National Institute of Technology Calicut
Department of Computer Engineering

Curriculum for Master of Computer Applications (MCA)
(From 2006 Admissions)

Semester 1

	Code	Title	L	T	P	C
1	MAG501	Discrete Mathematics	3	1	0	3
2	MAG502	Probability and Statistics	3	1	0	3
3	CSU101	Computer Programming	3	0	0	3
4	CSU202	Logic Design	3	0	2	4
5	SHU111	Professional Communication	3	0	0	3
6	CSM591	Programming Lab	0	0	5	3
		Total Credits				**19**

Semester 2

	Code	Title	L	T	P	C
1	MAG503	Graph Theory and Combinatorics	3	1	0	3
2	CSU215	Computer Organisation	3	0	2	4
3	CSU213	Database Management Systems	3	0	0	3
4	CSU211	Formal Languages and Automata	3	0	0	3
5	CSU230	Program Design	3	0	2	4
6	CSU296	DBMS Lab	0	0	3	2
		Total credits				**19**

Semester 3

	Code	Title	L	T	P	C
1	MAG505	Number Theory & Applications	3	1	0	3
2	MEG501	Principles of Industrial Management	3	0	0	3
3	CSU203	Data Structures and Algorithms	3	0	0	3
4	CSU304	Computer Networks	3	0	0	3
5		Elective	3	0	0	3
6	CSU291	Data Structures Lab	0	0	5	3
		Total credits				**18**

Semester 4

	Code	Title	L	T	P	C
1	CSU313	Operating Systems	3	0	0	3
2	CSU303	Compiler Construction	3	0	0	3
3	CSU321	Software Engineering	3	0	0	3
4	MEG502	Principles of Financial Management	3	0	0	3
5		Elective	3	0	0	3
6	CSU392	Compiler Lab	0	0	5	3
		Total credits				**18**

Semester 5

	Code	Title	L	T	P	C
1	MAG504	Numerical Analysis & Optimization Techniques	3	1	0	3
2		Elective	3	0	0	3
3		Elective	3	0	0	3
4		Elective	3	0	0	3
5		Elective	3	0	0	3
6	CSM581	Seminar	0	0	3	1
		Total credits				**16**

Semester 6

	Code	Title	L	T	P	C
1	CSM599	Project	-	-	-	15
		Total credits				**15**

Minimum Requirement

A student should have earned a total of at least 105 credits.

Notes:

- The electives can be chosen from the list of elective courses offered by the department for the B.Tech / MCA Programmes.
- Final year students satisfying the standard requirements can credit elective courses of the M.Tech program, with consent of the department.
- Students can also credit global electives, with permission from the department.

List of Elective Courses

Sl. No	Code	Title	L	T	P	C
1	CSU339	Advanced Data Structures	3	0	0	3
2	CSU358	Communication and Information Theory	3	0	0	3
3	CSU301	Design and Analysis of Algorithms	3	0	0	3
4	CSU363	Computational Intelligence	3	0	0	3
5	CSU371	Logic for Computer Science	3	0	0	3
6	CSU334	Web Programming	3	0	0	3
7	CSU331	Computer Graphics and Multimedia	3	0	0	3
8	CSU341	Distributed Computing	3	0	0	3
9	CSU343	Embedded System Design	3	0	0	3
10	CSU411	Computer Security	3	0	0	3
11	CSU353	Mobile Communication Systems	3	0	0	3
12	CSU354	Electronic Commerce	3	0	0	3
13	CSU333	Object Oriented-Analysis & Design	3	0	0	3
14	CSU431	Advanced Database Management Systems	3	0	0	3
15	CSU441	Advanced Computer Architecture	3	0	0	3
16	CSU352	Coding Theory	3	0	0	3
17	CSU356	Mobile Computing	3	0	0	3
18	CSU361	Image Processing	3	0	0	3
19	CSU362	Pattern Recognition	3	0	0	3
20	CSU364	Natural Language Processing	3	0	0	3
21	CSU373	Computational Complexity	3	0	0	3
22	CSU471	Advanced Topics in Algorithms	3	0	0	3
23	CSU472	Quantum Computation	3	0	0	3
24	CSU305	Theory of Computation	3	0	0	3
25	CSU315	Computer Hardware	3	0	0	3
26	MAU333	Stochastic Processes for Engineers	3	1	0	3
27	MAG521	Simulation and Modelling	3	1	0	3
28	MAU329	Fuzziness and Soft Computing	3	1	0	3
29	MAU325	Decision Modelling Technique	3	1	0	3
30	MAU342	Topics in Algebra	3	1	0	3
31	MAU326	Integer Optimization and Network Models	3	1	0	3
32	MAU336	Design and Analysis of Experiments	3	1	0	3
33	MAU335	Forecasting Techniques	3	1	0	3
34	MAU324	Fuzzy Optimization	3	1	0	3

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
DEPARTMENT OF COMPUTER ENGINEERING

Master of Computer Applications (MCA)
(From 2006 Admissions)

Syllabi of the
Courses Offered by the Department of Computer Engineering

PART I : CORE COURSES

CSU 101 COMPUTER PROGRAMMING

Pre-requisite: NIL

L	T	P	C
3	0	0	3

Module I

Introduction to computers and software	**(3 Hours)**
Problem solving, algorithm design, and algorithm analysis (mention only)	**(3 Hours)**
Design methodologies:	**(6 Hours)**
Stepwise refinement: Modules and Interfaces.	
Object oriented methodology: Encapsulation, Inheritance, Polymorphism	

Module II

Programming language concepts and constructs: Expressions, Statements, selection, repetition. **(11 Hours)**

Module III

Functions, recursion, I/O mechanisms (Exceptions optional) **(10 Hours)**

Module IV

Data types: Primitive types and structured types	**(4 Hours)**
Coding practices:	**(3 Hours)**
Indentation guidelines, naming conventions, documenting code, debugging	
Testing: Verification methods, test data selection.	**(2 Hours)**

Note: Programming language C++ / Java may be used as a vehicle to achieve the goal.

Text Books:

1.Bruce Eckel, *Thinking in Java*, 3/ed, Available online at www.bruceeckel.com
2. Bruce Eckel, *Thinking in C++,* 2/ed. Vol I and II, Available online at www.bruceeckel.com

References:

1. Robert Lafore, *Object Oriented Programming in Turbo C++,* The Waite Group's, Galgotia Publications Pvt. Ltd. 2000.
2. Rebecca Thomas, Lawrence R Rogers, Jean L Yates, *Advanced Programmer's Guide to UNIX System V*, McGraw Hill International Edition, Computer Science Series.
3. Patrick Naughton, Herbert Schildt, *Java ^{TM} 2: The Complete Reference*, Tata McGraw-Hill Publishing Company Ltd. 3/ed
4. Danny Kalev, *The ANSI/ISO C++ Professional Programmer's Handbook*, PHI 2000.

CSU 202 LOGIC DESIGN

Pre-requisite: CSU 101 Computer Programming

L	T	P	C
3	0	2	4

Module I **(10 + 5 Hours)**

Number Systems and codes, Boolean algebra: postulates and theorems, constants, variables and functions, switching algebra, *Boolean functions and logical operations*, *Karnaugh map*: prime cubes, minimum sum of products and product of sums, Quine-McClusky algorithm, Combinational Logic: analysis and design of combinational logic circuits, parallel adders and look-ahead adders, comparators, decoders and encoders, code conversion, multiplexers and demultiplexers, parity generators and checkers, ROMs,

Module II **(10 + 5 Hours)**

PLAs, .PLA minimization, PLA folding, design for testability, Counters and shift registers: excitation tables, ripple counters, synchronous counters, up-down counters, design of sequential circuits, shift registers and their applications. Clock mode sequential machines.

Module III **(11 + 10 Hours)**

Microprocessor architecture: real mode and protected mode memory addressing, memory paging. Addressing modes: data addressing, program memory addressing, stack memory addressing. Data movement instructions, Arithmetic and logic instructions, Program control instructions, Programming the microprocessor: modular programming, using keyboard and display, data conversions, disk files, interrupt hooks, using assembly language with C/C++.

Module IV **(11 + 8 Hours)**

Memory interface: memory devices, address decoding, 16 bit (8086), 32 bit (80486) and 64 bit (Pentium) ,Hardware architecture for embedded systems-processor-memory-latches and buffers-display unit-16 and 32 bit processors. Memory interfaces, dynamic RAM. I/O interface: port address decoding, PPI, 8279 interface, 8254 timer interface, 16550 UART interface, ADC/DAC interfaces, Interrupts- Interrupt controller, DMA Controller.

References:

1. N. N. Biswas, *Logic Design Theory*, Prentice Hall of India, New Delhi, 1993.
2. T. L. Floyd, *Digital Fundamentals*, 3/e, Universal Book Stall, New Delhi, 1986.
3. B. B. Brey*, The Intel Microprocessors 8086 to Pentium: Architecture, Programming and Interface*, 6/e, Prentice Hall of India, New Delhi, 2003.
4. Programming for embedded systems Dream Software team , Willey 2002
5. H. P. Messmer, *The Indispensable PC Hardware Book*, 3/e, Addison Wesley, 1997.
6. A. K. Ray, and K. M. Bhurchandi, Advanced Microprocessors and Peripherals, Tata McGraw Hill, 2000.
7. D. V. Hall, *Microprocessors and Interfacing: Programming and Hardware*, 2/e, Tata McGraw Hill, New Delhi, 1992.

CSM 591 PROGRAMMING LAB

Pre-requisite: CSU 101 Computer Programming

L	T	P	C
0	0	5	3

Module I (5 Lab Sessions)

HCF (Euclid's algorithm) and LCM of given numbers - find mean, median and mode of a given set of numbers – Conversion of numbers from binary to decimal, hexadecimal, octal and back – evaluation of functions like e^x, sin$x$, cos$x$ etc. for a given numerical precision using Taylor's series – testing whether a given number is prime.

Module II (3 Lab Sessions)

String manipulation programs: sub-string search, deletion – lexicographic sorting of a given set of strings – generation of all permutations of the letters of a given string using recursion..

Module III (3 Lab Sessions)

Matrix operations: Programs to find the product of two matrices – inverse and determinant (using recursion) of a given matrix – solution to simultaneous linear equations using Jordan elimination.

Module IV (3 Lab Sessions)

Files: Use of files for storing records with provision for insertion, deletion, search, sort and update of a record.

References

1. H. Schildt, *C: The Complete Reference*, 4/e, Tata McGraw Hill, 2000.
2. H. H. Tan and T. B. D'Orazio, *C Programming for Engineering & Computer Science*, McGraw Hill, 1999.
3. T. H. Cormen, C. E. Lieserson, R. L. Rivest, *Introduction to Algorithms*, PHI, 1998.

CSU 215 COMPUTER ORGANISATION

Pre-requisite: CSU 202 Logic Design

L	T	P	C
3	0	2	4

Module I (11 + 10 Hours)

Computer abstraction and technology: basic principles, hardware components, Measuring performance: evaluating, comparing and summarizing performance.

Instructions: operations and operands of the computer hardware, representing instructions, making decision, supporting procedures, character manipulation, styles of addressing, starting a program.

Module II (10 + 6 Hours)

Computer arithmetic: signed and unsigned numbers, addition and subtraction, logical operations, constructing an ALU, multiplication and division, floating point representation and arithmetic.

Module III (10 + 6 Hours)

The processor: building a data path, simple and multicycle implementations, microprogramming, exceptions, Case study: Pentium Pro implementation.

Module IV (11 + 6 Hours)

Memory hierarchy: caches, cache performance, virtual memory, common framework for memory hierarchies, Case study: Pentium Pro memory hierarchy.

Input/output: I/O performance measures, types and characteristics of I/O devices, buses, interfaces in I/O devices, design of an I/O system.

References:

1. D. A. Pattersen and J. L. Hennesy, *Computer Organisation and Design: The Hardware/ Software Interface*, 3/e, Morgan Kaufman, Singapore, 2004.

2. V. P. Heuring and H. F. Jordan, *Computer System Design and Architecture*, Addison Wesley, New Delhi, 1997.

CSU 213 DATABASE MANAGEMENT SYSTEMS

Pre-requisite: CSU 212 Computational Combinatorics / MAG 501 Discrete Mathematics

L	T	P	C
3	0	0	3

Module I (10 hours)

Database System concepts and architecture, Data modeling using Entity Relationship (ER) model and Enhanced ER model, Specialization, Generalization, Data Storage and indexing, Single level and multi level indexing, Dynamic Multi level indexing using B Trees and B+ Trees.

Module II (11 hours)

The Relational Model, Relational database design using ER to relational mapping, Relational algebra and relational calculus, Tuple Relational Calculus, Domain Relational Calculus, SQL.

Module III (11 hours)

Database design theory and methodology, Functional dependencies and normalization of relations, Normal Forms, Properties of relational decomposition, Algorithms for relational database schema design.

Module IV (10 hours)

Transaction processing concepts, Schedules and serializability, Concurrency control, Two Phase Locking Techniques, Optimistic Concurrency Control, Database recovery concepts and techniques, Introduction to database security.

References:

1. Elmasri, Navathe, Somayajulu, Gupta, *Fundamentals of Database Systems*, IE, Pearson Education, 2006
2. Ramakrishnan R. & Gehrke J., *Database Management Systems*, Third edition, 2003, McGraw Hill
3. S K Singh, *Database Systems-Concepts, Design and Applications*, Pearson Education, 2006

CSU 211 FORMAL LANGUAGES AND AUTOMATA

Pre-requisite: NIL

L	T	P	C
3	0	0	3

Module I (12 Hours)

Basic concepts of Languages, Automata and Grammar.

Regular Languages - Regular expression - finite automata equivalence, Myhill Nerode theorem and DFA State Minimization, Pumping Lemma and proof for existence of non-regular languages.

Module I (12 Hours)

Context Free languages, CFL-PDA equivalence, Pumping Lemma and proof for existence of non- Context Free languages, CYK Algorithm, Deterministic CFLs.

Module III (9 Hours)

Turing Machines: recursive and recursively enumerable languages, Universality of Turing Machine, Church Thesis

Module IV (9 Hours)

Chomsky Hierarchy, Undecidability, Reducibility.

References:

1. M. Sipser, Introduction to the Theory of Computation, Thomson, 2001.

2. Hopcroft J. E., Rajeev Motwani, and Ullman J. D., Introduction to Automata Theory, Languages and Computation, Pearson Education Asia,2001.

3. J. C. Martin, Introduction to Languages and the Theory of Computation, Mc Graw Hill, 2002.

4. P. Linz, Introduction to Formal Languages and Automata, Narosa,1998

CSU 230 PROGRAM DESIGN

Pre-requisites: CSU 101 Computer Programming, Knowledge of Graphs and Trees

L	T	P	C
3	0	2	4

Module1: (10 + 5 Hours)
Review of Programming Constructs- Conditional and Iterative constructs, Data types, Control Structures, Functions, Parameter passing- calling conventions, Recursion, Asymptotic notation for complexity analysis.

Module2: (11 + 12 Hours)
Pointers and dynamic memory allocation, Abstract Data Types, Lists, Stacks, Queues, Trees, Search Trees and traversal algorithms, Heaps and Priority queues.

Module3: .(11 + 7 Hours)
Searching - Linear and Binary, Sorting- Insertion and Selection sorting, Divide an conquer, Quick sort, Merge Sort, Heap Sort, External Sorting.

Module4: (10 + 4 Hours)
Memory Management, Garbage collection algorithms, , Storage allocation for objects with mixed sizes, Buddy systems, Storage compaction.

References:

1. Aho A.V., Hopcroft J.E., and Ullman J.D., *Data Structures and Algorithms*, Pearson Education, New Delhi, 1983.
2. Cormen T.H., Leiserson C.E, Rivest R.L. and Stein C, *Introduction to Algorithms*, Prentice Hall India, New Delhi, 2004
3. Sahni S., *Data Structures, Algorithms, and Applications in C++,* Mc Graw Hill, Singapore, 1998.
4. Wirth N., *Algorithms +Data Structures = Programs*, Prentice Hall India, New Delhi, 1976.

CSU 296 DBMS LAB

Pre-requisite: Knowledge of database design and applications

L	T	P	C
0	0	3	2

Lab 1: Familiarization of the MySQL database – creation and manipulation of tables. (3 Hours)

Lab 2: Analyze a given situation, develop an ER model and convert the ER model to Relational model. Implement the database using MySQL and manipulate the tables using SQL commands. (6 Hours)

Lab 3: Development of a 2 tier application using a suitable front end. (6 Hours)

Lab 4: Development of a 3 tier application involving manipulation of web databases. (6 Hours)

Lab 5: Implementation of B Trees and B+ Trees. (6 Hours)

Lab 6: Implementation of a single user RDBMS called 'Minibase'
Write codes for both logical layer and physical layer. (15 Hours)

References:

1. Elmasri, Navathe, *'Fundamentals of Database Systems',* 4/e, Pearson Education
2. Reghu Ramakrishnan*, Databse Management Systems,* McGrawHill
3. *http://www.cs.wisc.edu/coral/minibase/minibase.html*

CSU 203 DATA STRUCTURES AND ALGORITHMS

Pre-requisites: CSU 230 Program Design
CSU 212 Computational Combinatorics / MAG 503 Graph Theory and Combinatorics

L	T	P	C
3	0	0	3

Module I **(10 Hours)**
Review of basic data structures, Representation of sets, Set implementation using bit string, linked list.
Hashing – Introduction to simple hash functions, resolution of collisions, Disjoint sets- representations, Union, Find algorithms.

Module II **(12 Hours)**
Graphs: Representation of graphs, Depth First and Breadth First Traversals, Strong connectivity. Minimum Cost Spanning Tree algorithms- Prim's, Kruskal's. Path Finding algorithms – Single Source shortest path and All Pairs Shortest Path algorithms.

Module III **(10 Hours)**
Balanced Binary Search trees: Red-Black trees- Properties of Red Black trees, Rotations, Insertion, Deletion.
B-Trees- Basic operations on B-Trees – Insertion and Deletion.

Module IV **(10 Hours)**
Binomial Heaps- Binomial trees and Binomial heaps, Operations on Binomial Heaps.
Fibonacci heaps- Structure of Fibonacci heaps, Mergeable heap operations.

Text Book:
Cormen T.H., Leiserson C.E, Rivest R.L. and Stein C, *Introduction to Algorithms*, Prentice Hall India, New Delhi, 2004

References:

1. Aho A.V., Hopcroft J.E., and Ullman J.D., *Data Structures and Algorithms*, Pearson Education, New Delhi, 1983.
2. Sahni S., *Data Structures, Algorithms, and Applications in C++,* Mc Graw Hill, Singapore, 1998.
3. Aho A. V., Hopcroft J. E. & Ullman J. D., *The Design And Analysis of Computer Algorithms*, Addison Wesley

CSU 304 COMPUTER NETWORKS

Pre-requisite: CSU 203 Data Structures and Algorithms / CSU 230 Program Design

L	T	P	C
3	0	0	3

Module I   **(10 hours)**
Computer Networks and Internet, the network edge, the network core, network access, delay and loss, protocol layers and services, Application layer protocols, socket programming, content distribution.

Module II   **(10 hours)**
Transport layer services, UDP and TCP, congestion control, Network layer services, routing, IP, routing in Internet, router, IPV6, multicast routing, mobility.

Module III   **(10 hours)**
Link layer services, error detection and correction, multiple access protocols, ARP, Ethernet, hubs, bridges, switches, wireless links, PPP, ATM.

Module IV   **(12 hours)**
Multimedia networking, streaming stored audio and video, real-time protocols, security, Cryptography, authentication, integrity, key distribution, network management.

References:

1. J. F. Kurose and K . W. Ross, *Computer Networking: A Top-Down Approach Featuring Internet*, 3/e, Pearson Education, 2005.
2. Peterson L.L. & Davie B.S., *Computer Networks, A systems approach*, 3/E, Harcourt Asia, 2003.
3. Andrew S. Tanenbaum, *Computer Networks*, 3/E, PHI, 1996.
4. IEEE/ACM Trans on Networking

CSU 291 DATA STRUCTURES LAB

Prerequisite: CSU 230 Program Design

L	T	P	C
0	0	5	3

Module I (20 Hours)

Stack and Queue: Implementation using arrays and Linked lists
Searching Methods: Binary search and Hashing
Sorting: Recursive implementation of Quick Sort and Merge Sort

Module II (15 Hours)

Binary Search Tree: Implementation with insertion, deletion and traversal
Infix Expression Evaluation: Using expression tree

Module II (20 Hours)

Graph Search Algorithms: DFS and BFS on a connected directed graph
Minimal Spanning Tree: Implementation of Kruskal's and Prim's Algorithms
Shortest Path Algorithms: Dijkstra and Floyd Warshall Algorithms

Module II (15 Hours)

Disjoint Set operations: Union and Find using rank and path compression.
Applications of Heap: Priority Queue and Heap Sort.

References:

1. T. H. Cormen, C. E. Lieserson, R. L. Rivest, *Introduction to Algorithms*, PHI, 1998
2. S. Sahni, *Data structures, Algorithms, and Applications in C++*, McGraw Hill, 1998

CSU 313 OPERATING SYSTEMS

Pre-requisite: CSU 203 Data Structures and Algorithms

L	T	P	C
3	0	0	3

Module I (10 Hours)

Review of operating system strategies - resources - processes - threads - objects - operating system organization - design factors - functions and implementation considerations - devices - characteristics - controllers - drivers - device management - approaches - buffering - device drivers - typical scenarios such as serial communications - storage devices etc

Module II (12 Hours)

Process management - system view - process address space - process and resource abstraction - process hierarchy - scheduling mechanisms - various strategies - synchronization - interacting & coordinating processes - semaphores - deadlock - prevention - avoidance - detection and recovery

Module III (10 Hours)

Memory management - issues - memory allocation - dynamic relocation - various management strategies - virtual memory - paging - issues and algorithms - segmentation - typical implementations of paging & segmentation systems

Module IV (10 Hours)

File management - files - implementations - storage abstractions - memory mapped files - directories and their implementation - protection and security - policy and mechanism - authentication - authorization - case study of unix kernel and microsoft windows NT (concepts only)

References

1. Silberschatz, Galvin, Gagne, *Operating System Principless*, 7/e, 2006, John Wiley
2. William Stallings, *Operating Systems*, 5/e, Pearson Education
3. Crowley C., *Operating Systems- A Design Oriented Approach*, Tata McGraw Hill, New Delhi
4. Tanenbaum A. S., *Modern Operating Systems*, Prentice Hall, Pearson Education
5. Gary J. Nutt, *Operating Systems - A Modern Perspective*, Addison Wesley

CSU 303 COMPILER CONSTRUCTION

Pre-requisites: CSU 203 Data Structures and Algorithms
CSU 211 Formal Languages and Automata

L	T	P	C
3	0	0	3

Module I (6 hours)
Introduction to Programming language translation. Lexical analysis: Specification and recognition of tokens.

Module II (12 hours)
Syntax analysis: Top-down parsing-Recursive descent and Predictive Parsers. Bottom-up Parsing- LR (0), SLR, and LR (1) Parsers.

Module III (16 hours)
Semantic analysis: Type expression, type systems, type checking, and symbol tables.
Intermediate code generation: Intermediate languages. Intermediate representation-Three address code and quadruples. Syntax-directed translation of declarations, assignments statements, conditional constructs, and loops constructs.

Module IV (8 hours)
Runtime Environments: Storage Organization, activation records. Introduction to machine code generations and code optimizations.

References:

1. Aho A.V., Sethi R, and Ullman J.D. Compilers: Principles, Techniques, and Tools. Addison-Wesley, 1986.
2. Appel A.W, and Palsberg J. Modern Compiler Implementation in Java. Cambridge University Press, 2002.

CSU 321   SOFTWARE ENGINEERING

Pre-requisite: CSU 203 Data Structures & Algorithms

L	T	P	C
3	0	0	3

Module I   **(8 Hours)**
Introduction: Software process and the role of modeling and analysis, software architecture, and software design.

Module II   **(11 Hours)**
Software Modelling and Analysis: Analysis modeling and best practices, traditional best practice diagrams such as DFDs and ERDs, UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns.

Module III   **(11 Hours)**
Software Architecture: Architectural styles, architectural patterns, analysis of architectures, formal descriptions of software architectures, architectural description languages and tools, scalability and interoperability issues, web application architectures, case studies.

Module IV   **(12 Hours)**
Software Design : Design best practices, design patterns, design case studies, object oriented frameworks, distributed objects, object request brokers, case studies.

References:

1. G. Booch, J. Rumbaugh, and I. Jacobson, I. The Unified Modeling Language User Guide. Addison-Wesley, 1999 .
2. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995 .
3. F. Buschmann et al. Pattern Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley and Sons, 1996.
4. M. Shaw and D. Garlan. Software Architecture: Perspectives on an Emerging Discipline. Prentice-Hall, 1996

CSU 392 COMPILER LAB

Pre-requisite: Knowledge of Compiler Design and Implementation

L	T	P	C
0	0	5	3

Module I ***(7 Hours)***

Generation of lexical analyzer using tools such as LEX.

Module II ***(25 Hours)***

Generation of parser using tools such as YACC. Creation of Symbol tables.

Module III **(20 Hours)**

Semantic Analysis and intermediate code generation.

Module IV **(18 Hours)**

Generation of target code.

References

4. Holub A. I., *Compiler Design in C,* Prentice Hall India

5. Appel A.W., *Modern Compiler Implementation in C,* Cambridge University Press

CSM 581 SEMINAR

Pre-requisite: NIL

L	T	P	C
0	0	3	1

Each student is expected to present a seminar on a topic of current relevance in computer science and engineering – they have to refer papers from standard journals like ACM, IEEE, JPDC, IEE etc. – at least three cross references must be used – the seminar report must not be the reproduction of the original paper.

CSM 599 PROJECT

Pre-requisite: CSU 321 Software Engineering

Duration	1 Semester
Credits	15

The project is for the entire duration of the sixth semester. Each student is expected to develop a complete product. The design and development may include software and /or hardware. The project involves the design, development, testing, and installation of the product. The product should have user manuals. There will be regular evaluations of the progress of the work by the guide and the evaluation committee. A detailed Project Report is to be submitted at the end of the semester.

PART 2 :   ELECTIVE COURSES

CSU 339 ADVANCED DATA STRUCTURES

Pre-requisite: CSU 203 Data Structures and Algorithms

L	T	P	C
3	0	0	3

Module I   (10 hours)

Review of elementary data structures. Advanced Trees – Red Black Trees, AVL Trees, Optimal Binary Search Trees, Splay Trees.

Module II   (10 hours)

B Trees, Tries, Binary Heaps, Priority Queues, Binomial Heaps, Fibonacci Heaps.

Module III   (10 hours)

Disjoint set representation – Path compression algorithm – Graph algorithms, Connected components, topological sort, Minimum spanning tree, Algorithms of Kruskal and Prim,

Module IV   (12 hours)

Single-source shortest paths – Dijkstra's algorithm, Bellman-Ford Algorithm. All-Pairs shortest paths – Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs. Maximum Flow - Flow networks, Ford-Fulkerson Method.

References:

1. Cormen T.H., Leiserson C.E, and Rivest R.L., *Introduction to Algorithms*, Prentice Hall India, New Delhi, 1990.
2. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall India, New Delhi, 1976.
3. Sartaj Sahni, Data Structures, Algorithms and Applications in C++, Universities Press, 2005.

CSU 358 COMMUNICATION AND INFORMATION THEORY

Pre-requisites: CSU 201 Discrete Computational Structures / MAG 501 Discrete Mathematics,
Knowledge of Probability Theory

L	T	P	C
3	0	0	3

Module I (10 Hours)

Entropy – Joint entropy and conditional entropy. Source Coding theorem – Shannon-Fano, Huffman Coding. Mathematical properties of entropy function. Chain rules for entropy, relative entropy and mutual information. Efficiency of Shannon-Fano coding. Optimality of Huffman coding.

Module II (12 Hours)

Channel Models – Symmetric channels. Binary Symmetric Channel – Information – Channel Coding theorem – Review of associated mathematical background . Channel relationships. Uniform Channel. Converse of Shannon's theorem.

Module III (10 Hours)

Zero error cordes. Error Correcting Codes . Ideal observer decoding. Minimum distance decoding. Maximum Likelihood decoding. Single Error Correction and Double Error Correction. Syndrome Decoding.

Module IV (10 Hours)

Linear Codes . Study of Repetition codes. Parity codes. Cyclic codes. Hamming code. Introduction to Golay code and Reed-Solomon codes. Establishing the bounds on a couple of these codes and the process of decoding them.

Reference:

1. R. W. Hamming, Coding and Information Theory, Prentice Hall, 1986.
2. T. Cover and J. Thomas, Elements of Information Theory, Wiley, 1991.
3. P. Garret, The mathematics of coding theory, Pierson Education, 2005.

CSU 301 DESIGN AND ANALYSIS OF ALGORITHMS

Pre-requisite: CSU 203 Data Structures & Algorithms

L	T	P	C
3	0	0	3

Module I **(10 hours)**
Analysis: RAM model - cost estimation based on key operations - big Oh - big omega - little Oh - little omega and theta notations - recurrence analysis - master's theorem - solution to recurrence relations with full history, probabilistic analysis - linearity of expectations - worst and average case analysis of quick-sort - merge-sort - heap-sort - binary search - hashing algorithms - lower bound proofs for the above problems - amortized analysis - aggregate - accounting and potential methods - analysis of Knuth-Morris-Pratt algorithm - amortized weight balanced trees

Module II **(10 hours)**
Design: divide and conquer - Strassen's algorithm, o(n) median finding algorithm - dynamic programming - matrix chain multiplication - optimal polygon triangulation - optimal binary search trees - Floyd-Warshall algorithm - CYK algorithm - greedy - Huffman coding - Knapsack, Kruskal's and Prim's algorithms for mst - backtracking - branch and bound - travelling salesman problem - matroids and theoretical foundations of greedy algorithms

Module III **(10 hours)**
Complexity: complexity classes - P, NP, Co-NP, NP-Hard and NP-complete problems - cook's theorem (proof not expected) - NP-completeness reductions for clique - vertex cover - subset sum - hamiltonian cycle - TSP - integer programming - approximation algorithms - vertex cover - TSP - set covering and subset sum

Module IV **(12 hours)**
Probabilistic algorithms: pseudo random number generation methods - Monte Carlo algorithms - probabilistic counting - verifying matrix multiplication - primality testing - miller rabin test - integer factorization - Pollard's rho heuristic - amplification of stochastic advantage - applications to cryptography - interactive proof systems - les vegas algorithms - randomized selection and sorting - randomized solution for eight queen problem - universal hashing - Dixon's integer factorization algorithm

Text Books:

1. Cormen T.H., Leiserson C.E, Rivest R.L. and Stein C, *Introduction to Algorithms*, Prentice Hall India, New Delhi, 2004, Modules I, II and III.
2. Motwani R. & Raghavan P., *Randomized Algorithms,* Cambridge University Press, Module IV

References:

1. Anany Levitin, *Introduction to the Design & Analysis of Algorithms*, Pearson Education. 2003
2. Basse S., *Computer Algorithms: Introduction to Design And Analysis*, Addison Wesley.
3. Manber U., *Introduction to Algorithms: A Creative Approach*, Addison Wesley
4. Aho A. V., Hopcroft J. E. & Ullman J. D., *The Design And Analysis of Computer Algorithms*, Addison Wesley

CSU 363 COMPUTATIONAL INTELLIGENCE

Pre-requisite: CSU 203 Data Structures & Algorithms

L	T	P	C
3	0	0	3

Module I (12 Hours)
Artificial Intelligence: History and Applications, Production Systems,Structures and Strategies for state space search- Data driven and goal drivensearch, Depth First and Breadth First Search, DFS with Iterative Deepening,Heuristic Search- Best First Search, A* Algorithm, AO* Algorithm, ConstraintSatisfaction, Using heuristics in games- Minimax Search, Alpha BetaProcedure.

Module II (11 Hours)
Knowledge representation - Propositional calculus, Predicate Calculus, Theorem proving by Resolution, Answer Extraction, AI Representational Schemes- Semantic Nets, Conceptual Dependency, Scripts, Frames, Introduction to Agent based problem solving.

Module III (11 Hours)
Machine Learning- Symbol based and Connectionist, Social and Emergentmodels of learning, The Genetic Algorithm- Genetic Programming, Overview of Expert System Technology- Rule based Expert Systems, Introduction to Natural Language Processing.

Module IV (8 Hours)
Languages and Programming Techniques for AI- Introduction to PROLOG and LISP, Search strategies and Logic Programming in LISP, Production System examples in PROLOG.

References

1.George.F.Luger, *Artificial Intelligence- Structures and Strategies for Complex Problem Solving,* 4/e, 2002, Pearson Education.
2. E. Rich, K.Knight, *Artificial Intelligence*, 2/e, Tata McGraw Hill
3. Winston. P. H, *LISP*, Addison Wesley
4. Ivan Bratko, *Prolog Programming for Artificial Intelligence*, 3/e, Addison Wesley, 2000

CSU 371 LOGIC FOR COMPUTER SCIENCE

Pre-requisite: CSU305 Theory of Computation

L	T	P	C
3	0	0	3

Module I   (11 hours)
Propositional logic, syntax of propositional logic, main connective, semantics of propositional logic, truth tables and tautologies, tableaus, soundness theorem ,finished sets, completeness theorem,.

Module II   (12hours)
Predicate logic, syntax of predicate logic, free and bound variables, semantics of predicate logic,, graphs, tableaus, soundness theorem, finished sets, completeness theorem, equivalence relations, order relations, set theory.

Module III   (14 hours)
Linear time Temporal Logic(LTL), syntax of LTL, semantics of LTL, Buchi Automata, Buchi recognizable languages and their properties, Automata theoretic methods, Vardi-Wolper Construction, Satisfiability problem of LTLl, Model checking problem of LTL.

Module IV   (6hours)
Software Veification: Tools used for software verification.SPIN and SMV. Introduction to both tools. Method of verification by the tools.

References:

1. Jerome Keisler H. Joel Robbin, *Mathematical Logic and Computability,* McGraw-Hill International Editions, 1996.
2. Papadimitriou, C. H., *Computational Complexity,* Addison Wesley, 1994
3. Gallier, J. H., *Logic for Computer Science: Foundations of Automatic Theorem Proving,,* Harper and Row, 1986.

CSU 334   WEB PROGRAMMING

Pre-requisite: CSU 304 Computer Networks

L	T	P	C
3	0	0	3

Module I   **(10 hours)**
Internet and WWW, Creating Web Graphics, HTML, Paintshop, Photoshop, FrontPage, Introduction to XHTML, Cascading Style Sheets.

Module II   **(12 hours)**
Introduction to Scripting, JavaScript: Control Statements, Functions, Arrays, Objects, Dynamic HTML: Object Model and Collections, Filters and Transitions, Data Binding with Tabular Data Control

Module III   **(10 hours)**
Building Interactive Animations, Extensible Markup Language (XML), Web Servers, Database: SQL, MySQL, DBI and ADO.NET,

Module IV   **(10 hours)**
Active server pages, CGI and Perl, PHP, Case Studies.

1**References:**

1. H. M. Deitel, P. J. Deitel and T. R. Nieto, *Internet and World Wide Web: How To Program*, Pearson Education, 2000.
2. Harvey Deitel, Paul Deitel, Tem Nieto, *Complete Internet & World Wide Web Programming Training Course, Student Edition*, 2/e, Prentice Hall , 2002

CSU 331 COMPUTER GRAPHICS AND MULTIMEDIA

Pre-requisite: CSU203 Data Structures & Algorithms

L	T	P	C
3	0	0	3

Module I

Introduction to computer graphics - basic raster graphics algorithms for drawing 2D primitives - scan converting lines - circles - generating characters - geometrical transformations - 2D transformations - homogeneous coordinates and matrix representation of transformations - window-to-viewport transformation - input devices and interactive techniques - interaction hardware - basic interaction tasks - 3D graphics - viewing in 3D - projections - basics of solid modelling - 3D transformations.

Module II

Introduction to multimedia - media and data streams - properties of a multimedia system - data stream characteristics - information units - multimedia hardware - platforms - memory and storage devices - input and output devices - communication devices - multimedia software - multimedia software tools - multimedia authoring tools

Module III

Multimedia building blocks - audio - basic sound concepts - music - speech - MIDI versus digital audio - audio file formats - sound for the web - images and graphics - basic concepts - computer image processing - video and animation - basic concepts - animation techniques - animation for the web - multimedia building blocks - audio - basic sound concepts - music - speech - MIDI versus digital audio - audio file formats - sound for the web - images and graphics - basic concepts - computer image processing - video and animation - basic concepts - animation techniques - animation for the web

Module IV

Data compression - storage space and coding requirements - classification of coding/compression techniques - basic compression techniques like JPEG, H.261, MPEG and DVI - multimedia database systems - characteristics of multimedia database management system - data analysis - data structure - operations on data - integration in a database model

References

1. Foley J. D., Van Dam A., Feiner S. K., & Hughes J. F., *Computer Graphics Principles and Practice*, Second Edition, Addison Wesley
2. Ralf Steinmetz & Klara Nahrstedt, *Multimedia: Computing, Communications and Applications*, Pearson Education
3. Newmann W & Sproull R.F., *Principles of Interactive Computer Graphics*, McGraw-Hill
4. Rogers D.F., *Procedural Elements for Computer Graphics*, McGraw-Hill
5. Hearn D. & Baker P.M, *Computer Graphics*, Prentice Hall India
6. Koegel Buford J. F., *Multimedia System*, Addison Wesley
7. Vaughan T., *Multimedia: Making it Work*, Third Edition, Tata McGraw Hill

CSU 341 DISTRIBUTED COMPUTING

Pre-requisite: CSU 313 Operating Systems

L	T	P	C
3	0	0	3

Module I **(10 hours)**
Distributed systems versus Parallel systems, Models of distributed systems**,** Happened Before and Potential Causality Model, Models based on States.

Module II **(10 hours)**
Logical clocks, Vector clocks, Verifying clock algorithms, Direct dependency clocks, Mutual exclusion, Lamport's algorithm, Ricart Agrawala algorithm.

Module III **(10 hours)**
Mutual exclusion algorithms using tokens and Quorums, Drinking philosophers problem, Dining philosophers problem under heavy and light load conditions. Leader election algorithms. Chang-Roberts algorithm.

Module IV **(12 hours)**
Global state detection, Global snapshot algorithm, Termination detection- Dijikstra and Scholten's algorithm, Causal message ordering algorithms, Self stabilization , Mutual exclusion with K-state machines.

References:

1. Vijay K. Garg., Elements of Distributed Computing, Wiley & Sons, 2002
2. Chow R. & Johnson T., *Distributed Operating Systems and Algorithms*, Addison Wesley, 2002
3. Tanenbaum S., *Distributed Operating Systems*, Pearson Education.,2005
4. Coulouris G., Dollimore J. & Kindberg T., *Distributed Systems Concepts And Design*, 2/e, Addison Wesley 2004

CSU 343 EMBEDDED SYSTEM DESIGN

Pre-requisites: CSU 313 Operating Systems
CSU 202 Logic Design
CSU 321 Software Engineering

L	T	P	C
3	0	0	3

Module I (10 hours)
Embedded system overview, trends in embedded software development, applications of embedded systems.

Module II (12 hours)
Hardware architecture, software engineering practices in the embedded software development process, embedded software development environments.

Module III (10 hours)
Embedded OS, development tools for target processors, real-time embedded software.

Module IV (10 hours)
Embedded communication, Mobile and database applications, Recent trends in Embedded Systems.

References:

1. R. Kamal, *Embedded Systems: Architecture, Programming & Design*, Tata McGraw Hill, 2003.
2. F. Vahid & T. Givargis *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley.
3. DreamTech Software Team, *Programming of Embedded Systems*, Wiley DreamTech, 2002.

CSU 411 COMPUTER SECURITY

Pre-requisites: CSU 304 Computer Networks, CSU 313 Operating Systems
CSU 213 Database Management Systems

L	T	P	C
3	0	0	3

Module I (10 hours)
Concepts of Security, Confidentiality, Integrity, Authenticity, Availability, Accuracy, Utility, Reliability and Possession. Concepts of Computationally Secure and Information theoretic security. Associated proofs. Zero Knowledge Protocols.

Module II (8 hours)
Access Control Matrix and Mechanisms, Vulnerability Analysis. Auditing Computer Security. Security Policy Guidelines. Security Awareness and Employment practices and policies. Anonymity and Identity in the cyber world. Practical examples from Network Domain. Tools for analysis and fingerprinting.

Module III (12 hours)
Systems Security – Operating Systems and Database Security.
Buffer overflow related vulnerabilities and attacks. Prevention.
SQL injection attacks and other web based attacks.
Security Enhanced Linux – A case study. Kerberos.

Module IV (12 hours)
Network Security. Firewalls, Vulnerability Assessment. Intrusion Detection Systems. DOS and DDOS attacks. Prevention strategies. Honey pot approach. Analysis.
Program Security. Security features of a programming language. Java as an example. Malicious code and Mobile code.

Reference:

1. Introduction to Computer Security. Matt Bishop. Addison-Wesley. 2004.
2. Security in Computing. Charles P Pfleeger. Pearson Education India. 2003.
3. Principles of Information Security. Michael E Whitman, Herbert J Mattord. Thomson. 2003.
4. Computer Security Handbook. Fourth Edition. Seymour Bosworth, M E Kabay, Editors. John Wiley. 2002.

CSU 353 MOBILE COMMUNICATION SYSTEMS

Pre-requisite: CSU 304 Computer Networks

L	T	P	C
3	0	0	3

Module I (8 hours)
Introduction, wireless transmission - frequencies for radio transmission - signals - antennas - signal propagation - multiplexing - modulation - spread sprectrum - cellular systems - medium access control - specialized MAC - SDMA - FDMA - TDMA - aloha - CSMA - collision avoidance - polling - CDMA - comparison of S/T/F/CDMA

Module II (10 hours)
Telecommunication sys
tems - mobile services - system architecture - radio interface - protocols - localization and calling - handover - security - new data services - satellite systems- broadcast systems - digital audio broadcasting - digital video broadcasting, WDM Optical networks.

Module III (12 hours)
Wireless LAN - infrared Vs radio transmissions - infrastructure and adhoc networks - IEEE 802.11 b/a/g - bluetooth - IEEE 802.16, Mobile network layer - mobile IP - packet delivery - registration - tunneling and encapsulation - optimizations - reverse tunneling - dynamic host configuration protocol

Module IV (12 hours)
Adhoc networks - routing - algorithms - metrics - mobile transport layer - TCP - indirect TCP - snooping TCP - mobile TCP - retransmission - recovery - transaction oriented TACP - support for mobility - file systems - WWW - WAP - architecture - datagram protocol - transport security - transaction protocol - session protocol - application - environment - WML - WML script - wireless telephony application.

References

1. Schiller J., *Mobile Communications*, 2/e, Pearson Education, 2003.
2. C. Siva Ram Murthy, *Ad Hoc Wireless Networks: Architectures and Protocols*, Pearson Education, 2004.
3. C. Siva Ram Murthy, *WDM Optical Networks: Concepts, Design, and Algorithms,* Pearson Education.
4. Singhal et.al S., *The Wireless Application Protocol*, Addison Wesley

CSU 354 ELECTRONIC COMMERCE

Pre-requisite: CSU 302 Number Theory & Cryptography

L	T	P	C
3	0	0	3

Module I (10 hours)
Web commerce concepts – the e-commerce phenomenon - electronic marketplace technologies - web based tools for e-commerce - e-commerce softwares - hosting services and packages

Module II (10 hours)
Security issues - approaches to safe e-commerce - PKI- biometrics for security in e-commerce – smart cards and applications

Module III (11 hours)
Wireless infrastructure – payment agents – mobile agent based systems – digital cash – security requirements for digital cash - Digital cheques, netcheque systems

Module IV (11 hours)
Secure electronic transaction- secure online payment – micropayments – industrial epayment systems – challenges and opportunities of e-payment.

References

1. Weidong Kou, *Payment Technologies for E-Commerce*, Springer, 2003.
2. Kalakota R. & Whinston A.B., *"Frontiers of Electronic Commerce*", Addison-Wesley, New Delhi
3. Janice Raynolds, The Complete E-Commerce Book, 2/e, CMP Books, 2004.
4. Schneider G. P. & Perry J. T., *Electronic Commerce, Course Technology*, Cambridge
5. Westland J. C. & Clark T.H. K., "*Global Electronic Commerce"*, University Press, 2001.
6. Minoli D. & Minoli E., "*Web Commerce Technology Handbook"*, Tata McGraw Hill, New Delhi

CSU 333 OBJECT ORIENTED ANALYSIS AND DESIGN

Pre-requisite: CSU 203 Data Structures and Algorithms

L	T	P	C
3	0	0	3

Module I **(10 hours)**
Introduction to Object-Oriented paradigm – The need, Examples – Basic notations and conventions. Object-oriented Modeling Concepts – Objects, Classes, Relationships, Encapsulation, Message sending, Inheritance, Polymorphism.

Module II **(10 hours)**
Unified Modeling Language – Types of models – Use-case diagrams – Class diagrams – Object diagrams – Sequence diagrams – Collaboration diagrams – state-chart diagrams, Activity diagrams – Component diagrams – Deployment diagrams

Module III **(10 hours)**
Introduction to Design Patterns – Creational Patterns, Structural Patterns, Behavioral Patterns, Case Study.

Module IV **(12 hours)**
Object Oriented Testing Methodologies – Implications of Inheritance on Testing, State based Testing, Adequacy and Coverage, Scenario based Testing, Testing Work Flow, Case Studies, Object Oriented Metrics.

References:

1. Erich Gamma, Richard Helm, Ralph Johnson, John M.Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional Computing Series, 1995.
2. James O.Coplien, Advanced C++ Programming Styles and Idioms, Addison Wesley, 1991.
3. Peter Coad and Edward Yourdon, Object-Oriented Analysis, Prentice Hall, 1990.
4. Margaret A. Ellis, Bjarne Stroustrup, Annotated C++: Reference Manual, Addison-Wesley Professional, 1990.
5. Booch G. Rumbaugh J & Jacobsons I, The Unified Modeling Language user guide, Addison Wesley. 1999.

6. Bahrami A, Object Oriented System Development, Mc Graw Hill, 1998.

CSU 431 ADVANCED DATABASE MANAGEMENT SYSTEMS

Pre-requisite: CSU 213 Database Management Systems

L	T	P	C
3	0	0	3

Module I **(10 hours)**

Overview of relational database concept - object oriented database - overview of CORBA standard for distributed objects.

Module II **(10 hours)**

Distributed database concepts - overview of client - server architecture and its relationship to distributed database, Deductive database - basic inference mechanism for logic programs.

Module III **(10 hours)**

Data warehousing and data mining - database on the World Wide Web - multimedia database - mobile database - geographic information system - digital libraries.

Module IV **(12 hours)**

Oracle and microsoft access - basic structure of the oracle system, database structures and its manipulation in oracle - programming oracle applications - oracle tools - an overview of microsoft access features and functionality of access - distributed databases in oracle.

References:

1. Elmasri, Navathe, Somayajulu, Gupta, *Fundamentals of Database Systems*, Pearson Education, 2006.
2. Ramakrishnan R. & Gehrke J *Database Management Systems*, 3rd Edition., McGraw Hill.
3. Connolly and Begg, Database systems, 3rd Edition, Pearson Education, 2003
4. O'neil P. & O'neil E *Database Principles, Programming and Performance*, 2nd Edition., Harcourt Asia (Morgan Kaufman).
5. Silberschatz, Korth H. F. & Sudarshan S, *Database System Concepts*, Tata McGraw Hill.

CSU 441 ADVANCED COMPUTER ARCHITECTURE

Pre-requisite: CSU 215 Computer Organization

L	T	P	C
3	0	0	3

Module I **(5 hours)**
Parallel Computation, Performance, Programming models, algorithms, evaluation

Module II **(13 hours)**
Shared Memory Multiprocessors, Memory Consistencey models, snoop based design, scalability, directory based cache coherence

Module III **(12 hours)**
Relaxed memory Consistency, Interconnection network design, Latency tolerance techniques, Multithreading architectures

Module IV **(12 hours)**
Advanced Topics: Selected Topics from Superscalar Design, Classical papers in Computer architecture, quantum architecture, Processor based Security

References:

1. Culler D and Singh J. P., *Parallel Computer Architecture: A Hardware Software Approach*, Harcourt Asia Pte Ltd, Singapore, 1999.
2. Hill M, Jouppi N and Sohi G, *Readings in Computer Architecture,* Morgan Kauffman, 2000.
3. Shen J. P. and Lipasti M.,, *Modern Processor Design*: Fundamentals of Superscalar Processors, McGraw Hill, First edition, 2000.

CSU 352 CODING THEORY

Pre-requisite: CSU 201 Discrete Computational Structures

L	T	P	C
3	0	0	3

Module I **(12 hours)**
Review of linear algebra - Linear codes and syndrome decoding. Generator and parity check matrices. Hamming geometry and code performance. Hamming codes. Error correction and concept of hamming distance.

Module II **(8 hours)**
Cyclic codes – Bose, Ray-Chaudhuri, Hocquenghem – BCH codes, RS codes – Polynomial time decoding. Shift register encoders for cyclic codes. Cyclic hamming codes. Decoding BCH – key equation and algorithms. Berlekamp's Iterative Algorithm for Finding the Error-Locator Polynomial.

Module III **(12 hours)**
Convolutional codes – Viterbi decoding. Concept of forward error correction. State diagram, trellises.
Concept of space time codes. Space Time Trellis codes. Path enumerators and proof of error bounds.
Applications to wireless communications.

Module IV **(10 hours)**
Graph theoretic codes – concept of girth and minimum distance in graph theoretic codes. Expander Graphs and Codes – linear time decoding. Basic expander based construction of list decodable codes. Sipser Spielman algorithm. Bounding results.

References:

1. R.J. McEliece, The Theory of Information and Coding, Addison Wesley, 1997.
2. R. Johannesson, K. Sh. Zigangirov, Fundamentals of Convolutional Coding, Universities Press, 2001.
3. Van Lint, J. H. An Introduction to Coding Theory, 2nd ed. New York: Springer-Verlag, 1992.

CSU 356 MOBILE COMPUTING

Pre-requisite: CSU 304 Computer Networks

L	T	P	C
3	0	0	3

Module I (10 hours)
Introduction to mobile computing, mobile development frameworks and tools, introduction to XML and UML.

Module II (10 hours)
Device independent and multichannel user interface development using UML, developing mobile GUIs, VUIs and mobile applications, multichannel and multimodal user interfaces.

Module III (11 hours)
Mobile agents and peer-to-peer architectures for mobile applications, wireless connectivity, synchronization and replication of mobile data, mobility and location based services, active transactions.

Module IV (11 hours)
Mobile Security, the mobile development process, architecture design and technology selection, mobile application development hurdles, testing mobile applications.

References:

1. Reza B'Far, Mobile Computing Principles, Cambridge University Press, 2005.
2. U. Hansmann, L. Merk, M. S. Nicklous and T. Stober, *Principles of Mobile Computing*, 2/e, Springer, 2003.
3. Harold Davis, *Anywhere Computing with Laptops: Making Mobile Easier*, O'Reilly, 2005
4. I. Stojmenovic, Handbook of wireless and Mobile computing, Wiley, 2002.

5. Schiller J., *Mobile Communications*, 2/e, Pearson Education, 2003.

CSU 361 IMAGE PROCESSING

Pre-requisite: CSU 201 Discrete Computational Structures / MEG 501 Discrete Mathematics

L	T	P	C
3	0	0	3

Module I

Introduction - digital image representation - fundamental steps in image processing - elements of digital image processing systems - *digital image fundamentals* - elements of visual perception - a simple image model - sampling and quantization - basic relationship between pixels - image geometry - image transforms - introduction to Fourier transform - discrete Fourier transform - some properties of 2-fourier transform (DFT) - the FFT - other separable image transforms - hotelling transform

Module II

Image enhancement - point processing - spatial filtering - frequency domain - color image processing - *image restoration* - degradation model - diagonalization of circulant and block circulant matrices - inverse filtering - least mean square filter

Module III

Image compression - image compression models - elements of information theory - error-free compression - lossy compression - image compression standards

Module IV

Image reconstruction from projections - basics of projection - parallel beam and fan beam projectio*n* - method of generating projections - Fourier slice theorem - filtered back projection algorithms - testing back projection algorithms

References

1. Rafael C., Gonzalez & Richard E. Woods, *Digital Image Processing*, Addison Wesley, New Delhi
2. Rosenfeld A. & Kak A.C., *Digital Picture Processing*, Academic Press
3. Jain A.K, Fundamentals of *Digital Image Processing*, Prentice Hall, Englewood Cliffs, N.J.
4. Schalkoff R. J., *Digital Image Processing and Computer Vision*, John Wiley and Sons, New York
5. Pratt W.K., *Digital Image Processing*, 2nd edition, John Wiley and Sons, New York

CSU 362 PATTERN RECOGNITION

Pre-requisite: CSU 203 Data Structures and Algorithms

L	T	P	C
3	0	0	3

Module I

Introduction - introduction to statistical - syntactic and descriptive approaches - features and feature extraction - learning - *Bayes Decision theory* - introduction - continuous case - 2-category classification - minimum error rate classification - classifiers - discriminant functions - and decision surfaces - error probabilities and integrals - normal density - discriminant functions for normal density

Module II

Parameter estimation and supervised learning - maximum likelihood estimation - the Bayes classifier - learning the mean of a normal density - general bayesian learning - *nonparametric technic* - density estimation - parzen windows - k-nearest neighbour estimation - estimation of posterior probabilities - nearest - neighbour rule - k-nearest neighbour rule

Module III

Linear discriminant functions - linear discriminant functions and decision surfaces - generalised linear discriminant functions - 2-category linearly separable case - non-separable behaviour - linear programming procedures - clustering - data description and clustering - similarity measures - criterion functions for clustering

Module IV

Syntactic approach to PR - introduction to pattern grammars and languages - higher dimensional grammars - tree, graph, web, plex, and shape grammars - stochastic grammars - attribute grammars - parsing techniques - grammatical inference

References

1. Duda & Hart P.E, *Pattern Classification And Scene Analysis*, John Wiley and Sons, NY
2. Gonzalez R.C. & Thomson M.G., *Syntactic Pattern Recognition - An Introduction*, Addison Wesley
3. Fu K.S., *Syntactic Pattern Recognition And Applications*, Prentice Hall, Englewood cliffs, N.J.

CSU 364 NATURAL LANGUAGE PROCESSING

Pre-requisite: CSU 203 Data Structures and Algorithms

L	T	P	C
3	0	0	3

Module I (8 hours)
Introduction to Natural Language Processing, Different Levels of language analysis, Representation and understanding, Linguistic background.

Module II (12 hours)
Grammars and parsing, Top down and Bottom up parsers, Transition Network Grammars, Feature systems and augmented grammars, Morphological analysis and the lexicon, Parsing with features, Augmented Transition Networks.

Module III (12 hours)
Grammars for natural language, Movement phenomenon in language, Handling questions in context free grammars, Hold mechanisms in ATNs, Gap threading, Human preferences in parsing, Shift reduce parsers, Deterministic parsers, Statistical methods for Ambiguity resolution

Module IV (10 hours)
Semantic Interpretation, word senses and ambiguity, Basic logical form language, Encoding ambiguity in logical from, Thematic roles, Linking syntax and semantics, Recent trends in NLP.

References:

1. James Allen, *Natural Language Understanding*, Second Edition, 2003, Pearson Education.
2. D Juraffsky, J H Martin, *Speech and Language Processing*, Pearson Education

CSU 373 COMPUTATIONAL COMPLEXITY

Pre-requisite: CSU 305 Theory of Computation

L	T	P	C
3	0	0	3

Module I **(10 hours)**
Review of Complexity Classes, NP and NP Completeness, Space Complexity, Hierarchies, Circuit satisfiability, Karp Lipton Theorem.

Module II **(10 hours)**
Randomized Computation, PTMs, Examples, Important BPP Results, Randomized Reductions, Counting Complexity, Permanent's and Valiant's Theorem

Module III **(10 hours)**
Review of Interactive Proofs, Lowerbounds: Randomized Decision Trees, Yao's minimax lemma, Communication Complexity, Multiparty Communication Complexity

Module IV **(12 hours)**
Advanced Topics: Selected topics from Average case Complexity, Levin's theory, Polynomial time samplability, random walks, expander graphs, derandomization, Error Correcting Codes, PCP and Hardness of Approximation, Quantum Computation

References:

1. Papadimtriou C. H.., *Computational Complexity*, Addison Wesley, First Edition, 1993.
2.` Motwani R*, Randomized Algorithms*, Cambridge University Press, 1995.
3. Vazirani V., *Approximation Algorithms*, Springer, First Edition, 2004.
4. Mitzenmacher M and Upfal E., *Probability and Computing, Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2005.

5. Arora S and Boaz B, *Computational Complexity*, (Web Draft) http://www.princeton.edu/theory/complexity

CSU 471 ADVANCED TOPICS IN ALGORITHMS

Pre-requisite: CSU 301 Design and Analysis of Algorithms

L	T	P	C
3	0	0	3

Module I (10 hours)

Discrete Probability: Probability, Expectations, Tail Bounds, Chernoff Bound, Markov Chains. Random Walks. Review of Generating functions, Exponential Generating Functions. Review of Recurrence Relations – both homogeneous and non-homogeneous of first and second degrees. Review of Analysis of recursive and non recursive algorithms.

Module II (12 hours)

Randomized Algorithms, Moments and Deviations. Tail Inequalities. Randomized selection.
Las Vegas Algorithms. Monte Carlo Algorithms. Parallel and Distributed Algorithms. Concept of De-Randomization and techniques.

Module III (10 hours)

Complexity: Probabilistic Complexity Classes, Proof Theory. Interactive Proof Systems.
Examples of probabilistic algorithms. Proving that an algorithm is correct 'Almost sure'.
Complexity analysis of probabilistic algorithms . The complexity classes PP and BPP

Module IV (10 hours)

Kolmogorv Complexity – basic concepts. Models of Computation. Applications to analysis of algorithms. Lower bounds. Relation to Entropy. Kolmogorov complexity and universal probability.
Godel's Incompleteness Theorem. Different Interpretations. Chatin's Proof for Godel's Theorem.

References:

1. R. Motwani and P. Raghavan, Randomized Algorithms, Cambrdige University Press, 1995
2. C. H. Papadimitriou, Computational Complexity, Addison Wesley, 1994
3. Dexter C. Kozen, The Design and Analysis of Algorithms, Springer verlag N.Y, 1992

CSU 472 QUANTUM COMPUTATION

Pre-requisites: CSU 203 Data Structures and Algorithms, CSU 301 Design and Analysis of Algorithms

L	T	P	C
3	0	0	3

Module I (12 hours)
Review of Linear Algebra. The postulates of quantum mechanics. Review of Theory of Finite Dimensional Hilbert Spaces and Tensor Products.

Module II (8 hours)
Models of computation – Turing machines. Quantifying resources. Computational complexity and the various complexity classes. Models for Quantum Computation. Qubits. Single and multiple qubit gates. Quantum circuits. Bell states. Single qubit operations. Controlled operations and measurement. Universal quantum gates.

Module III (12 hours)
Quantum Algorithms – Quantum search algorithm - geometric visualization and performance. Quantum search as a quantum simulation. Speeding up the solution of NP Complete problems. Quantum search as an unstructured database. Grover's and Shor's Algorithms.

Module IV (10 hours)
Introduction to Quantum Coding Theory. Quantum error correction. The Shor code. Discretization of errors, Independent error models, Degenerate Codes. The quantum Hamming bound. Constructing quantum codes – Classical linear codes, Shannon entropy and Von Neuman Entropy.

References:

1. Nielsen M.A. and I.L. Chauang, Quantum Computation and Quantum Information, Cambridge University Press, 2002.
2. Gruska, J. Quantum Computing, McGraw Hill, 1999.
3. Halmos, P. R. Finite Dimensional Vector Spaces, Van Nostrand, 1958.

CSU 305 THEORY OF COMPUTATION

Pre-requisite: CSU 211 Formal Languages and Automata

L	T	P	C
3	0	0	3

Module I (8 hours)
Undecidability: Recursive and Recursively enumerable sets, Undecidability, Rice theorems.

Module II (16 hours)
Complexity: P, NP, PSPACE and Log space. Reductions and Completeness. Hierarchy theorems, Probabilistic classes, BPP, EXP time and space complexity classes.

Module III (8 hours)
Logic: Propositional logic, compactness, decidability, Resolution.

Module IV (10 hours)
Undecidability in first order predicate calculus, Resolution. Gödel’s incompleteness theorem

Text Books:

1. M. Sipser, *Introduction to the Theory of Computation*, Thomson, 2001.
2. C. H. Papadimitriou., *Computational Complexity*, Addison Wesley, 1994.

References:

1. C. H. Papadimitriou, H. Lewis., *Elements of Theory of Computation*, Prentice Hall, 1981.
2. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Narosa, 1989.

3. J. C. Martin, *Introduction to Languages and the Theory of Computation*, Mc Graw Hill, 2002.
4. M. R. Garey and D. S. Johnson. *Computers & Intractability*, W. H. Freeman & Co., San Farnisco, 1979.

CSU 315 COMPUTER HARDWARE

Prerequisite: CSU 202 Logic Design

L	T	P	C
3	0	0	3

Module I (8 Hours)

PC hardware: motherboard, memory SDRAM, RDRAM Adapters – graphic adapter, network adapter. Controllers, floppy and hard disk controllers, streamers and other drives, Interfaces - parallel and serial interfaces, keyboard, mice and other rodents, the power supply, operating system, BIOS, and memory organization. *8086/8088 Hardware specification*: clock generator, bus buffering and latching, bus timing, ready and wait states, minimum and maximum mode operations. Features of Pentium IV processor

-
Module II (12 Hours)

Microprocessor architecture: real mode and protected mode memory addressing, memory paging. *Addressing modes*: data addressing, program memory addressing, stack memory addressing. *Data movement instructions, Arithmetic and logic instructions, Program control instructions, Programming the microprocessor*: modular programming, using keyboard and display, data conversions, disk files, interrupt hooks, using assembly language with C/C++.

-
Module III (13 Hours)

Memory interface: memory devices, address decoding, 16 bit (8086), 32 bit (80486) and 64 bit (Pentium) ,Hardware architecture for embedded systems-processor-memory-latches and buffers-display unit-16 and 32 bit processors. Memory interfaces, dynamic RAM. I/O interface: port address decoding, PPI, 8279 interface, 8254 timer interface, 16550 UART interface, ADC/DAC interfaces.

-
Module IV (9 Hours)

Interrupts: interrupt processing, hardware interrupts, expanding the interrupt, 8259A programmable interrupt controller. DMA: DMA operation, 8237 DMA controller, shared bus operation, disk memory systems, video displays.
Bus interface: ISA bus, EISA and VESA buses, PCI bus.

References:

1. B. B. Brey*, The Intel Microprocessors 8086 to Pentium: Architecture, Programming and Interface*, 6/e, Prentice Hall of India, New Delhi, 2003.
2. Programming for embedded systems Dream Software team , Willey 2002
3. H. P. Messmer, *The Indispensable PC Hardware Book*, 3/e, Addison Wesley, 1997.
4. A. K. Ray, and K. M. Bhurchandi, Advanced Microprocessors and Peripherals, Tata McGraw Hill, 2000.
5. D. V. Hall, *Microprocessors and Interfacing: Programming and Hardware*, 2/e, Tata McGraw Hill, New Delhi, 1992.
6. K. Miller, *An Assembly Language Introduction to Computer Architecture using the Intel Pentium*, Oxford University Press, 1999.
7. S. J. Bigelow, *Troubleshooting, Maintaining, and Repairing PCs*, 2/e, Tata McGraw Hill, New Delhi, 1999.