithms.

Unit 4: Integer Linear Programming (8 Hrs)

Illustrative applications, integer programming algorithms, unimodularity and cutting-plane methods, traveling salesperson problem.

Unit 5: Dynamic Programming and its Application (7 Hrs)

Recursive nature of computations in DP, forward and backward recursion, selected DP applications, problem of dimensionality, branch and bound method and dynamic programming, some deterministic inventory models.

Unit 6: Nonlinear Programming (8 Hrs)

Convex programming problems, unconstrained problems and algorithms, constrained problems and algorithms.

References:

1. H. A. Taha, Operations Research: An Introduction, Pearson Prentice Hall.

2. C. H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice Hall India.

Course Title: Internet Programming
Course Code: CSc 525
Credit Hours = 3

Nature of the course: Theory + Lab
Full Marks: 60+15
Pass Mark: 30+7.5

Course objectives:

1. Introduction tools for creating and maintaining websites
2. To impart basic understanding of Internet.
3. The course also aims to build foundation on E-mail and E-Commerce.

Course Contents:

Unit 1: (3 Hrs)

Introduction to Networking and its advantages, Centralized Vs. Distributed System, Introduction to File Server Systems and Client/Servers Systems, Introduction to Internet and its Evolution, Internet as a Client/Server Technology

Unit 2: (10Hrs)

Markup Languages, Editing HTML, Different sections of an HTML document, Common Tags and Headers, Text Styles, Linking and Images, Formatting text with <FONT>, Special characters, Horizontal rulers and Line breaks, Unordered List, Nested and ordered lists in HTML, Different types of ordered lists, HTML tables, Complex tables with formatting and color, Simple Forms, Forms including text areas, password boxes and check boxes, HTML forms including radio buttons and pull down boxes, Frames

Unit 3: (4 Hrs)

DHTML Vs. HTML, Introduction to Cascading Style Sheets, Declaring styles in the header section of a document, Linking external style sheets

Unit 4: (3 Hrs)

Introduction to different image formats (GIF, JPEG, BMP etc.), Working with colors, Adding text to images, Converting images from one format to another, Creating GIFs

Unit 5: (5 Hrs)

Introduction to PHP, Server-side processing, Interleaving HTML and PHP, Loops and arrays, PHP Database functions

Unit 6: (8 Hrs)

Introduction, Elements, Attributes, Values, Rules for writing XML, Creating Root Elements, Writing non-empty elements, Nesting Elements, Adding attributes, XML Schema: Simple and Complex types, Using Namespace in XML:XSLT and XPath, Transforming XML with XSLT, Beginning an XSLT style sheet, Test expressions and functions

Unit 7: (8Hrs)

Creating a DTD, Declaring internal DTD, Writing an External DTD, Naming an external DTD, Declaring a personal External DTD, Declaring a public external DTD, Defining elements and attributes in a DTD

Unit 8: (4Hrs)

Introduction to E-mail, Privacy issues of E-mail, Types of E-mail (Web Vs. Client), Introduction to an E-mail client, Configuring the E-mail client to connect to the service provider, Email – Names and addresses, Introduction to E-Commerce, Types of E-Commerce

References:

1. Internet and World Wide Web – How to Program by Deitel, Deitel and Nieto
2. HTML & XHTML: The definitive Guide , fifth edition by Chuck Musciano, Bill Kennedy
3. Web Database applications with PHP & MySQL, by Huge E. Williams, David Lane
4. XML for the world wide web , Elizabeth Castro

Course Title: Embedded System

Course Code: CSc 526

Credit Hours = 3

Nature of the Course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course objective: To allow the student to study the design and development process for dedicated computer systems in relation to the environment in which they operate.

Unit 1: Introduction (7 Hrs)

Overview of dedicated and automated systems and their specific requirements (robust design, environmental issues, temporal constraints, technological constraints, software systems); the product design cycle.

Unit 2: System Specification and Integration (10 Hrs)

Development of a system specification, including case studies, Evaluation and justification of the available levels of system integration (custom chipdesign through to turnkey-systems) and technological choice.

Unit 3: Software Issues (14 Hrs)

Development environment: compilers, linkers, debuggers, emulators, real time operating systems and kernels, Designing and implementing code for dedicated systems

Unit 4: Hardware Issues (14 hrs)

Choice of processor: I/O, memory, speed, integration, development facilities, economics; DSP devices, Interfacing to commonly used peripheral devices, Backplane Bus standards, Transducers: sensors for measuring physical phenomena, output devices such as power actuators and motors, Data transformation, signal conditioning and data conversion. The impact of EMC regulations on design practice.

Reference:

1 S Heath, "Embedded System Design", Butterworth-Heinemann 1997, ISBN0-75063-237-2

Course Title: Multimedia Technology
Course Code: CSc 527
Credit Hours = 3

Nature of the course: Theory +Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives: The main objective of this course cover three main objectives on multimedia technology these are devices, systems and applications

Unit 1: Introduction (4 Hrs)

Global structure of Multimedia, Medium, Multimedia system and properties

Unit 2: Sound / Audio System (5 Hrs)

Concepts of sound system, Music and speech, Speech Generation, Speech Analysis, Speech Transmission

Unit 3: Images and Graphics (5 Hrs)

Digital Image Representation, Image and graphics Format, Image Synthesis , analysis and Transmission

Unit 4: Video and Animation (5 Hrs)

Video signal representation, Computer Video Format, Computer- Based animation, Animation Language, Methods of controlling Animation, Display of Animation, Transmission of Animation

Unit 5: Data Compression (6 Hrs)

Storage Space, Coding Requirements, Source, Entropy and Hybrid Coding, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-based Mode, Hierarchical mode, Video and Audio Encoding, Audio and still Image Encoding

Unit 6: Communication Systems in Multimedia (6 Hrs)

Application Subsystem, Transport subsystem, Quality of service and resource, management, Trends in collaborative Computing, Trends in Transport Systems

Unit 7: User Interfaces (4 Hrs)

Basic Design Issues, Video and Audio at the User Interface, User- friendliness as the Primary Goal

Unit 8: Abstractions for programming (5 Hrs)

Abstractions Levels, Libraries, System Software, Toolkits, Higher Programming Languages, Object –oriented approaches

Unit 9: Multimedia Application (5 Hrs)

References:

1. Multimedia: Computing, Communications and Applications, Ralf Steinmetz and Klara Nahrstedt, Pearson Education Asia
2. Multimedia Communications, Applications, Networks, Protocols and Standards, Fred Halsall, Pearson Education Asia
3. Multimedia Systems, John F. Koegel Buford, Pearson Education Asia

Course Title: Distributed Computing

Course Code: CSc 528

Credit Hours = 3

Nature of the Course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objective: To allow the student to study the design and development process for distributed system.

Unit1. Introduction (5 hrs)

Concept Distributed Computing-a more practical approach, comparison with parallel computing , Grid computing, Peer to peer/Workgroup computing, Cluster computing; Basic layers of software concurrency-Distributed computing and real time systems, Compute-to-Data and Compute-to- Communication ratios; Distributed Computing Applications, Query search, Exhaustive search, Complex modeling and simulation etc., Tightly coupled, loosely coupled and shared systems, Atomicity- fine grained, medium grained and large grained paradigms, Shift of paradigm-Service Oriented Architecture (SOA)

Unit 2: Performance issues (3 hrs)

Single Vs. Multiple program performance, Scalable Vs. Parallel program performance, Benchmarks (serial, parallel, PERFECT)

Unit 3: Interconnection networks (5 hrs)

Cache hit-rate model, Cache coherency, Static and Dynamic networks, Internet mega computer-Network resources and their identifications, OLE/ COM, Distributed objects and CORBA, RPC and Rendezvous, Internet agents, Porting of applications- accessibility, scalability, security, fault tolerance

Unit 4: The PRAM model and algorithms (6 hrs)

CRCW, CREW, ERCW, EREW models, Design and analysis of Parallel algorithms – All sum, Matrix multiplication, Sorting, Searching, Merging, Minimum spanning tree, Prime numbers, Image filtering

Unit 5: Message-passing models and algorithms (5 hrs)

Synchronous and Asynchronous message passing models, Leader-Election algorithm, Breadth-First Search. Shortest Paths, Broadcast and Converge cast.

Unit 6: Scheduling (8 hrs)

Scheduling problem, scheduling model, Two processor scheduling, Scheduling a Directed Acyclic Graph (DAG), Heuristic algorithms, priority based scheduling, clustering, task duplication, Scheduling in heterogeneous environments, Computing topology of networks, Broadcast, Logical clocks and event ordering, Token passing, Distributed Mutual Exclusion, Distributed Dining Philosophers and Decentralized Dining Philosophers

Unit 7: Programming issues (9 hrs)

Reverse engineering of sequential programs, Semantics of concurrent execution, loop conversions, techniques for avoiding interference, fundamental building blocks-Task, Fan, Tree, Par, Pipe, Message buffer; Defining Virtual Machine, configuration, connections, vector variables, parallel and reduction operations, , data distribution, work assignment- Connect, Send, Receive, Disconnect, Program structure, Synchronization, Locks and Barriers, Semaphores, Message passing Vs. Distributed objects, Supervision, Message Passing Interface (MPI), Standard Template Adaptive Parallel Library (STAPL), Multithreading, Client–server/ Internet applications - Java packages, Socket API and Class Libraries, Java Applets, Java Scripts, CGI and Servlets, Web services and Simple Object Access Protocol (SOAP)

Unit 8: Advanced Distributed Computing Paradigms (4hrs)

Message Queuing, Mobile agents, Network services, Object spaces, Recent trends

References:

1. Distributed and Parallel Computing By Hesham El-Rewini and Ted G. Lewis (Manning Publications)
2. Distributed Computing: Principles and Applications By M.L. Liu (Addison-Wesley)
3. Parallel and Distributed Programming using C++ By Cameron Hughes (Addison Wesley Professional)
4. Foundations of Multithreaded, Parallel and Distributed Programming By Gregory R. Andrews (Addison-Wesley)

Course Title: Artificial Intelligence Full Marks: 60+15

Course No: CSc 529 Pass Marks: 30+7.5

Credit Hours = 3

Nature of the course: Theory +Lab

Course Description: Knowledge Representation, inference rules and predicate calculus. Search methods, state space search, pattern directed search, predicate calculus, blackboard architecture. Rule based expert system, model based reasoning, network representation.

Course Objectives:

- Introduce tools representation and search such as predicate calculus, state space search, and/or graph, heuristic search, monotonocity, Alpha-Beta procedures, and pattern directed search.
- Introduce expert system technology; rule based reasoning methods, issues of incomplete information, and network representation.

Course Contents:

Unit1:

1.1 Historical foundation, development of logic, biological and social models of intelligence, the Turing test, modeling human performance. 3hrs

1.2 The predicate calculus: semantics of propositional calculus, syntax of predicate calculus, inference rules. 4hrs

1.3 Search Strategies: state space representation and state space search, data driven and goal driven search, depth first search and breadth first search, application of And/or graphs 6hrs

Unit2:

2.1 Heuristic search: "Best –Fit" search, admissibility and monotonocity, heuristic in games, exhaustive search, minimax procedure, alpha-beta procedures. 5hrs

2.2 Recursion -based search, pattern directed search, control of search in production, production system in AI, predicate calculus and planning, use of blackboard architecture. 6hrs

Unit3:

3.1 Expert system technology, design of rule based expert systems, knowledge engineering process. Goal driven reasoning, data driven reasoning. 5hrs

3.2 Model based reasoning, case based reasoning, the problem of knowledge representation. 4hrs

3.3 Reasoning with incomplete information: Statistical approach to uncertainty, Bayesian Belief Network, theory of evidence, causal networks. 5hrs

3.4 Nonmonotonic systems, reasoning with Fuzzy sets, issues in knowledge representation, semantic sets, conceptual graphs, structured representation. 7hrs

Recommended Books:

1. George F. Luger and William A. strublefied, : "Artificial Intelligence: Structure and Strategies for Complex Problem Solving" Third Edition , Addospm Wes:pmg,am.Ome., 1998.

2. S. J Russel and P. Norvig "Artificial Intelligence:A Modern Approach " Prentice Hall, 1995.

Course Title: Compiler Construction Full Marks: 60+15

Course No: CSc 631 PassMarks:30+7.5

Credit Hours = 3

Nature of the course: Theory +Lab

Course Description: Basic tools of compiler construction, lexical analysis, parsing, LALR, LL, and recursive descent methods. Quadruples generation and machine code generation. Implementation of a compiler for a simple high-level language.

Course Objectives:

- Introduce the concept of lexical analysis, parsing, code generation and programming language translation.
- Implement a compiler in C/C++ to compile a simple programming language.

Course Contents:

Unit1:

1.1 Analysis of source program, grouping of phases, compiler construction tools. 4hrs

1.2 Syntax directed translation, parsing, simple expression, lexical analysis, stack machines. 4hrs

1.3 Lexical analysis, input buffering, token recognition, finite automata, regular expression, lexical analyzer generator, DFA based pattern matching. 5hrs

Unit2:

2.1 Syntax analysis, parser, context free grammars, top-down parsing, bottom-up parsing, operator precedence parsing. 4hrs

2.2 LR parser, ambiguous grammar, parser generators. 3hrs

2.3 Syntax directed definitions, construction of syntax trees, recursive evaluators. 3hrs

2.4 Type checking, type conversion, operator and function overloading, polymorphic function, unification. 5hrs

Unit3:

3.1 Source language issues, storage allocation, parameter passing, symbol table. 5hrs

3.2 Intermediate languages, declarations and statements, back patching,

procedure calls. 3hrs

3.3 Code generation, target machine, runtime storage management, flow-graphs, simple

code generator, register allocation 4hrs

3.4 Each student implements a compiler in C/C++ for a simple programming

language. 5hrs

Recommended Books:

1. A.V. Aho, R. Sethi and J.D. Ullman, Compilers: Principle techniques and tools, Addison Wesley, ISBN:0201100886

Course Title: Software Engineering

Course Code: CSc 632

Credit Hours = 3

Nature of the course: Theory +Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives:

1. This course aims to give students both a theoretical and a practical foundation in software engineering.
2. In the theoretical part, students will learn about the principles and methods of software engineering, including current and emerging software engineering practices and support tools.
3. In the practical part, students will become familiar with the development of software products from an industry perspective.

Course Contents:

Unit 1: Introduction (6 Hrs)

Introduction to software engineering, Computer Based system engineering, Software processes, Project management

Unit 2: Software requirements (8 Hrs)

Software requirements, Requirements engineering processes, System models, Software prototyping, Formal specification

Unit 4: Software Design (8 Hrs)

Architectural design, Distributed systems architectures, Object-oriented design, Real time software design, Design with reuse, User interface design

Unit 6: Verification and Validation (10 Hrs)

Verification and validation planning, Software inspections, Clean room software development, Defect Testing, Integrating Testing, Object –Oriented testing, Testing workbenches, Critical system validation

Unit 7: Software Quality and Quality Assurance (8 Hrs)

Software cost estimation, Software Quality assurance planning, Software quality assurance process, Software quality attributes, Guidelines and checklists, Software safety

Unit 8: Evolution (5 Hrs)

Software Change, Software re-engineering, Configuration management

References:

1. Software Engineering, " Ian Sommerville"
2. Software Engineering Fundamentals, " Ali Behforooz and Frederick J. Hudson"
3. Software Engineering: A Practitioner's Approach, "Roger S. Pressman"

Course Title: Computational Geometry Full Marks: 60+15

Course No: CSc 633 Pass Marks: 30+7.5

Credit Hours = 3

Nature of the course: Theory +Lab

Course Description: Elementary geometric objects, line segment interaction, convex hull computation in 2D and 3D, Proximity, Voronoi diagram , plane sweep, triangulation of polygons and point sets, path planning.

Course Objectives: Introduce the development of algorithmic tools for solving problems having geometric flavor. Apply such tools for solving problems related to robotics, graphics, mesh generation, and other areas.

Course Contents:

Unit1:

1.1 Representation of points , lines, line segments. Computation and detection of segment intersection. Point inclusion in polygon. 3hrs

1.2 Polygon triangulation , polygon illumination, intersection of polygons. 5hrs

1.3 Plane –sweep paradigm, efficient algorithms for detecting intersection of line segments in the plane. 6hrs

Unit2:

2.1 Convex hull computation in 2D: gift-wrap method, Graham scan, divide and conquer, quick hull, approximation algorithms. 8hrs

2.2 Convex hull computation in 3D: box enclosing method, Chand-Kapoor method, gift-wrap. 4hrs

2.3 Computing Voronoi diagrams, plane-sweep method, Delaunay Triangulation, relationship between Delaunay triangulation and Voronoi diagrams. 4hrs

2.4 Computing closest pair of points, using Voronoi diagram to compute closest pairs. Other applications of Voronoi diagram. 4hrs

Unit3:

3.1 Path planning and motion planning algorithms in 2D, visibility graphs, computing shortest paths by using visibility graphs, object growing. 4hrs

3.2 Computing shortest paths in three dimensions, approximation algorithms, interactability of shortest path computation in 3D. 3hrs

3.3 Algorithms for generation two dimensional mesh, application of Delaunay triangulation quad-tree method, mesh refinements, adaptive mesh refinements. 4hrs

Recommended Books:

1. J.O' Rourke, Computational Geometry in C. Cambridge University Press.

2. Edward Angel, Interactive Computer Graphics Addison Wesley,ISBN: 0-201-38596-X

Course Title: Object Oriented Analysis and Design

Course Code: 634

Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives:

- To develop a strong foundation in object-oriented analysis, design, and implementation of software systems.
- To choose appropriate design patterns for common design problems
- To use UML notation in the design of systems.

Unit 1: Object Oriented Methodology and UML

1.1: Introduction to Object Oriented Methodology **5 hrs**

Object Oriented Methodology, Object Oriented Methodologies - Booch Methodology, Jacobson et. al. methodology, Rambaugh et. al. object modeling technique. Object Oriented Modeling Techniques and OOM characteristics: class & objects, links & association, generalization & inheritance, An Object Model & Benefits of Object Oriented Modeling

1.2: Using UML **5 hrs**

Basic Concepts: Structural Diagrams - Class diagram, Object diagram, Composite diagram, Package diagram; Component diagram; Behavioral Diagrams - Use Case diagram, Communication diagram, Sequence diagram, Interaction diagram, Activity diagram, State diagram, Using CASE Tools

Unit 2 : Modeling

2.1: Object Modeling **4 hrs**

Modeling of Objects, Classes, Links, Associations; Advanced Modeling Concepts:- Aggregation, Abstract Class, Multiple Inheritance, Generalization

2.2: Dynamic Modeling **4 hrs**

Events and states, Elements of State Diagrams, Advanced Dynamic Modeling Concepts

Relation of Object and Dynamic Models

2.3: Functional Modeling **4 hrs**

Functional Models, Data Flow Diagrams- Features of a DFD, Design flaws in DFD, Relationship between Object, Dynamic, and Functional Models

Unit 3: Analysis, Design and Implementation

3.1: Object Oriented Analysis **6 hrs**

Object Oriented Analysis, Structured Analysis Vs Object Oriented Analysis, Analysis Techniques - Object Modeling, Dynamic Modeling, Functional Modeling, Adding operations and iterating analysis

3.2: Object Oriented Design **12 hrs**

System Design: An Object Oriented Approach - Breaking into Subsystems, Concurrency Identification and Management of data Store, controlling events between Objects

Object Design - Object Design for Processing, Object Design Steps, Advance Object Design

- Control and its Implementation, Inheritance Adjustment, Association Design, Object Representation, Design Optimization

3.3: Implementation **5 hrs**

Implementation Strategies - Unidirectional Implementation, - Bi-directional Implementation

Implementing associations, Constraints, State charts, Mapping design to code, Optimization, Components, Deployment

Recommended Books:

1.) **Grady Booch ,** *Object Oriented Analysis and Design*

2.) **G. Booch, J. Rambaugh and I. Jakobson,** *The Unified Modeling Language User Guide*, Addison Wesley

3.) **Ali Bahrami,** *Object Oriented System Development***,** McGraw Hill International Edition

4.) **Blaha and Rambaugh,** *Object Oriented Modeling and Design with UML*

Course Title: Systems Programming Full Marks: 60+15

Course No: CSc 635 Pass Marks: 30+7.5

Credit Hours = 3

Nature of the course: Theory +Lab

Course Description: This course will introduce the design and implementation of machine dependent, as well as machine independent aspects of assembler, loader, linker, and microprocessor. An overview of compilers will be provided. A project involving implementation of an assembler, a linker, and a loader will form an integral part of the course.

Course Objectives: Most Computer Science curriculum requires, or at least includes, stand-alone courses in such important software topics as Operating Systems and Compilers. No such course usually exists on some other important software topics such as assemblers, linkers, and loaders. The purpose of this course is to present the basic structure and design of a micro-assembler, a linker, and a loader. Since software components are best learned by implementation, each student will complete a project independently which will involve the design and implementation of these three software components.

Prerequisites: Logic Design, Advanced Data Structure,(familiarity with assembly language, and representation if instructions and data including complements within a computer, proficiency in a modern programming language, and various types of data structures such as linked list, trees, and hash tables).

Course Contents:

Unit1: SIC and SIC/XE machine structure 6hrs

1.1 Introduction, system software and machine architecture

1.2 Simplified Instructional computers SIC, SIC/XE architecture.

1.3 RISC and CISC machine architecture.

Unit 2: Assembler Features 11hrs

2.1 Basic assembles functions, machine dependent and Independent assemblers features.

2.2 One and two pass Assemblers Design Options.

Unit3: Loader functions: 12hrs

3.1 Basic loader functions, machine dependent and Independent features.

3.2 Machine dependent and independent Design Options

Unit4: Macro processor functions: 16hrs

4.1 Basic Macro processor functions, machine Independent features.

4.2 Macro processor Design Options

4.3 Overview of compiles.

Textbook: System Software: An Introduction to Systems Programming by Leyland L.Beck: Addision Wesley Publishing Company,1985.

Reference Books:

1. Daniel H. Marcellus, Systems programming for Small Computers, prentice-Hall, New

Jersey,1984.

2. Roy S. Ellzey, Computer System Software, The Programmer/Machine Interface,

Inc.,1987.

3. Robert M. Grahm , Principles of Systems Programming, John Wiley & Sons, 1975.

4. Thomas G. Windeknecht, 6502Systems programming, Little Brown & Company,1983.

Outcomes and Assessment:

At the completion of this course, students will be able to

i) Understand what is meant by machine architecture , and will; be able to answer questions related to the architecture(such as memory, register, instruction set, addressing modes, instruction formats) of the Simplified Instructional Computer(SIC) as well as its extension (SIC/XE) as described in the textbook.

ii) Understand and answer questions related to machine independent assembler features (literals, expressions, symbol definition statements, control sections and program blocks).

iii) Understand the difference between instruction and directive.

iv) Understand the differences in addressing modes and instruction formats.

v) Understand the concepts and implementation of the program relocation.

vi) Understand the difference between the design option of one-pass versus two-pass assembly.

vii) Understand the difference between absolute and relocatable loader.

viii) Understand the concepts and implementation of program linking.

ix) Understand the concept and implementation differences between static and dynamic linking.

x) Understand the concept and implementation of bootstrap loader.

xi) Understand the concept of overlays.

xii) Understand the role of macros.

xiii) Be able to complete macro expansion from macro definition.

xiv) Understand the logic and implementation of macro processors including conditional assembly.

xv) Understand difference options of using macro parameters.

xvi) Understand the basic working of a compiler.

Evaluation scheme:

a) There will be two internal exams which will account for 10% of the grade.

b) Five programming assignments which will contribute to the other 10% of the grade.

c) Final examination will contribute to 80% of the grade.

Course Title: Cryptology and Internet Security

Course Code: CSc 636

Credit Hours = 3

Nature of the Course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives: To be familiar with classical and modern encryption and decryption techniques and apply in the security system

Unit 1: Introduction (2 Hrs)

Introduction and Terminology, Shannon's Description of a Conventional, Cryptosystem, Statistical Description of a Plaintext Source,

Unit 2: Traditional Cryptosystems (5 Hrs)

Caesar, Simple Substitution, Vigenère, Caesar Cipher, Simple Substitution, The System and its Main Weakness, Cryptanalysis by The Method of a Probable Word, Vigenère Cryptosystem, The Incidence of Coincidences, Kasiski's Method, The Incidence of Coincidences, Kasiski's Method, Vernam, Playfair, Transpositions, Hagelin, Enigma, The One-Time Pad, The Playfair Cipher, Transposition Ciphers, Hagelin, Enigma,

Unit 3: Shift Register Sequences (5 Hrs)

Pseudo-Random Sequences, Linear Feedback Shift Registers, (Linear) Feedback Shift Registers, PN-Sequences, Which Characteristic Polynomials give PN-Sequences, An Alternative Description for Irreducible *f,* Cryptographic Properties of PN Sequences, Non-Linear Algorithms, Minimal Characteristic Polynomial, The Berlekamp-Massey Algorithm, A Few Observations about Non-Linear Algorithms,

Unit 4: Block Ciphers (4 Hrs)

Some General Principles, Some Block Cipher Modes, Codebook Mode, Cipher Block Chaining, Cipher Feedback Mode, An Identity Verification Protocol, DES, Triple DES, IDEA

Unit 5: Shannon Theory and Data Compression Techniques (4 Hrs)

Entropy, Redundancy, and Unicity Distance, Mutual Information and Unconditionally Secure Systems, Basic Concepts of Source Coding for Stationary Sources, Huffman Codes, Universal Data Compression - The Lempel-Ziv Algorithms, Initialization, Encoding, Decoding

Unit 6: Public-Key Cryptography (3 Hrs)

The Theoretical Model, Motivation and Set-up, Confidentiality, Digital Signature, Confidentiality and Digital Signature

Unit 7: Discrete Logarithm Based Systems (4 Hrs)

The Discrete Logarithm System, The Discrete Logarithm Problem, The Diffie-Hellman Key Exchange System, Other Discrete Logarithm Based Systems, ElGamal's Public-Key Cryptosystems, ElGamal's Secrecy System, ElGamal's Signature Scheme, Further Variations, Digital Signature Standard, Schnorr's Signature Scheme, The Nyberg-Rueppel Signature Scheme, How to Take Discrete Logarithms, The Pohlig-Hellman Algorithm, Special Case, General Case and example, The Baby-Step Giant-Step Method, The Pollard Method, The Index-Calculus Method

Unit 8: RSA Based Systems (5 hrs)

The RSA System, Some Mathematics, Setting Up the System, RSA for Privacy, RSA for Signatures, RSA for Privacy and Signing, The Security of RSA: Some Factorization Algorithms, What the Cryptanalist Can Do?, A Factorization Algorithm for a Special Class of Integers, Pollard's Method, General Factorization Algorithms, Random Square Factoring Methods, Quadratic Sieve, Some Unsafe Modes for RSA, A Small Public Exponent, Sending the Same Message to More Receivers, Sending Related Messages to a Receiver with Small Public Exponent, A Small Secret Exponent; Wiener's Attack, Some Physical Attacks, Timing Attack, The "Microwave" Attack, How to Generate Large Prime Numbers; Some Primality Tests, Trying Random Numbers, Probabilistic Primality Tests, The Solovay and Strassen Primality Test, Miller-Rabin Test, A Deterministic Primality Test, The Rabin Variant, The Encryption Function, Decryption, Precomputation, Finding a Square Root Modulo a Prime Number, The Four Solutions, How to Distinguish Between the Solutions, The Equivalence of Breaking Rabin's Scheme and Factoring n

Unit 9: Elliptic Curves and Coding Theory Based Systems (5 Hrs)

Some Basic Facts of Elliptic Curves, The Geometry of Elliptic Curves, Line Through Two Distinct Points, A Tangent Line, Addition of Points on Elliptic Curves, Cryptosystems Defined over Elliptic Curves, The Discrete Logarithm Problem over Elliptic Curves, The Discrete Logarithm System over Elliptic Curves, The Security of Discrete Logarithm Based EC Systems, Introduction to Goppa codes, The McEliece Cryptosystem, Encryption, Decryption, Discussion, Summary and Proposed Parameters, Heuristics of the Scheme, Not a Signature Scheme, Security Aspects, Guessing *SB* and *PB,* Exhaustive Codewords Comparison, Syndrome Decoding, Guessing *k* Correct and Independent Coordinates, Multiple Encryptions of the Same Message, A Small Example of the McEliece System, Another Technique to Decode Linear Codes, The Niederreiter Scheme

Unit 10: Knapsack Based Systems (3 hrs)

The Knapsack System, The Knapsack Problem, Setting Up the Knapsack System, Encryption 265, Decryption, A Further Discussion, The *L*3-Attack, Introduction, Lattices, A Reduced Basis, The *L*3-Lattice Basis Reduction Algorithm, The Chor-Rivest Variant, Setting Up the System, Encryption, Decryption

Unit 11: Hash Codes & Authentication Techniques (3 hrs)

Introduction, Hash Functions and MAC's, Unconditionally Secure Authentication Codes, Notions and Bounds, The Projective Plane Construction, A Finite Projective Plane, A General Construction of a Projective Plane, The Projective Plane Authentication Code, A-Codes From Orthogonal Arrays, A-Codes From Error-Correcting Codes

Unit 12: Zero Knowledge Protocols and Secret Sharing Systems (2 hrs)

The Fiat-Shamir Protocol, Schnorr's Identification Protocol, Threshold Schemes, Threshold Schemes with Liars, Secret Sharing Schemes, Visual Secret Sharing Schemes

References:

1. Henk C.A. Van Tilborg, *Fundamental of Cryptology*, Kluwer Academic Publishers, 2007.
2. William Stalling, *Cryptography and Network Security*, Pearson Education Asia, 2003.

3. Dhren R Patel, *Information Security, Theory and Practice*, Prentice Hall of India Private Limited, 2008

Course Title: Neural Networks

Course Code: CSc 637

Credit Hours = 3

Nature of the Course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course Objectives: To be familiar different neural networks algorithms and apply it for practical applications.

Unit 1: Introduction (8 hrs)

Introduction and Motivation, Bilogical Neural Network and Simple Models, The Artificial Neuron Model, Hopfield Nets, Energy Functions and Optimization.

Unit 2: Multilayer Networks (10 hrs)

Perceptions and Threshold Logic Machines, Multilayer Networks? Their Variants and Applications, Capacity of Multilayer Networks.

Unit 3: Learning System (14 hrs)

Backpropagation Recurrent Nets, Tree Structured Networks, Unsupervised Learning, Hebbian Learning, Principal Component Analysis Competitive Learning, Feature Mapping.

Unit 4: Recent Trends (13 hrs)

Self Organising Maps Adaptive Resonance Theory, Hardware Relization of ANNs, Recent Trends and Future Directions

References:

1. S Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall of India Private Limited, 2008
2. C M Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 2006
3. R Callan, The Essence of Neural Networks, Prentice Hall of India Private Limited, 2006

Course Title: Principles of Programming Language Full Marks: 60+15

Course No: CSc 641 PassMarks: 30+7.5

Credit Hours = 3

Nature of the Course: Theory+ Tutorial

Course Description: Design, evaluation, implementation issues of programming language. Data types, data abstraction, sequence control, procedural abstraction, parameter passing techniques, scope, storage management, object oriented approach, concurrency, co routines and overview of a few high level programming languages.

Course Objectives:

- Introduce the design and implementation issue of high level programming languages.
- Principles for the evaluation of procedural, object- oriented and concurrent programming languages.

Unit 1:

1.1 History of the development of programming language, attributes, standardization and programming environments. 4hrs

1.2 Language design issue: firmware computers, software simulated computer, syntax, semantic, binding time. 3hrs

1.3 Syntactic elements, program-subprogram structures, source code analysis, synthesis.4hrs

1.4 BNF-grammars, finite state automata, push-down automata. Parsing algorithms, semantic modeling. 5hrs

Unit 2:

2.1 Data types, specification and implementation of data types checking and type conversion, assignment and initialization, structured data types, data structures, records, lists, strings, pointers and programmer -constructed data types. 4hrs

2.2 Abstract data type, information hiding, encapsulation, type equivalence, storage management, static, stack-based, heap based. 3hrs

2.3 Sequence control: tree structure representation, execution time representation, non-arithmetic expression, basic statement. 3hrs

2.4 Subprogram control, parameter passing call by value ,call by reference, etc., static and dynamic scope, block structure, local referencing. 5hrs

Unit 3:

3.1 Inheritance, derived class, abstract classes, objects and messages, abstraction concepts. 4hrs

3.2 Exceptions and exception handlers, co routines, concurrent Executions, guarded commands, tasks, synchronization. 4hrs

3.3 Paradigms and languages (overview): procedural (FORTRAN & C), block structure (PASCAL), object oriented (C++, SMALLTALK), functional LISP), and logic (PROLOG). 6hrs

Recommended Books:

1. T.W. Pratt and M.V. Zelkowitz , Programming Languages: Design and Implementation. Prentice Hall.

2. R.W. Sebesta , Concepts of Programming Languages, Addison Wesley, ISBN : 0-201 385961

Course Title: Thesis Full Marks: 100

Course No: CSc 681 Pass Marks: 50

Credit Hours = 3

Nature of the Course: Dissertation

Course Description: CSc-681 is a thesis paper of full mark 100 offered in the curriculum of the M. Sc. Second year computer science. A student, opting for thesis, prepares a proposal under the supervision of a thesis advisor and defends his/her proposal in the department. Once accepted by the department, a student can start his/her work under the supervision of his/her advisor.

Course Objectives:

- To provide student with skill and knowledge in conduction research on fundamental and application aspects of computer science.
- To train student in developing analytical as well as argumentative skill.

A student who completes and passes his first year course can opt. for the thesis course. The number of the students to be accepted in the thesis work may be limited on the basis of resource person and materials available in the department. A student must complete this course within the academic year. However, a grace period of three months can be extended to complete the work on the recommendation of his/her thesis supervisor.

The M. Sc. thesis in computer science should be an original work in Computer Science or its application. It is expected that a student would demonstrate his ability to conduct an independent investigation. The thesis should be logically presented and clearly written. The completed thesis put in spirally bound copes must be presented to the office of the Head of the Department to facilitate his/her final oral presentation.

The thesis committee appointed by the department will evaluate the student's thesis write-up and oral presentation. The individual committee member will grade the thesis and the average marks allotted by the committee will be the final grade of the thesis works.

Course Title: Graphics and Image Processing

Course Code: CSc 642

Credit Hours = 3

Nature of the Course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course objectives: To be familiar with processing of the images, recognition of the pattern and their applications.

Unit 1: Introduction to digital image processing (4 Hrs)

Digital image representation, Digital image processing: Problems and applications, Elements of visual perception, Sampling and quantization, relationships between pixels

Unit 2: Two-dimensional systems (5 Hrs)

Fourier transform and Fast Fourier Transform, Other image transforms and their properties: Cosine transform, Sine transform, Hadamard transform, Haar transform

Unit 3: Image enhancement and restoration (8 Hrs)

Point operations, contrast stretching, clipping and thresholding, digital negative, intensity level slicing, bit extraction, Histogram modeling: Equalization modification, specification, Spatial operations: Averaging, directional smoothing, median, filtering spatial low pass, high pass and band pass filtering, magnification by replication and interpolation

Unit 4: Image coding and compression (4 Hrs)

Pixel coding: run length, bit plan, Predictive and inter-frame coding

Unit 5: Introduction to pattern recognition and images (3 Hrs)

Unit 6: Recognition and classification (5 Hrs)

Recognition classification, Feature extraction, Models, Division of sample space

Unit 7: Grey level features edges and lines (6 Hrs)

Similarity and correlation, Template matching, Edge detection using templates, Edge detection using gradient models, model fitting, Line detection, problems with feature detectors

Unit 8: Segmentation (3 Hrs)

Segmentation by thresholding, Regions for edges, line and curve detection

Unit 9: Frequency approach and transform domain (3 Hrs)

Unit 10: Advanced Topics (4 Hrs)

Neural networks and their application to pattern recognition, Hopfield nets, Hamming nets, perceptron

Laboratory Work: Developing programs of above features.

References:

1. K. Castlemann, “Digital Image Processing”, Printice Hall of India Ltd., 2006.
2. A. K. Jain, “Fundamental of Digital Image Processing”, Printice Hall of India Pvt. Ltd., 2007.
3. R. C. Gonzalez and P. Wintz, “Digital Image Processing”, Addison-Wesley Publishing, 2000
4. Sing_Tze Bow, M. Dekker, “Pattern Recognition and Image Processing”, 2003
5. M. James, “Pattern Recognition”, BSP professional books, 2000.
6. P. Monique and M. Dekker, “Fundamentals of Pattern Recognition”, 2002.

Course Title: Introduction to Randomized Algorithm Full Marks: 75

Course No: CSc 643 PassMarks: 37.5

Credit Hours = 3

Nature of the Course: Theory+Lab

Synopsis: General introduction to the development of randomized and probabilistic algorithms. Markov chains and random walk. Applications in linear programming, graphs, and counting.

Course Objectives:

- Techniques to cope with intractable and difficult problems by using probabilistic and randomized approaches.
- Development of approximation algorithms based on randomized tools.

Course Contents:

Unit1:

1.1 Introduction to the historical development of randomized algorithms including randomized quick sort. Las Vegas algorithms versus Monte Carlo algorithm. Computation models and probabilistic recurrence. 6hrs

1.2 Game theoretic techniques including tree evaluation , minimax principle. Randomness and non-uniformity. 4hrs

1.3 Randomized selection , two-point sampling, occupancy problems, the Markov and Chebyshev Inequalities. 5hrs

Unit2:

2.1 Tail inequalities, example of routing and parallel computers, introduction to probabilistic methods. Examples including satisfiability, expanding graphs, and routing. 6hrs

2.2 Introduction to Markov chains and random walks. Examples include electrical networks, graph connectivity, and 2-SAT. 5hrs

2.3 Algebraic techniques: polynomial identities, matching in graphs, string equality, finger printing techniques, and pattern matching. 5hrs

Unit3:

3.1 Data structures useful for developing randomized algorithms: random treaps, skip list, and hash tables. 6hrs

3.2 Randomized algorithms for solving geometric problems: convex hull, duality, Delaunay triangulation, binary space partition, diameter of point set, random sampling, linear programming. 3hrs

3.3 Randomized algorithms for graph problems: shortest paths, min-cut, minimum spanning tree. Approximate counting. 5hrs

Recommended Books:

1. Rajiv Motwani and P. Raghavan , "Randomized Algorithms" Cambridge University press. ISBN:0-521-47465-5

Course Title: Evolutionary Computing and Genetic Algorithm Full Marks: 60+15

Course No: CSc 644 PassMarks: 30+7.5

Credit Hours = 3

Nature of the Course: Theory+Lab

Synopsis*:* The difference between traditional algorithms and genetic algorithms. Mathematical foundation : population, mutation, crossover. Data structure for genetic algorithm. Applications in optimization, search, recognition etc.

Course Objective:

- Overview of genetic algorithms, biological terminologies, evolving programs theoretical foundation.
- Application in data analysis, prediction, optimization, computer vision, pattern recognition. Relationship to neural networks.

Course Contents:

Unit1

1.1 Historical development, distinction between traditional algorithms and genetic algorithms, simple example of genetic algorithm, the appeal of evolution, building block hypothesis. 6hrs

1.2 Reproduction, crossover, and mutation, schemata, objective function and fitness form, fitness scaling coding, data structures. 4hrs

1.3 De Jung and function optimization, improvement in basic techniques, discretization. 5hrs

Unit2

2.1 Advanced operators, inversion and reordering operators, micro-operators, multi-objective optimization, knowledge –based techniques. 7hrs

2.2 Genetic based machine learning, classifier system, rule and message system, the risk of GBML, Smith's poker player. 6hrs

Unit3

3.1 Genetic algorithms in scientific models: modeling interaction between learning and evolution, modeling sexual selection, modeling echo system, measuring evolutionary activity. 9hrs

3.2 Implementation of genetic algorithms: encoding a problem for genetic algorithm, adapting the encoding, selection methods, genetic operators, parameters for genetic algorithms.

3.3 Application of genetic algorithm for classical problems: traveling sales person problem, path planning, set-cover, and satisfiabiliy. Fusing genetic algorithms and neural nets. 8hrs

Recommended Books:

1. David E. Goldberg, "**Genetic algorithms in Search, optimization, and machine learning.**",Addison Wesley, and Benjamin Cummings. ISBN: 0-201-15767-5.

2. Malanie Mitchell, "**An introduction to Genetic algorithms"** The MIT pres, 1998, ISBN: 0-262-63185-7.

Course Title: Data Communication and Networking

Course Code: CSc 645

Credit Hours = 3

Nature of course: Theory + Lab

Full Marks: 60+15

Pass Marks: 30+7.5

Course objectives:

1. To provide the student with basic knowledge of how different types of Networks can be physically interconnected, including the major protocols employed in Internetworking.
2. To provide the students with basic knowledge of how the major internetworking technology are implemented.
3. To provide the students with basic knowledge of major application layer uses and Internet technologies and application.

Unit 1: Introduction (3 Hrs)

Data communication concept, Types of networks: LAN, MAN, WAN and Internet, Network topology: Star, Bus, Ring and Mesh, Protocol and standards, Layered architecture and its need, Network model: OSI Reference model and TCP/IP reference model.

Unit 2: Physical Layer (6 Hrs)

Signals: Analog Signals and Digital Signals, Modulation: Analog Modulation and Digital Modulations, Multiplexing: FDM, TDM, and WDM, Encoding: Polar, RZ, NRZ and Manchester, Switching: Packet Switching and Circuit Switching, Transmission Media: Guided and Unguided media, DCE- DTE interfacing.

Unit 3: Data Link Layer (10 Hrs)

Flow control: Stop and Wait, and sliding window, Error Control: Error Detection and Error correction, Multiple Accesses: ALOHA, CSMA, CSMA/CD and Collision free Protocol, Data Link

Protocol: HDLC, PPP, Ethernet LAN (IEEE 802.3) Token Ring (IEEE 802.5), WLAN (IEEE 802. 11(a) (b) (g)) and fast Ethernet FDDI; WMAN (IEEE 802.16), Circuit Switched Service: N-ISDN, B-ISDN, Packet Switched Service: X-25, Frame relay, ATM, xDSL and SONET, Network Components: Hub, Repeaters, Bridges, Routers and Gateway

Unit 4: Network Layer (9 Hrs)

Internetworking concept, Internet Protocol, IP Addressing, Sub netting, Supper netting, CIDR, ARP, RARP, ICMP, NAT and IPv6, Routing Algorithm: Static and Dynamic, Link state routing, Distance vector Routing and Congestion Control, Routing Protocol: RIP, OSPF, BGP and EIGRP, Quality of Service: Traffic Shaping, Leaky Bucket Algorithm and Token Bucket Algorithms.

Unit 5: Transport Layer (3 Hrs)

Port Number Concept, Connection less Transport: UDP, Connection Oriented Transport: TCP, TCP: Connection Establishment Process and Connection Termination Process, Flow control.

Unit 6: Application Layer (6 Hrs)

Client / Server interaction, Domain Name System DNS, DHCP, FTP, HTTP, SSH, SMTP, POP, INAP, Telnet and E-mail.

Unit 7: Network Security (3 Hrs)

Why Networks need Security, Types of Security Threat, Cryptography: Symmetric key Cryptography and Public key Cryptography, Communication Security: IPSec, VPN, E-mail Security: PGP, Web Security: SSL (Secure Socket Layer), Digital Signature and Fire wall

Unit 8: Network Management (3 Hrs)

What is Network Management, Infrastructure for Network Management, The Internet Standard Management Frame work: SNMPv1, v2 and v3.

Unit 9: Multimedia (2 Hrs)

VoIP, Video Compression, JPEG and MPEG.

Unit 10: Lab Work

Clamping, File Sharing and Security, Printer Sharing, Configuring DHCP Server and Client, Configuring DNS server and Client, Configuring Static route and Dynamic route, Use Telnet, SSH, Trace route Command, Web Server Configuration, Mail Server Configuration, Network monitoring and Control through SNMP, Fire Wall Policy implementation

References:

1. Computer Networks, By: Andrew S. Tanenbaum
2. Data and Computer Communication, By: W.Stalling
3. Data Communication and Networking, By: B.A. Forauzan
4. Computer Networking a Top- Down Approach, Featuring the Interenet, By: J.F. Kurose & K.W. Ross
5. Communication Networks, By: Alberto Leon-Garcia

Course Title: Scheduling Algorithms

Course Code: CSc 646

Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60 + 15

Pass Marks: 30 + 7.5

Course description: This course is about scheduling algorithms and complexity analysis of dedicated and parallel processors. It is designed to provide the students with in-depth knowledge of theoretical as well as practical background and their implementations. The approach of the course will be mathematical modeling, numerical methods, algorithmic analysis and implementations.

This course mainly focuses in theoretical computer science and scientific computing. It relates with operations research methods, optimization techniques, numerical mathematics and operations scheduling.

Course objectives: The main objectives of this course are enabling the students:

1. To broaden their knowledge towards modeling techniques, numerical methods and algorithms.
2. To realize the importance of various aspects of scheduling algorithms in every day experiences in the industry like computer, manufacturing and IT.
3. To implement the knowledge of scheduling algorithms in real life problems.
4. To be motivated and prepared for the study of more advanced field of scheduling algorithms and complexity analysis.

Unit 1: Computational Complexity of Scheduling Problems (7 Hrs)

Classification of scheduling problems, applications of some combinatorial optimization algorithms to solve some scheduling problems, computational complexity and reduction relations between scheduling problems, techniques of solving NP-hard problems.

Unit 2: Single machine scheduling problems (8 Hrs)

Mini-max criteria, maximum lateness and total weighted completion times, introduction to problems related to weighted number of late jobs and total weighted tardiness.

Unit 3: Parallel Machine Problems (7 Hrs)

Problems with identical parallel machines, introduction to problems with uniform parallel machines and unrelated parallel machines.

Unit 4: Open shop Scheduling Problems (8 Hrs)

Disjunctive graph model, block-matrices model, problems with arbitrary processing times, problems with unit processing times.

Unit 5: Flow and Job Shop Problems (7 Hrs)

Flow shop problems minimizing Makespan, job shop problems with two machines, job shop problems with two jobs.

Unit 6: Due Date Scheduling (8 Hrs)

Problems with lateness objectives, problems with earliness and tardiness objectives.

References:

1. P. Brucker – Scheduling Algorithms, Springer-Verlag.
2. J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, J. Weglarz – Scheduling Computer and Manufacturing Processes, Springer-Verlag.
3. M. Pinedo, X. Chao - Operations Scheduling with Applications in Manufacturing and Services, Irwin McGraw-Hill (1999) Springer-Verlag.

Course Title: Data Warehousing and Data Mining
Course Code: CSc 647
Credit Hours = 3

Nature of the course: Theory + Lab

Full Marks: 60 + 15

Pass Marks: 30 + 7.5

Course objectives: The data warehousing part of module aims to give students a good overview of the ideas and techniques which are behind recent development in the data warehousing and online analytical processing (OLAP) fields, in terms of data models, query language, conceptual design methodologies, and storage techniques. Data mining part of the model aims to motivate, define and characterize data mining as process; to motivate, define and characterize data mining applications.

Unit 1:(5 Hrs)
Introduction : Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Major issues in Data Mining, Data Warehouse and OLAP Technology for Data Mining Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Further Development of Data Cube Technology, From Data Warehousing to Data Mining.

Unit 2: (4 Hrs)
Data Pre-processing: Needs Pre-processing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

Unit 3:(5 Hrs)
Data Mining Primitives, Languages, and System Architectures : Data Mining Primitives, Data Mining Query Languages, Designing Graphical User Interfaces Based on a Data Mining Query Language Architectures of Data Mining Systems,

Unit 4:(5 Hrs)
Concepts Description: Characterization and Comparison: Data Generalization and Summarization-Based Characterization, Analytical Characterization: Analysis of Attribute Relevance, Mining Class Comparisons: Discriminating between Different Classes, Mining Descriptive Statistical Measures in Large Databases.

Unit 5: (8 Hrs)
Mining Association Rules in Large Databases : Association Rule Mining, Mining Single-Dimensional Boolean Association Rules from Transactional Databases, Mining Multilevel Association Rules from Transaction Databases, Mining Multidimensional Association Rules from Relational Databases and Data Warehouses, From Association Mining to Correlation Analysis, Constraint-Based Association Mining.

Unit 6: (6 Hrs)

Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Back propagation, Classification Based on Concepts from Association Rule Mining, Other Classification Methods, Prediction, Classifier Accuracy.

Unit 7: (6 Hrs)

Cluster Analysis Introduction : Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Outlier Analysis.

Unit 8: (8 Hrs)

Mining Complex Types of Data: Multi Dimensional Analysis and Descriptive Mining of Complex, Data Objects, Mining Spatial Databases, Mining Multimedia Databases, Mining Time-Series and Sequence Data, Mining Text Databases, Mining the World Wide Web.

References:

1. Data Mining – Concepts and Techniques - JIAWEI HAN & MICHELINE KAMBER Harcourt India.
2. Data Mining Introductory and advanced topics –MARGARET H DUNHAM, PEARSON EDUCATION
3. Data Mining Techniques – ARUN K PUJARI, University Press.
4. Data Warehousing in the Real World – SAM ANAHORY & DENNIS MURRAY. Pearson Asia.
5. Data Warehousing Fundamentals – PAULRAJ PONNAIAH WILEY STUDENT EDITION.
6. The Data Warehouse Life cycle Tool kit – RALPH KIMBALL WILEY STUDENT EDITION.