DETAILED COURSE CONTENTS FOR B. TECH. PROGRAMME IN COMPUTER SCIENCE & ENGINEERING

(2012 Batch onwards)

1ST / 2ND SEMESTER

CSX-101 Computer Programming **[2 0 0 2]**

Introduction: Computer system, Software and hardware concepts. Memories, Booting of Computer system, Disk operating systems, Internal and external commands of DOS, Concepts of the finite storage, bits bytes, kilo, mega and gigabytes, Flow charts.

Major elements of C program, Declarations of Variables, Body of Statements and functions, Identifiers, Keywords, Data types, Variables, Expressions, Statements, Type Conversions, Operators, Arithmetic operators, Unary operators, Binary operators, Char type, Cast operator, Precedence and Associatively of operations, Change of Precedence, Assignment operators, Bitwise operators, Shift operators, Relational operators, Ternary operators, Logical operators, Loops, if statement, if else statement, For loop and it's variants, While loop, do-while loop, , Switch function, Break function, Continue function, Null statement, Goto Function, Escape Sequences.

Function and Macros, What is function, Passing arguments to function, passing arguments by value, returning value, passing arguments by reference, scope rule of function, recursion, storage classes and scope, library functions like math library, recursion, C preprocessor like macro expansion, macro with arguments, macro vs. functions, file inclusion, conditional compilation.

Array and Pointers: Array, array initialization: static and dynamic, bound checking, passing, array elements to functions, Pointers. Passing an entire array to functions, 2D array, initialization of 2D array, Array of pointers, Operations on pointers, Multi-dimensional array.

Strings: Why use strings, declaring strings, pointers and strings, string library functions, 2D array of characters, Array of pointers to strings.

Structures: Types of structure declarations, User defined data types, Nested structures declaration, Initialization of structure, processing structures with initialized values, using array as structures member, array of structure variable, Pointers to structures, Unions.

File Handling: Data organization. File operations, File opening modes, String I/O in file, Record I/O in files, Text files and Binary files.

Text/References:

1. Yashwant Kanetkar, "Let us C", BPB Publications, 8/e, 2008.
2. Herbert Schildt, "C:The complete reference", Mcgraw Hill, 4/e, 2002.
3. Raja Raman, "Computer Programming in C", Prentice Hall of India, 1995.
4. Kernighan and Ritchie, "C Programming Language", PHI, 2/e, 1998.
5. Jeri R. Hanly and Elliot B. Koffman, "Problem Solving and Program Design in C", Pearson Addison-Wesley, 7/e, 2012.

CSX-102 COMPUTER PROGRAMMING LABORATORY [0 0 2 1]

1. To work at command prompt and running internal and external commands on it.
2. Installation and Introduction of C Compiler.
3. Program to demonstrate the use of various operators.
4. Programs to demonstrate the use of conditional statements and loop statements: if, if-else, nested-if else, for, while and do-while.
5. Program to print various patterns using nested for loops.
6. Program to make use of jump statements in loop: break, continue and goto.
7. Program to demonstrate the concept of call by value and call by reference methods of parameter passing to function.
8. Program to find prime factor of number using function.
9. Program to generate Fibonacci series using recursive function.
10. Program to find root of quadratic equation. Use sqrt( ) function in math.h header file to find the suare root of number.
11. Write macro definitions with arguments for calculation of area and parameter of a triangle, a square and a circle. Store these macro definitions in a file called areaperei.h. Include this file in your program and call the macro definitions for calculating area and perimeter for different squares, triangles and circles.
12. Write down macro definitions for the following.
    a. To find arithmetic mean of two numbers
    b. To convert uppercase alphabet to lowercase.
    c. To obtain bigger of two numbers.
13. Program to find largest and smallest number in 1D array.
14. Program to find the sum of even and odd numbers in 1D and 2D array.
15. Program to Program to multiply two matrices.
16. Program to demonstrate the concept of passing array to function.
17. Program to compare, concatenate two strings without using inbuilt functions.
18. Program to check whether a string is palindrome or not.
19. Program to swap to numbers using pointers.
20. Program to reverse the strings stored in the array of pointers to strings.
21. Program to demonstrate the operations on pointers.
22. Program to multiply two arrays using pointers to the functions.
23. Program to display the details of an employee record using nested structure.
24. Program to read a file and display its contents.
25. Suppose a file contains student's records with each record containing name and age of a student. Write a program to read these records and display them in sorted order by name.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

3rd SEMESTER

CSX-201 DIGITAL CIRCUITS AND LOGIC DESIGN [3 0 0 3]

Binary Systems: Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic.

Boolean Algebra And Logic Gates: Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean algebra, Boolean functions canonical and standard forms, other logic operations, Digital logic gages, integrated circuits.

Gate – Level Minimization: The map method, Four-variable map, Five-Variable map, product of sums simplification Don't-care conditions, NAND and NOR implementation other Two-level implementnations, Exclusive – Or function, Hardward Description language (HDL).

Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

Synchronous Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, HDL for sequential circuits, State Reduction and Assignment, Design Procedure**.**

Registers and Counters: Registers, shift Registers, Ripple counters synchronous counters, other counters, HDL for Registers and counters.

Memory, CPLDs, and FPGAs: Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.

Asynchronous Sequential Logic: Introduction, Analysis Procedure, Circuits with Latches, Design Procedure, Reduciton of state and Flow Tables, Race-Free state Assignment Hazards, Design Example.

Text/References:

1. M. Morris Mano, "Digital Design", Pearson Education, 5/e, 2012.
2. Donald D. Givone, "Digital Principles and Design", Tata McGraw Hill, 2002.
3. John F. Wakerly, "Digital Design Principles and Practices", Pearson Education, 4/e, 2005.
4. J P. Hayes, "Introduction to Digital Logic Design", Addison-Wesley Publishing, 1993.
5. H. Roth Charles and L. Kinney, "Fundamentals of logic design", Cengage Learning, 7/e, 2013.

CSX-203 INFORMATION CODING THEORY [3 0 0 3]

Information: channel capacity, the concept of amount of information, entropy, Information rate, Conditional and joint entropies.

Source coding: Noise less coding, Shannon's first fundamental theorem, Discrete memory less channel, Mutual information, Sources with finite memory, Markov sources, Shannon's second fundamental theorem on coding, Huffman coding, Lempel – Ziv algorithm, Shannon-Fano algorithm.

Channel coding: Error detecting codes, Hamming distance, Error correcting codes, Repitition codes, Linear block codes, binary cyclic codes, BCH codes, Reed-Soleman codes, Golay codes.

Convolution Coding: Code tree, state diagram, Trellis diagram, Maximum-Likelihood decoding – Viterbi's algorithm, sequential decoding.

Network Information theory, Introduction to Cryptography

Text/References:

1. Thomas M. Cover and Joy A. Thomas, "Elements of Information Theory", Wiley-Interscience, 1991.
2. S. Haykins, "Digital Communications", Wiley, 1988.
3. J. G. Proakis, "Digital Communications", Mc Graw Hill, 5/e, 2007.
4. D. H. Ballard and C. M. Brown, "Computer Vision", Prentice Hall, 1982.
5. S. Roman, "Introduction to Coding and Information Theory", Springer-Verlag, 1997.

CSX-205 COMPUTER ARCHITECTURE & ORGANIZATION [3 1 0 4]

Introduction: Historical overview, economic trends, underlying technologies, Data Representation- Data Types, Complements. Fixed-Point Representation, Floating-Point Representation. Error Detection and Correction. Addition, Subtraction, Multiplication and Division algorithms and hardware.

Register Transfer and Micro operations: Register transfer language, Inter-Register Transfer, Arithmetic Micro-operations, Logic and Shift micro-operations Language, Control functions.

Arithmetic Logic Unit: Arithmetic, logic and shift micro operations. Constructing an arithmetic logic shift unit.

Basic Computer Architecture and Design: Computer registers, Computer Instructions-Instruction Set Completeness. Classifying Instruction Set Architecture. Basic steps of Instruction Execution. Hardwired Control. Micro programmed Control. Horizontal and Vertical Microprogramming. Interrupts.

Central Processing Unit: General Register Organization. Stack Organized CPU. Instruction Formats, Addressing Modes. Data Transfer and Manipulation.RISC Vs CISC.

Pipelining: Parallel and pipeline Processing, Pipeline Control, Pipeline Implementations, Conflicts Resolution, and Pipeline Hazards. Vector Processing, and Array Processors.

Memory Organization: Memory Systems: principle of locality, principles of memory hierarchy Caches, associative memory, main memory, Virtual memory, Paging and Segmentation, Memory Interleaving.

Input Output Organization: I/O performance measures, types and characteristics of I/O devices, I/O Modes-Programmed I/O, Interrupt Initiated I/O and DMA.Buses: connecting I/O devices to processor and memory, interfacing I/O devices to memory, processor, and operating system.

Parallel Computers: Classification, SIMD, MIMD Organizations, Connection Networks, Data Flow Machines, and Multithreaded Architectures.

Text/References:

1. M. Moris Mano, "Computer System Architecture", Pearson Education, 3/e, 1992.
2. David A. Patterson and John L. Hennessy, "Computer Organization & Design-The Hardware/Software Interface", Morgan Kaufmann, 5/e, 2013.
3. William Stallings, "Computer Organisation and Architecture, Designing for Performance", Pearson Education Asia, 9/e, 2012.
4. Harry F. Jordan and Gita Alaghband, "Fundamentals of Parallel Processing", Pearson Education, 2002.
5. Barry Wilkinson and Michael Allen, "Parallel Programming", Prentice Hall, 2/e, 2004.

CSX-207 OBJECT ORIENTED PROGRAMMING [3 0 0 3]

Object oriented thinking: Need for OOP Paradigm, Procedural programming v/s object oriented programming, object oriented concepts.

Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, function overloading, Math library functions.

Class and Object: Difference between C structure and class, specifying a class, Defining member functions: inside and outside class, scope resolution operator, Array within a class, array of objects, Static data members and member functions,Object as function arguments,returning objects,Friend function, memory allocation for objects,pointer to members,pointer to object,this pointer local classes.

Constructor and destructor:Constructor,types of constructors:default,parameterized and copy constructor,constructor overloading, constructor with default parameter,dynamic initialisation of objects,destructor

Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator,Data Conversion: Basic to User Defined , User defined to basic, Conversion from one user-defined to other.

Inheritance and polymorphism: Base class, derived class, visibility modes, derivation and friendship,Types of inheritance,Containership,virtual function binding,pure virtual functions,Abstract class,pointer to derived class.

Console IO operations: C++ stream classes, Unformatted IO operations, formatted IO operations, managing output with manipulators.

Exception handing: Exceptions and derived classes, function exception declarations, Unexpected exceptions, Exceptions when handling exceptions, resource capture and release etc.

Working with files: Classes for file stream operations, opening and closing files, detectinf cof, File opening modes, file Pointers, Error handling during file operations, command line arguments. Templates: Class template, class template with parameter, functiontemplate, function template with parameter.

Text/References:

1. Bjrane Stroustrup, "C++ Programming language", Pearson education Asia, 4/e, 2013.
2. R. Lafore, "Object oriented Programming in C++", Techmedia New Delhi, 4/e, 2002.
3. Yashwant Kanetkar, "Let us C", BPB Publications, 8/e, 2008.
4. B. A. Forouzan and R. F. Gilberg, "A structured approach using C++", Cengage Learning, 1999.

CSX-209 DATA STRUCTURES AND ALGORITHMS [3 1 0 4]

Introduction: Basic Terminology, Elementary Data Organization, Structure operations, Algorithm Complexity and Time-Space trade-off

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C++, Character string operation, Array as Parameters, Ordered List, Sparse Matrices and Vectors.

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Recursion: Recursive definition and processes, recursion in C, example of recursion, Tower of Hanoi Problem, simulating recursion, Backtracking, recursive algorithms, principles of recursion, tail recursion, removal of recursion.

Queues: Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues**.**

Linked list: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion

Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list, Garbage Collection and Compaction.

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm**.**
Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-trees.

Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting**.**

Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

File Structures: Physical Storage Media File Organization, Organization of records into Blocks, Sequential Files, Indexing and Hashing, Primary indices, Secondary indices, B+ Tree index Files, B Tree index Files, Indexing and Hashing Comparisons.

Text/References:

1. E. Horowitz, S. Sahni and Dinesh Mehta, "Fundamentals of Data structures in C++", Silicon Pr, 2/e, 2006.
2. R. Kruse and A. Ryba, "Data Structures and Program Design in C", Pearson Education Asia, Delhi, 2002.
3. Yedidyah Langsam, Moshe J. Augenstein and Aaron M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India, 2/e, 1995.
4. Bruno R. Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", John Wiley & Sons, 1998.
5. Richard F. Gilberg and Behrouz A. Forouzan, "Data Structure – A pseudo code approach with C++", Cengage Learning, 2001.

CSX-221 DIGITAL AND ANALOG CIRCUITS LABORATORY [0 0 2 1]

1. Verification of Boolean Theorems - Implementation of Boolean Function - Adders / Subtractors - Decoders Encoders - Multiplexer - Demultiplexers - Comparators - Parity Checker/Generator.
2. Register Counters - Shift Registers - General-purpose shift registers - Data transmission.
3. Project - A mini project involving clocked sequential networks design.
4. To see the working of a BCD-to-7 Segment decoder and to verify the truth table.
5. To study the operation of Arithmetic and logic unit using 74181 IC.
6. To study the ckt. Of 16 line-to-1 line Multiplexer using 74150 and 7493IC, s with DEMO board.
7. To construct different types of flip-flops and verify their truth tables. Flip-flops like J-K flip-flops. S-R flip-flop. And D-flip-flops etc.
8. To construct and verify a Master-Slave flip-flop.
9. Construction and study of Modulo-N counter using IC's 7490 decade counter, 7493 binary counter.
10. Study of various Interfacing card
    Stepper motor with voluntary unit.
    12-bit high-speed data acquisition card.
    PC Bus Extension unit.
    16-bit channel Relay output card.
    Digital IC tester.
    7-Segment display card.
    Amplifier and multiplexer card.
    IEEE 488 GPIB card.
    Digital I/O and timer counter card.

11. EPROM Programming
12. Study & working of DMA controller.
13. Designing of Traffic control system.

CSX-227 OBJECT ORIENTED PROGRAMMING LABORATORY [0 0 2 1]

Objectives:

- To make the student learn a object oriented way of solving problems.
- To teach the student to write programs in C++ to solve the problems

1. Write a program to read a matrix of size m x n form the keyboard and display the same using function.
2. Program to make the use of inline function.
3. Write a function power () which raise a number m to a power n. The function takes double value of m and integer value of n and returns the result. Use a default value of n is 2 to make the function to calculate squares when this argument is omitted.
4. Program to show that the effect of default arguments can be alternatively achieved by overloading.
5. Write a class ACCOUNT that represents your bank account and then use it.
   The class should allow you to deposit money, withdraw money, calculate interest,
   send you a message if you have insufficient balance.
5. Write a class STRING that can be used to store strings, add strings, equate string,
   output strings.
7. Create the class TIME to store time in hours and minutes. Write a friend function to add two TIME objects.
8. Create two classes DM and DB. DM stores the distance in meter and centimeters and DB stores the distance in feet and inches. Write a program two add object of DM with the object of DB class.
9. Write a program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes inherits the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
10. Program to demonstrate the concept of:
    a. Default constructor
    b. Parameterized constructor
    c. Copy constructor
    d. Constructor overloading
11. Program to demonstrate the concept of destructor.
12. Program to show multiple inheritance
13. Program to show multilevel inheritance
14. Program to show hybrid inheritance
15. Program to show the concept of containership.
16. Program to overload unary operator.
17. Program to overload binary operator
18. Program to show the concept of run time polymorphism using virtual function.
19. Program to work with formatted and unformatted IO operations.
20. Program to read the name and roll numbers of students from keyboard and write them into a file and then display it.
21. Program to copy one file onto the end of another, adding line numbers
22. Write a function template for finding the minimum value contained in an array.
23. Write a class template to represent generic vector (a series of float values). Include member function to perform following tasks.
    a. Create vector
    b. Modify the value of a given element
    c. To multiply by a scalar value
    d. To display vector in the form of (10, 20, 30,……)

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

CSX-229 DATA STRUCTURES AND ALGORITHM LABORATORY [0 0 2 1]

Write Program in C or C++ for following.

1. Write a C+ program to the following operations on stack of integers:
   a> push b>pop c> display
   The program should print appropriate messages for stack overflow, stack Underflow & stack empty.
2. Write a C++ program to convert & print a given valid parenthesized in fix Arithmetic expression to postfix expression. The expression consists of single character operands & + , - ,*,/ operators .
3. Write a c program to evaluate a valid suffix / postfix expression using a Stack, assume that the suffix / postfix expression is read as a single line consisting of non negative single digit operands & binary arithmetic operands. The arithmetic operators are + (ADD), - (subtract), *(multiply) & / (divide).
4. Write a C++ program to simulate the working a queue of integers using an array. Provide the a) insert b) delete c) display
5. Write a C++ program to simulate the working of a circular queue of integers using an array. Provide the following operations: a) insert b) delete c)Display
6. Write a program to design a priority queue which is maintained as a set of queue (assume a maximum of 3 queues). The elements are inserted based upon the given priority. The deletion of an element is to be done starting from the 1st queue, if it is not empty .If it is empty ,The elements from the 2nd queue will be deleted & so on.
7. Write a C++ program using dynamic variable & pointers to construct a singly linked list consisting of the following information in each node. Student id (integer), student name(character string) & semester(integer). The operations to be supported are
   a) inserting in front of list
   b) Deleting a node based on student id, if the specified node is not present in the list, error message should be displayed
   c) Searching a node based on student id , if the specified node is not present in the list, error message should be displayed
   d) Displaying all the nodes in the list
8. Write a C++ program using dynamic variables & pointers to construct an ordered(ascending) singly linked list based on the rank of the student, where each node consists of the following information student id(integer)student name(character), rank(integer)
9. Write a C++ program using dynamic variables & pointers to construct a singly linked list to perform the operations of a stack of integers
   a) Push b) pop c) display
10. The program should print appropriate message for stack overflow & stack empty
11. Write a C++ program to support the following operations on a doubly linked where each node each node consists of integers
    a) Create a doubly linked list by adding each node front
    b) Insert a new node to the left of the node whose key value is read as a input
    c) Delete the node of a given data, if it is found, otherwise display appropriate message
    d) Display the content of the list
12. Write C++ program
    a) To construct a binary search tree of integers
    b) To traverse the tree using all the methods i.e. inorder, preorder & postorder to display the elements in the tree
13. Write C++ program for the following searching techniques over a list of integers.
    a> Linear search,
    b> Binary search
14. Write a C++ program to sort a list of N integers using the quick sort algorithm.
15. Write a C++ program to sort a list of N strings using the insertion sort algorithm.
16. Write a C++ program to sort a list of N integers using Heap sort algorithm.

4TH SEMESTER:

CSX-202 DISCRETE STRUCTURES [3 1 0 4]

Set Theory: Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets Set Theory, Functions and Relations: Subsets, Power Set, Null Set, Singleton, Finite Set, Infinite Set, Universal Set, Disjoint Sets, Operation on Sets, Venn Diagrams, Cartesian Product of Sets, Partition of Sets, Concept of Relation & Properties of Relations, Different types of Relations, Tabular and Matrix Representation of Relations, Relations and Diagraphs, Composition of Relations, Functions and their different mappings, Composition of Function, Recursion and Recurrence Relations.

Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded I and complemented lattices.

Boolean Algebra: Partial Ordering, Totally ordered Sets, Dual Order, Hasse Diagram Lexicographic Ordering, Cover of an Element, Least and Greatest Elements, Minimal and Maximal Elements ,Upper and Lower Bound , Well-Order Set, Binary and n-Ary Operations, Lattices, Atoms of a Boolean Algebra, Boolean Expressions, Applications of Boolean Algebra to Switching Theory.

Tree: Definition, Rooted tree, properties of trees, binary search tree, tree traversal.

Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Universal and existential quantifiers.

Combinatorics & Graphs: Recurrence Relation, Generating function., Simple graph, multi graph, graph terminology, representation of graphs, Bipartite, Regular, Planar and connected graphs, connected components in a graph, Euler graphs, Hamiltonian path and circuits, Graph coloring, chromatic number, isomorphism and Homomorphism of graphs.

Text/References:

1. S. Liptschutz and M. Lipson, "Discrete Mathematics", McGraw Hill, 3/e, 2007.
2. J. P. Trembley and R. Manohar, "Discrete Mathematical Structure with Application to Computer Science", McGraw Hill, 1987.
3. Kenneth H. Rosen, "Discrete Mathematics and its applications", McGraw Hill, 4/e, 1998.
4. Narsingh Deo, "Graph Theory With application to Engineering and Computer.Science", PHI, 2004.
5. V. Krishnamurthy, "Combinatorics Theory & Application", East-West Press Pvt. Ltd., 1986.

CSX-204 DATABASE MANAGEMENT SYSTEM [3 0 0 3]

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model:
ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

Data Base Design & Normalization:
Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

Crash Recovery: Failure classification, recovery concepts based on deferred update, recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Client/Server Databases: Client/Server concepts, approach, Client/Server environments, characterization of Client/Server computing, application partitioning, the two-layer, and the Three layer architecture, Client/Server communication, APIs in Client/Server computing, middleware technology, application developments, design concepts, Client application development tools, and database servers.

Integrity, Security and Repositories: Needs for database integrity, integrity constraints, non-procedural integrity constraints, integrity constraints specifications in SQL, introduction to database security mechanism, security specification in SQL, system catalogues

Case Studies:
Oracle: Database Design and Querying Tools; SQL Variations and Extensions; Storage and Indexing; Query Processing and Optimization; Concurrency Control and Recovery; System Architecture; Replication, Distribution and External Data; Database Administration Tools.
IBM DB2: Universal database; Database Design and Querying Tools; SQL Variations and Extensions Storage and Indexing; Query Processing and Optimization; Concurrency Control and Recovery; System Architecture; Replication, Distribution and External Data; Database Administration Tools.

Text/References:

1. C. J. Date, "An Introduction To Database System", Addision Wesley, 8/e, 2003.
2. Korth, Silbertz and Sudarshan, "Database Concepts", McGraw Hill, 5/e, 2005.
3. Elmasri and Navathe, "Fundamentals Of Database Systems", Addision Wesley, 6/e, 2010.
4. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication, 1990.
5. Rob and Coronel, "Database Systems", Cengage Learning, 7/e, 2006.

ECX-206 MICRO-PROCESSOR AND PROGRAMMING [3 0 0 3]

Introduction to Microcomputers & Microprocessor: Digital computing, Computer languages, From large chip computers to single chip Microcomputers, Microcomputers organization, and 4- bit Microprocessors.

Introduction to 8085 Microprocessor : Microprocessor architecture & its operations, Memory mapped and peripheral I/O, 8085 based Microcomputer, Instruction classification, Instruction format, Instruction timings, 8080 A MPU, and Overview of 8085/8080A instruction set, Introduction to 16 bit microprocessors.

Assembly Language: Simple sequence program jump,flags,Conditional jump ,LoopsDelay, Programming techniques &indexing ,debugging.

Interrupts: 8085 interrupts,Additional I/O concepts & processes.

Interfacing and Programmable Devices : Basic interfacing concepts,Progrmmable parallel ports,Handshake I/O,Interfacing keyboards and displays ,Introduction to PPI ,Programmable interval timer, Programmable interrupt controller,SID &SOD.

Introduction to 8051 Microcontroller: 8051 Microcontroller: Comparison of Microprocessor and Micro Controller, Micro Controller and Embedded Processors, Overview of 8085 families ,Introduction to 8051 Assembly Programming, 8051 flag bits and PSW Register.Register bank and stack.

Text/References:

1. Ramesh S Gaonkar, "Microprocessor Architecture- Programming & Applications with 8085/8080A", 5th Ed., Penram International Publishing (India) Pvt. Ltd
2. James L Antonakes , "An Introduction to Intel family Of Microprocessors" 3rd Ed.,Pearson Education.
3. Ali Mazidi, "The 8051 Microcontroller and Embedded System ", Pearson Education,2005.
4. Kenneth J Ayala, "The 8051 Microcontroller" 3rd edition Cenage Learning Hall Douglas V,"Microprocessor and Interfacing",Revised 2nd edition ,Tata McGraw Hill,2005

SX-208 DATA COMMUNICATION [3 0 0 3]

Data Transmission/The Physical Layer: Concepts and Terminology, Analog and Digital Data Transmission, Transmission Impairments, Guided Transmission Media, Wireless Transmission, Communication Satellites, The Public Switched Telephone Network, The Mobile Telephone System, Cable Television

Data Encoding: Digital Data: Digital and Analog Signals, Analog Data: Digital and Analog Signals, Spread Spectrum

Data Communication Interface: Asynchronous and Synchronous Transmission, Line Configurations, Interfacing

Multiplexing: Frequency-Division Multiplexing, Synchronous Time-Division Multiplexing, Statistical Time-Division Multiplexing

Circuit Packet and Switching: Switched Networks, Circuit-Switching Networks, Switching Concepts, Routing in Circuit-Switched Networks, Control Signaling, Packet-Switching Principles, Routing, Congestion Control, X.25 282

Frame Relay: Frame Relay Protocol Architecture, Frame Relay Call Control, User Data Transfer, Network Function, Congestion Control

LAN Technology and Systems: LAN Architecture, BusITree LANs, Ring LANs, Star LANs, Wireless LANs, Ethernet and Fast Ethernet (CSMAICD), Token Ring and FDDI, 100VG-AnyLAN, ATM LANs, Fibre Channel, Wireless LANs, Bridge Operation, Routing with Bridges

Protocols and Architecture: Protocols, OSI, TCP/IP Protocol Suite
Examples of networks: Novell Netware, Arpanet, and Internet. Examples of Data Communication Services: X.25 Networks, Frame relay, Broad band ISDN and ATM. Physical Layer: Transmission media- Narrow band ISDN: Services-Architecture- Interface, Broad band ISDN and ATM- Virtual Circuits versus Circuit Switching –Transmission in ATM networks. FDDI

Link Layer and Local Area Networks Data link layer: Service provided by data link layer-Error detection and correction Techniques-Elementary data link layer protocols -Sliding Window protocols - Data link layer in HDLC, Internet and ATM . Multiple Access protocols: Channel partitioning protocols: TDM-FDM-Code Division Multiple Access(CDMA) .Random Access protocols : ALOHACSMA and CSMA/CD . Local area Network: LAN addresses-

Address Resolution Protocol-Reverse Address Resolution Protocol. Ethernet: Ethernet Technologies-IEEE standards-Hubs-Bridges and Switches

Text/References:

1. Andrew S. Tanenbaum, "Computer Networks", Pearson Education, 5/e, 2010.
2. James F. Kurose and Keith W. Ross, "Computer Networking" Pearson Education, 6/e, 2012.
3. William Stalling, "Data and Computer Communication", Pearson Education, 7/e, 2004.
4. M. A. Miller, "Data and Network Communication", Thomson Learning, 2001.
5. Douglas E. Comer, "Computer Networks and Internets", Pearson Education, 5/e, 2008.

CSX-210 PRINCIPLES OF PROGRAMMING LANGUAGES [3 1 0 4]

Introduction: Characteristics of programming Languages, Factors influencing the evolution of programming language, developments in programming methodologies, desirable features and design issues. Programming language processors: Structure and operations of translators, software simulated computer, syntax, semantics, structure, virtual computers, binding and binding time.

Elementary and Structured Data Types: Data object variables, constants, data types, elementary data types, declaration, assignment and initialization, enumeration, characters, strings. Structured data type and objects: Specification of data structured types, vectors and arrays, records, variable size data structure, pointers and programmer constructed data structure, Sets files. Sub Program and programmer defined data types: Evolution of data types, abstractions, encapsulations, information hiding, sub programmes, abstract data types.

Sequence Control; Implicit and Explicit sequence control, sequence control with within expression and statements, recursive sub programmes, exception handling, co routines, Scheduled sub programmes, concurrent execution. Data control referencing environments, static and dynamic scope, local data local data referencing environment, shared data: Explicit common environment dynamic scope parameter passing mechanism.

Storage Management: Major run time requirements, storage management phases, static storage management, stack based, heap based storage management. Syntax and translation: General syntactic criteria, syntactic element of a language, stages in translation, formal syntax and semantics.

Operating and Programming Environment: Batch Processing Environments, Embedded system requirements, Theoretical models, Introduction to Functional Programming, Lambda calculus, Data flow language and Object Oriented language, Comparison in various general and special purpose programming languages e.g. Fortran, C, Pascal, Lisp, etc.

Text/References:

1. Terrance W. Pratt, "Programming Languages: Design and Implementation", PHI, 4/e, 2000.
2. R. W. Sebesta, "Concept of Programming Language", Addison Wesley, 10/e, 2012.
3. E. Horowitz, "Programming Languages", Addison Wesley, 2/e, 1984.
4. Ellis Horowitz, "Fundamentals of Programming Languages", Galgotia, 2/e, 1985.
5. K. C. Louden, "Programming Languages-principles and practice", Cengage Learning, 3/e, 2011.

CSX-212 DESIGN AND ANALYSIS OF ALGORITHMS [3 1 0 4]

Data Structures: Trees, binary search trees, heaps, graphs, sets, Data structure for disjoint sets

Basic of Analysis of Algorithm: Introduction to design and analysis of algorithms, Growth of Functions :Asymptotic notations, standard notations and common functions

The Divide and Conquer Method: Recurrences, solution of recurrences by substitution, recursion tree and Master methods, worst case analysis of Merge sort, Quick sort and Binary search, Design & Analysis of Divide and conquer algorithms. string matching etc.

Dynamic Programming: Overall technique, matrix chain problem, optimal binary search trees, Longest Common subsequence(LCS) etc.

The Greedy Method: Overall technique, the knapsack problem, optimal merge pattern, Huffman coding etc

Graph Traversal Techniques: Tree traversal and applications, depth-dirst search, bread-first search, connectivity algorithms, biconnectivity algorithms, etc. Minimum Spanning Trees, Kruskal and Prim's algorithms, single- source shortest paths (Bellman-ford and Dijkstra's algorithm), All-pair shortest paths(Floyd-Warshall algorithm)

Backtracking: Overall technique, generation of combinatorial objects such as graphs, sets, permutations, graph colorings, cliques, Hamiltonian cycles, etc.

Branch and Bound method: Overall method, the 0/1 knapsack problem, the job assignment problem, the traveling salesman problem, etc.

Lower bound theory: Techniques for determining complexity lower bounds of problems, algorithm modeling, application to lower bound on sorting, searching, and merging.

Introduction to the Theory of NP-completeness: Nondeterministic algorithms, complexity classes, NP-completeness, problem reduction, Specific NP-complete problems. Approximation algorithms (Vertex-Cover Problem, Traveling Salesman Problem).

Text/References:

1. Thomas H. Cormen, Charles E. Leiserson, "Introduction to Algorithms", McGraw-Hill, 3/e, 2009.
2. Sartaj Sahni and Ellis Horwitz, "Design and Analysis of Algorithms", Galgotia Publications, 2/e, 2007.
3. Jeffrey D. Ullman and John E. Hopcroft, "Design and Analysis of Algorithms", Pearson Education, 1974.
4. John Kleinberg, Eva Tardos, "Algorithm Design"Pearson publication, 2013.
5. Anany Levitin, "Indroduction to Design & Analysis of Algorithms", Pearson Education, 2/e, 2012.

CSX-224 DATABASE MANAGEMENT SYSTEM LABORATORY [0 0 2 1]

1. Write the queries for Data Definition and Data Manipulation language.
2. Write SQL queries using Logical operators (=,<,>,etc.).
3. Write SQL queries using SQL operators (Between…. AND, IN(List), Like, ISNULL and also with negating expressions).
4. Write SQL query using character, number, date and group functions.
5. Write SQL queries for Relational Algebra (UNION, INTERSECT, and MINUS, etc.).
6. Write SQL queries for extracting data from more than one table (Equi-Join, Non-Equi-Join , Outer Join)
7. Write SQL queries for sub queries, nested queries.
8. Write programs by the use of PL/SQL.
9. Concepts for ROLL BACK, COMMIT & CHECK POINTS.
10. Create VIEWS, CURSORS, and TRIGGRS & write ASSERTIONS.
11. Create FORMS and REPORTS.
12. Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
13. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.
14. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
15. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

16. i) Creation of simple PL/SQL program which includes declaration section, executable section and exception – Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
    ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
17. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
18. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.
19. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.
20. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
21. Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
22. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
23. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

* Students are advised to use **Developer 2000/Oracle-10i** or higher version or other latest version for above listed experiments. However depending upon the availability of software's, students may use **Power Builder /SQL SERVER**. Mini Project may also be planned & carried out through out the semester to understand the important various concepts of Database.
This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

ECX-216 MICRO-PROCESSOR AND PROGRAMMING LABORATORY [0 0 2 1]

8085 Assembly Language Program like addition, subtraction, multiplication, division, sorting, ascending, descending, advanced programming concepts, looping, branching, interfacing.

Introduction to 8051 Assembly Language Program like addition, subtraction, multiplication, division, sorting, ascending, descending, advanced programming concepts.

Experimentation to be supported by computer simulation using Keil Software and Hardware kits

CSX-228 DATA COMMUNICATION LABORATORY [0 0 2 1]

1. Making Straight, Rollover and Cross-Over cables
2. Cable & RJ-45 Jack outlet installation
3. Installation of NIC Card & using TCP/IP
4. Design, build & test a simple communication system
5. Overview and basic Configuration of Router
6. Router show Command
7. Basic LAN Setup
8. Designing & Implementing LAN using subnetting
9. Study of Amplitude Modulation
10. Study of frequency Modulation
11. Study of ASK Modulation
12. Study of FSK Modulation
13. Simple point-to-point communication & error detection
14. Implementation of STOP and Wait protocol
15. Implementation of Sliding Window protocol

This is only the suggested list of practicals. Instructor may frame additional practicals relevant to the course contents

For ECE Deptt.

CSX-206 Data Structures and Algorithms **[2 0 0 2]**

Introduction: Basic Terminology, Elementary Data Organization, Structure operations, Algorithm Complexity and Time-Space trade-off

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C++, Character string operation, Array as Parameters, Ordered List, Sparse Matrices and Vectors.

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Queues: Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues**.**

Linked list: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list, Garbage Collection and Compaction.

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm**.**

Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

Text/References:

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., 1983.
2. R. Kruse and A. Ryba, "Data Structures and Program Design in C", Pearson Education Asia, Delhi, 2002.
3. Yedidyah Langsam, Moshe J. Augenstein and Aaron M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India, 2/e, 1995.
4. Bruno R. Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", John Wiley & Sons, 1998.
5. Richard F. Gilberg and Behrouz A. Forouzan, "Data Structure – A pseudo code approach with C++", Cengage Learning, 2001.

CSX-216 Data Structures and Algorithm Lab. **[0 0 2 1]**

1. Liner Search for an array of values.
2. Binary Search for an array of values.
3. Bubble Sort for an array of values.
4. Push. Pop and Display operations of a Stack.
5. To convert an expression in infix notation into postfix notation.
6. To evaluate an expression in postfix notation.
7. Insert, Delete and Display operation on a simple queue.
8. Insert, Delete and Display operation on a circular queue.
9. Insertion, Deletion and Display of Linked List.
10. Construction of Binary Search Tree (BST).
11. Pre-order traversed of a BST.
12. Inorder traversed of a BST.
13. Postorder traversed of a BST.

This is only the suggested list of practicals. Instructor may frame additional practicals relevant to the course contents

5TH SEMESTER:

CSX-301 COMPUTER NETWORKS [3 0 0 3]

Introduction: Introduction to Computer Network and Physical Layer

Types of Networks: Broadcast and Point-to-point- LAN-MAN-WAN- Wireless networks.

Architecture and Reference Models: Layered architecture- OSI reference model, TCP/IP reference model –Internet Protocol Stack – Network Entities in Layers- Connection oriented and Connection less services,

Internetworking: Principles of Internetworking, Connectionless Internetworking, the Internet Protocol, Routing Protocol, IPv6 (IPng)

Distributed Applications: Abstract Syntax Notation One (ASN.l), Electronic Mail-SMTP and MIME, Uniform Resource Locators (URL) and Universal Resource Identifiers (URI), Hypertext Transfer Protocol (HTTP)

Network Layer and Routing: Network Service model – Datagram and Virtual circuit service-Routing principles-Link state routing-distance vector routing-hierarchical routing-multicast routing-IGMP Internet Protocol (IP): IPv4 addressing-routing and forwarding datagram-datagram format-datagram fragmentation- ICMP- Network Address Translators (NATs)-IPv6 packet format-transition from IPv4 to IPv6-Mobile IP. Routing in the Internet: Intra Autonomous System Routing: RIP - Inter Autonomous System Routing: BGP.

Transport Layer: Transport Layer Services-Relationship between Transport Layer and Network Layer-Transport Layer in Internet-Multiplexing and De multiplexing. Connectionless Transport: UDP-Segment structure-Checksum Connection Oriented Transport: TCP-TCP connection-TCP Segment Structure-Round trip Time estimation and Time out-Reliable Data transfer-Flow control-TCP connection Management. Congestion Control: Causes and costs of congestion-Approaches to congestion control- TCP congestion control: Fairness. TCP Timers: Retransmission Timer- Persistence Timer- Keepalive Timer- TIME-WAIT Timer. TCP Options: EOP-NOP-MSS-WSF-Timestamp.

Application Layer and Network Security: Application Layer Protocols - WWW and HTTP-File transfer Protocol: FTP Commands and Replies – Domain Name System (DNS) - DHCP- SMTP- POP- IMAP - SNMP. Security in Computer Networks: Principles of Cryptography-Symmetric key-Public key-authentication protocols -Digital Signatures – Firewalls. Security in different Layers: Secure E-mail- SSL – IP security.

Text/References:

1. James F. Kurose and Keith W. Ross, "Computer Networking – A Top-Down Approach Featuring the Internet", 5/e Pearson Education, 2002.
2. Behrouz A. Forouzan, "TCP/IP Protocol Suite", Tata McGraw Hill, 3/e, 2006.
3. S. Keshav, "An Engineering Approach to Computer Networking", Pearson education, 2002.
4. F. Halsall, "Data Communication, Computer Networks and Open Systems", Addison Wesley, 4/e, 1996
5. Andrew S. Tanenbaum, "Computer Networks", 4/e, Pearson education, 5/e, 2010.

CSX-303 SOFTWARE ENGINEERING [3 0 0 3]

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

Design Engineering: Design process and Design quality, Design concepts, the design model.

Creating an architectural design: Software architecture, Data design, Architectural styles and patterns, Architectural Design.

Object-Oriented Design: Objects and object classes, An Object-Oriented design process, Design evolution.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modelling Language (UML)

Text/References:

1. K. K. Agarwal and Yogesh Singh, "Software Engineering", New Age International Publishers, 2008.
2. James F. Peters and Witold Pedrycz, "Software Engineering, an Engineering approach", John Wiely, 1999.
3. Waman S. Jawadekar, "Software Engineering principles and practice", McGraw-Hill Companies, 2004.
4. Roger S. Pressman, "Software Engineering, A practitioner's Approach", McGrawHill, 7/e, 2009.
5. Ian Sommerville, "Software Engineering", Addison Wesley, 7/e, 2004.

CSX-305 OPERATING SYSTEMS **[3 0 0 3]**

Operating systems objectives, services and functions: Characteristics of Modern Operating Systems, Characteristics of Batch and multiprogramming operating systems. Comparisons between real time systems and time-sharing systems, Operating system services and kernel features.

I/O management, I/O devices: Introduction to I/O management, I/O devices, Concepts of threading, Organization of I/O functions, polling, various modes of data transfer, Hardware/Software interface, I/O buffering.

Disk scheduling policies and processes: Motivation for disk scheduling policies, Introduction to processes management, operating system views of processes, various process transition states, Introduction to Processor scheduling, Introduction to various types of schedulers, Performance criteria in scheduling algorithms, Concept of FCFS

scheduling algorithm, Concept of priority scheduling algorithm like SJF, Concept of non-preemptive and preemptive algorithms, Concept of round-robin scheduling algorithm, , Concept of multi-level queues, feedback queues.

Concurrency control schemes: Various approaches to concurrency control schemes, Concept of prouder/consumer problem, Mutual Exclusion, Concept of mutual exclusion first and second algorithm, Concept of mutual exclusion third algorithm including introduction and characteristics of semaphores, Introduction to Mutual exclusion with semaphores, Introduction to Interprocess Communication and Synchronization, Critical regions and Conditional critical regions in a Semaphore. Introduction to monitors, various modes of monitors, Issues in message implementation, Concept of mutual exclusion with messages.

Dead Locks: Concept of Deadlocks, issues related to its prevention, avoidance and detection/recovery, Concept of deadlock prevention and its avoidance, Concept of deadlock detection and recovery.

Memory Management: Need of Memory management and its requirements, paging, segmentation, concept of fragmentation. Characteristics of contiguous & non-contiguous allocation techniques, Detail study of fragmentation, Virtual memory management, introduction to page-replacement, Need of various page-replacement policies, Concept of FIFO and optimal page-replacement algorithms, Concept of LRU approximation and its page-replacement algorithm, Concept of allocation algorithms.

File management System: Need of file management, its requirements, User's and operating system's view of file system, Concept of file directories and file sharing, Motivation for disk space management, Characteristics of file related system services, Generalization of file services**.**

Text/References:

1. Peterson and Silberschatz, "Operating System Concepts", Addison-Wesley, 8/e, 2008.
2. M. Milenkoviac, "Operating Systems Concepts and Design", Tata McGraw Hill, 2/e, 1992.
3. Charles Crowley, "Operating Systems a Design Oriented Approach", Tata McGraw Hill, 1996.
4. Andrews S. Tanenbaum, "Modern Operating Systems", Pearson Education, 2/e, 2001.
5. W. Richard Stevens, "Linux Network Programming", PHI, 2003.

HMX-311 FINANCIAL AND COST ACCOUNTING **[3 0 0 3]**

From HUM department

MAX-XXX NUMERICAL METHODS **[3 1 0 4]**

Roots of algebraic and transcendental equations, Bisection method, Regula – Falsi method, Newton –Raphson method, Bairstow's method and Graeffe's root squaring method.

Solution of simultaneous algebraic equations, matrix inversion and eigen-value problems, triangularisation method, Jacobi's and Gauss-Siedel iteration method, partition method for matrix inversion, power method for largest eigen-value and Jacobi's method for finding all eigen-values.

Finite differences, interpolation and numerical differentiation, forward, backward and central differences, Newton's forward, backward and divided difference interpolation formulas, Lagrange's interpolation formula, Stirling's and Bessel's central difference interpolation formulas, numerical differentiation using Newton's forward and backward difference formulas and numerical differentiation using Stirling's and Bessel's central difference interpolation formulas.

Numerical integration, Trapezoidal rule, Simpson's one-third rule and numerical double integration using Trapezoidal rule and Simpson's one-third rule.

Taylor's series method, Euler's and modified Euler's methods, Runge-Kutta fourth order methods for ordinary differential equations, simultaneous first order differential equations and second order differential equations.

Boundary value problems, finite difference methods for boundary value problems.

Partial differential equations, finite difference methods for elliptic, parabolic and hyperbolic equations.

Text/References:

1. S. S. Sastry, "Introductionary Methods of Numerical Analysis", Prentice Hall of India, 5/e, 2012.
2. S. C. Chapra and R. P. Canale, "Numerical Methods for Engineers", McGraw Hill Book Company, 6/e, 2009.
3. B. S. Grewal, "Numerical Methods", Khanna Publishers, Delhi.
4. S. Kalavathy, "Numerical Methods", Cengage Publishers, New Delhi.
5. Richard L. Burden and J. Douglas Faires, "Numerical Anlaysis" , Cengage Learning ,9/e, 2010.

CSX-321 COMPUTER NETWORKS LABORATORY [0 0 2 1]

1. Introduction to Network Simulator OPNET/NS2.
2. Evaluate Ethernet Delay and Load Statics of Switched Ethernet
3. Comparative investigation on Hub and Switch as Interconnecting Device for verifying performance of LAN with various applications
4. Simulation and comparative investigations on the performance issues of a unicast network with FTP applications.
5. Comparative study of a network with multiple TCP and UDP applications.
6. Simulation and study of multicast routing protocols.
7. Simulation of Wireless data Network with different with physical characteristics.
8. Implementation of CSMA/CD Protocol and its comparative investigation with ALOHA Protocol.
9. Design and Implementation of Simple Transfer Protocol in C/ C++.
10. Design of substitution Cipher in C/ C++.
11. Design of Transposition Cipher in C/ C++.
12. Design of Public Key Algorithm in C/ C++.

* Students are advised to use **QualNet/NS2** for above listed experiments.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

CSX-323 SOFTWARE ENGINEERING Lab. [0 0 2 1]

1. Introduction and project definition
2. Software process overview
3. Project planning
4. Software requirements and RequisitePro
5. Introduction to UML and use case diagrams
6. System modeling (DFD and ER)
7. Flow of events and activity diagram
8. OO analysis: discovering classes
9. Interaction diagrams: sequence and collaboration diagrams
10. Software Design: software architecture and object-oriented design
11. State Transition Diagram

12. Component and deployment diagrams
13. Software testing

CSX-325 OPERATING SYSTEMS LABORATORY [0 0 2 1]

1. Simulation of the CPU scheduling algorithms a) Round Robin b) SJF c) FCFS d) Priority
2. Simulation of MUTEX and SEMAPHORES.
3. Simulation of Bankers Deadlock Avoidance and Prevention algorithms.
4. Implementation of Process Synchronization (Reader-Writer, Sleeping Barber and Dining Philosopher's Problem)
5. Simulation of page Replacement Algorithms a) FIFO b) LRU c) LFU
6. Simulation of paging techniques of memory management.
7. Simulation of file allocation Strategies a) Sequential b) Indexed c) Linked
8. Simulation of file organization techniques a) Single Level Directory b) Two Level c) Hierarchical d) DAG
9. To automate the allocation of IP addresses i.e. to set and configure the DHCP server and DHCP client.
10. To share files and directories between RedHat Linux operating systems i.e. To set and configure the NFS server and NFS clients.
11. To share files and directories between Red Hat Linux and Windows operating systems i.e. To set and configure the samba server.
12. To set and configure the DNS (Domain Name Server).
13. To set and configure the print server and to share printers between Windows and Red Hat Linux operating systems.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

DEPARTMENTAL ELECTIVE (DE) –I

CSX-331 ADVANCED PROGRAMMING CONCEPTS USING JAVA [3 0 0 3]

Overview of Basic OOP Concepts: Need for object-oriented paradigm: Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, datatypes, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, classes and objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling, inheritance, super keyword, polymorphism- method overriding, abstract classes.

Packages and Interfaces: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring packages – Java.io, Java.util.

Exception handling and multithreading: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes.The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – boarder, grid, flow, card and grib bag.

Applets: Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Swing: Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing-JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Networking: Basics of network programming, addresses, ports, sockets, simple client server program, multiple clients, Java .net package Packages – java.util,

Text/References:

1. J. Nino and F. A. Hosch, "An Introduction to programming and OO design using Java", John Wiley & Sons, 3/e, 2008.
2. T. Budd, "An Introduction to OOP", Addison-Wesley, 3/e, 2001.
3. Y. Daniel Liang, "Introduction to Java programming", Prentice Hall, 9/e, 2012.
4. R.A. Johnson, "An introduction to Java programming and object oriented application development", Cengage Learning, 2006.
5. Cay S. Horstmann and Gary Cornell, Core Java 2, Vol 1, Fundamentals, Pearson Education, 7/e.

CSX-333 WINDOWS PROGRAMMING [3 0 0 3]

The Windows Architecture: The Concept of Handlers, The concept of Windows class, Registering a Window class, Style like CS_HREDRAW, Instance handlers, Icon handlers, Cursor handlers, The concept of Windows class, Registering a Window class, Style like CS_HREDRAW, Instance handlers, Icon handlers, Cursor handlers, Menu name, Create a Window, Class names-predefined and user defined, Window name, Draw styles, Width Height etc., Parent-Child Windows, The concept of Window Messages, Messages Queue

Compiling and Linking for Windows: Compiling for Windows, Memory models in Windows, Linking, The stub file

I/O Techniques: Drawbacks of Windows, Screen printing, determining the size of Windows, The concept of device context, Device context handles, Text formatting, Using fonts, the concept of scroll bars, setting scroll bar range and position

Keyboard: Keyboard messages, Virtual key codes, Parameters like repeat count, scan code etc, System keys like Alt-Tab, Ctrl-Esc etc.

Mouse: Mouse action, Mouse messages, Activating Windows, Change mouse cursors File I/O: Win 3.1 file functions, Buffered file I/O, Common dialog boxes, Open file name structure

Child Windows: The concept of child Windows, Child Window control, Child Window control classes, The static class, The button class, Button messages, Push buttons, Check boxes, Radio buttons, Edit class, Edit control messages, Edit styles, The list box class.

Menus: Creating menus, working of menus, Using menus Inputs, Creating pop-ups to top level menus, Menu messages, Menu templates, Using system menus, Messages from menus

Dialog Boxes: Concept, Control state, working with dialog boxes.

Printing: Obtaining printer's driver content, determining the printer name, Notifying the print job, Page breaks, determining device capabilities, printing graphics, about printing.

Graphics: Fonts, Portrait of a character, Types of fonts, creating logical fonts, Font handle.

Graphical Device Interface: Need, The device context, determining device capabilities.

Drawing Graphics: Pixels, Lines, Rectangles, and Circles.

Bitmaps: Bitmap handle, Obtaining handles, internal representation of bitmaps, Creating bitmaps, Memory device context.

Memory Management: Memory handles and locks, Problems of Windows memory handling, Intel's segmented memory architecture, 80286 protected mode, the memory API, Kernel Macros

Windows Resources: Accessing resources, loading techniques.

Windows Timer: Timekeeping on the PC, Windows interaction with 8259 chip, the timer API function.

Windows Chipboard: Concept, The clipboard API, Working of clipboard, the concept of clipboard viewers, Clipboard viewer message.

Text/References:

1. Ben Ezzell with Jim Blaney, "NT4/Windows 95 Developer's Handbook", BPB Publications, 1997.
2. Charies Petzold, "Programming Windows 95", Microsoft Press, 1996.
3. Richard J. Simson, "Windows NT Win 32, API Super Bible", SAMS 1997.
4. James F. Kurose and Keith W. Ross, "Computer Networking", Pearson Education, 6/e, 2012.
5. Nance, "Introduction to Networking", PHI, 4/e, 2002.

CSX-335 UNIX AND SHELL PROGRAMMING [3 0 0 3]

Introduction to Unix:- Architecture of Unix, Features of Unix , Unix Commands – PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip.

Unix Utilities:- Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, unlink, du, df, mount, umount, find, unmask, ulimit, ps, w, finger, arp, ftp, telnet, rlogin.Text processing utilities and backup utilities , detailed commands to be covered are tail, head , sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm, cmp, diff, tr, awk,
Cpio

Introduction to Shells: Unix Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, Command-Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment Customization.

Filters : Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count characters, Words or Lines, Comparing Files.

Grep : Operation, grep Family, Searching for File Content.

Sed : Scripts, Operation, Addresses, commands, Applications, grep and sed.

awk: Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays, String Functions, String Functions, Mathematical Functions, User – Defined Functions, Using System commands in awk, Applications, awk and grep, sed and awk.

Interactive Korn Shell: Korn Shell Features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, Options, Startup Scripts, Command History, Command Execution Process.

Korn Shell Programming: Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

Interactive C Shell: C shell features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, On-Off Variables, Startup and Shutdown Scripts, Command History, Command Execution Scripts.

C Shell Programming: Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

File Management: File Structures, System Calls for File Management – create, open, close, read, write, lseek, link, symlink, unlink, stat, fstat, lstat, chmod, chown, Directory API – opendir, readdir, closedir, mkdir, rmdir, umask.

Text/References:

1. Behrouz A. Forouzan, "Unix and shell Programming", Cengagae Learning, 2003.
2. Sumitabha Das , "Your Unix the ultimate guide", McGraw Hill, 2/e, 2005.
3. Graham Glass and King Ables, "Unix for programmers and users", Prentice Hall, 3/e, 2003.
4. B. W. Kernighan and R. Pike, "Unix programming environment", PHI/Pearson Education, 1983.
5. Eric Foster-Johnson, John C. Welch and Micah Anderson, "Beginning shell scripting", Wile Y-India, 2005.

CSX-351 ADVANCED PROGRAMMING CONCEPTS USING JAVA LABORATORY [0 0 2 1]

Objectives:

- To make the student learn the application of advanced object oriented concepts for solving problems.
- To teach the student to write programs using advanced Java features to solve the problems

1. a) Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
   b) Write a Java program to multiply two given matrices.
2. a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.
   b) Write a Java program for sorting a given list of names in ascending order.
3. Write a java program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
4. a) Develop an applet that displays a simple message.
   b) Develop an applet that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.
5. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.
6. Write a Java program for handling mouse events.
7. a) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.
   b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
8. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.
9. Write a Java program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. (Use java.net)
10. a) Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No light is on when the program starts.
    b) Write a Java program that allows the user to draw lines, rectangles and ovals.
11. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are seperated by commas. Write a java program to display the table using JTable component.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

CSX-353 WINDOWS PROGRAMMING LAB [0 0 2 1]

Instructor may frame Practicals relevant to the course contents.

CSX-355 UNIX AND SHELL PROGRAMMING LAB [0 0 2 1]

To teach students various unix utilities and shell scripting

1. a) Login to the system
   b) Use the appropriate command to determine your login shell
   c) Use the /etc/passwd file to verify the result of step b.
   d) Use the who command and redirect the result to a file called myfile1. Use the more command to see the contents of myfile1.
   e) Use the date and who commands in sequence (in one line) such that the output of date will display on the screen and the output of who will be redirected to a file called myfile2. Use the more command to check the contents of myfile2.

2) a) Write a sed command that deletes the first character in each line in a file.
   b) Write a sed command that deletes the character before the last character in each line in a file.
   c) Write a sed command that swaps the first and second words in each line in a file.

3. a) Pipe your /etc/passwd file to awk, and print out the home directory of each user.
   b) Develop an interactive grep script that asks for a word and a file name and then tells how many lines contain that word.
   c) Repeat
   d) Part using awk

4. a) Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.
   b) Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.
   c) Write a shell script that determines the period for which a specified user is working on the system.

5. a) Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
   b) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
6. a) Write a shell script that computes the gross salary of a employee according to the following rules:
   i) If basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.
   ii) If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic. The basic salary is entered interactively through the key board.
   b) Write a shell script that accepts two integers as its arguments and computers the value of first number raised to the power of the second number.

7. a) Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
   b) Write shell script that takes a login name as command – line argument and reports when that person logs in
   c) Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.

8. a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
   b) Develop an interactive script that ask for a word and a file name and then tells how many times that word occurred in the file.

c) Write a shell script to perform the following string operations:
i) To extract a sub-string from a given string.
ii) To find the length of a given string.

9. Write a C program that takes one or more file or directory names as command line input and reports the following information on the file:
i) File type
ii) Number of links
iii) Read, write and execute permissions
iv) Time of last access
(Note: Use stat/fstat system calls)

10. Write C programs that simulate the following unix commands:
a) mv
b) cp (Use system calls)

11. Write a C program that simulates ls Command (Use system calls / directory API)

For ECE Deptt

CSX-305 OPERATING SYSTEMS [3 0 0 3]

Operating systems objectives, services and functions: Characteristics of Modern Operating Systems, Characteristics of Batch and multiprogramming operating systems. Comparisons between real time systems and time-sharing systems, Operating system services and kernel features.

I/O management, I/O devices: Introduction to I/O management, I/O devices, Concepts of threading, Organization of I/O functions, polling, various modes of data transfer, Hardware/Software interface, I/O buffering.

Disk scheduling policies and processes: Motivation for disk scheduling policies, Introduction to processes management, operating system views of processes, various process transition states, Introduction to Processor scheduling, Introduction to various types of schedulers, Performance criteria in scheduling algorithms, Concept of FCFS scheduling algorithm, Concept of priority scheduling algorithm like SJF, Concept of non-preemptive and preemptive algorithms, Concept of round-robin scheduling algorithm, , Concept of multi-level queues, feedback queues.

Concurrency control schemes: Various approaches to concurrency control schemes, Concept of prouder/consumer problem, Mutual Exclusion, Concept of mutual exclusion first and second algorithm, Concept of mutual exclusion third algorithm including introduction and characteristics of semaphores, Introduction to Mutual exclusion with semaphores, Introduction to Interprocess Communication and Synchronization, Critical regions and Conditional critical regions in a Semaphore. Introduction to monitors, various modes of monitors, Issues in message implementation, Concept of mutual exclusion with messages.

Dead Locks: Concept of Deadlocks, issues related to its prevention, avoidance and detection/recovery, Concept of deadlock prevention and its avoidance, Concept of deadlock detection and recovery.

Memory Management: Need of Memory management and its requirements, paging, segmentation, concept of fragmentation. Characteristics of contiguous & non-contiguous allocation techniques, Detail study of fragmentation, Virtual memory management, introduction to page-replacement, Need of various page-replacement policies, Concept of FIFO and optimal page-replacement algorithms, Concept of LRU approximation and its page-replacement algorithm, Concept of allocation algorithms.

File management System: Need of file management, its requirements, User's and operating system's view of file system, Concept of file directories and file sharing, Motivation for disk space management, Characteristics of file related system services, Generalization of file services**.**

Text/References:

1. Peterson and Silberschatz, "Operating System Concepts", Addison-Wesley, 8/e, 2008.
2. M. Milenkoviac, "Operating Systems Concepts and Design", Tata McGraw Hill, 2/e, 1992.
3. Charles Crowley, "Operating Systems a Design Oriented Approach", Tata McGraw Hill, 1996.
4. Andrews S. Tanenbaum, "Modern Operating Systems", Pearson Education, 2/e, 2001.
5. W. Richard Stevens, " Linux Network Programming", PHI, 2003.

CSX-325 OPERATING SYSTEMS LABORATORY [0 0 2 1]

1. Simulation of the CPU scheduling algorithms a) Round Robin b) SJF c) FCFS d) Priority
2. Simulation of MUTEX and SEMAPHORES.
3. Simulation of Bankers Deadlock Avoidance and Prevention algorithms.
4. Implementation of Process Synchronization (Reader-Writer, Sleeping Barber and Dining Philosopher's Problem)
5. Simulation of page Replacement Algorithms a) FIFO b) LRU c) LFU
6. Simulation of paging techniques of memory management.
7. Simulation of file allocation Strategies a) Sequential b) Indexed c) Linked
8. Simulation of file organization techniques a) Single Level Directory b) Two Level c) Hierarchical d) DAG
9. To automate the allocation of IP addresses i.e. to set and configure the DHCP server and DHCP client.
10. To share files and directories between RedHat Linux operating systems i.e. To set and configure the NFS server and NFS clients.
11. To share files and directories between Red Hat Linux and Windows operating systems i.e. To set and configure the samba server.
12. To set and configure the DNS (Domain Name Server).
13. To set and configure the print server and to share printers between Windows and Red Hat Linux operating systems.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

6TH SEMESTER:

CSX-302 THEORY OF COMPUTATION [3 1 0 4]

Basics in Theory of Computations: Basic concepts of strings, alphabets, languages, Principles of Mathematical Induction.

Languages and Grammars: Construct of a language, Grammar, Chomsky Classification of Formal Languages.

Finite Automata: Automata and Applications of Automata Theory, Deterministic and Non-Deterministic FA, Comparison and Equivalence of DFA and NFA.

Regular Expressions: Regular Expression, Equivalence of Regular Expression and Finite Automata, Equivalence of Regular Grammar and Finite Automata, Regular and Non- Regular Languages, Pumping Lemma for Regular Sets

Finite State Machines: Moore and Mealy Machines , Equivalence of Moore and Mealy Machines.

Context Free Language: Context Free Grammar, Derivation trees, Context Free Grammar Simplification, Chomsky & Greibach Normal forms, Ambiguities.

Pushdown Automata: Definition, Equivalence of PDA by Empty Store and PDA by Final State. Construction of PDA for CFLs.

Turing Machines: Introduction and Turing Machine Model, Computable functions and languages. Techniques for construction of Turing machines, Church's Hypothesis.

Undecidability: Recursive and recursively enumerable languages, Rice theorem, Post's correspondence problem.

Text/References:

1. J E Hopcroft And J D Ullman, "Introduction to Automata Theory, Languages and Computation", Narosa Publishers 2002.
2. K L P Mishra and N Chandrasekaran, "Theory of Computer Science", Prentice Hall Inc, .2002
3. Harry R Lewis and Chritos H Papadimitriou, "Elements of the Theory of Computation", Pearson Education 2001.
4. Peter Linz, "An Introduction to Formal Languages and Automata", Narosa Publishers 2002.
5. Michael Sipser, "Introduction to the theory of computation ", Cengage Learning, New Delhi

CSX-304 INFORMATION SECURITY SYSTEMS [3 0 0 3]

Overview: Services, Mechanisms, and Attacks, the OSI Security Architecture, A Model for Network, Security.

Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Stegnography.

Block Ciphers And The Data Encryption Standard: Simplified DES, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles, Block Cipher Modes of Operation.

Introduction To Finite Fields: Groups, Rings, and Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields of the Form GF(p), Polynomial Arithmetic, Finite Fields of the Form GF(2n).

Advanced Encryption Standard: Evaluation Criteria for AES, The AES Cipher.

Contemporary Symmetric Ciphers: Triple DES, Blowfish, RC5, Characteristics of Advanced Symmetric Block

Ciphers, RC4 Stream Cipher.

Confidentiality Using Symmetric Encryption: Placement of Encryption Function, Traffic Confidentiality, Key Distribution, Random Number Generation.

Public-Key Encryption and Hash Functions: Introduction to Number Theory: Prime Numbers, Format's and Euler's Theorems, Testing for Primality, The Chinese Remainder Theorem, Discrete Logarithms.

Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, the RSA Algorithm, Recommended Reading and Web Site, Key Terms, Review Questions, and Problems.

Key Management and Other Public-Key Cryptosystems: Key Management, Diffie-Hellman Key Exchange, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.

Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and MACs.

Hash Algorithms: MD5 Message Digest Algorithm, Secure Hash Algorithm, RIPEMD-160, and HMAC.

Digital Signatures and Authentication Protocols: Digital Signatures, Authentication Protocols, Digital Signature Standard.

Network Security Practice: Authentication Applications: Kerberos, X.509 Authentication Service, Electronic Mail Security: Pretty Good Privacy, S/MIME. IP Security: IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management, Web Security: Web Security Considerations, Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction.

System Security: Intruders: Intruders, Intrusion Detection, Password Management, Malicious Software: Viruses and Related Threats, Virus Countermeasures, Firewalls: Firewall Design Principles, Trusted Systems.

Text/References:

1. W. Stallings, "Cryptography and Network Security Principles and Practice", Prentice Hall, 5/e, 2010.
2. W. Trappe & L. C. Washington, "Introduction to Cryptography with Coding Theory", Prentice-Hall, 2/e, 2005.
3. D. Stinson, "Cryptography: Theory and Practice", Chapman & Hall, 3/e, 2005.
4. Charlie Kaufman, Radia Perlman, "Network Security: Private Communication in a Public World", Prentice Hall, 2/e, 2002.
5. Michael E. Whitman, "Principles of information Security" , Cengage Learning, 4/e, 2011.

HMX-201 ENGINEERING ECONOMICS AND INDUSTRIAL MANAGEMENT [3 0 0 3]

Definition and Scope of Engineering Economics: Concept of revenue and costs, break-even analysis. Law of demand & supply, time value of money, present and future worth methods.

Decision Making: Decision making process, decision making under risk certainty, uncertainty and conflict.

Replacement and maintenance Analysis: Types of maintenance, determination of economic life of an asset, replacement of items that fail suddenly and that fail over a period of time.

Methods of depreciation: straight line method, sum-of-the year's digest method, declining balance method, sinking fund method and service output method of depreciation.

Inventory control: Introduction and objective of inventory control, purchase model with instantaneous replenishment, model with shortages, price break model, ABC analysis.

Forecasting: Demand forecasting by quantitative and qualitative techniques, applications of demand forecasting.

Make or Buy Decision: Criteria for make or buy, approaches for make or buy decision.

Value Engineering Analysis: Value analysis vs. value engineering function, aims and value engineering procedure, advantages & applications.

Linear Programming: Linear programming as a tool of decision making, graphical and Simplex Methods and applications in decision making.

Text/References:

1. R. Panaeerselvam, "Engineering Economics", Prentice Hall of India, 2001.
2. Eugene L. Grant, W. Grant Ireson and Richard S. Leavenworth, "Principles of Engineering Economy", Cengage Learning, 4/e, 2011.
3. H. A. Taha, "Operations Research: An Introduction", Prentice-Hall of India, 9/e, 2010.
4. N. D. Vohra, "Quantitative Techniques in Managerial Decision Making", Tata McGraw Hill, 4/e, 2010.
5. C. Dougherty, "Introduction to Econometrics", Oxford University Press, 4/e, 2011.

CSX-306 SYSTEM PROGRAMMING [3 0 0 3]

Introduction: Introduction to Software processors, Translators and Loaders, Interpreters

Assemblers: Elements of Assembly Language Programming, Design of Two-Pass assemblers

Macros and Macro Processors: Macro Instructions, Features of a Macro facility, Implementation of Two pass Macro.

Compilers: Aspects of Compilation, Phases of compilation, Scanning and Parsing, Compilation of Expressions, Compilation of Control Structures Code Generation and Code optimization techniques, Compiler Writing Tools

Loaders & Linkage Editors: Loading Linking and Relocation, Overview of Linkage Editing, Linking for Program Overlay.

Editors and debuggers: introduction to editors, types of editor, design of an editor, debug monitors, introduction to various debugging techniques, turbo c++ debuggers.

Grammar and automation: introduction to grammar, types of grammar, acceptability of grammar, introduction to automation, characteristics of automation, finite control, transition system, finite automation. Case study on LEX and YACC**.**

Introduction to Operating systems: Introduction, Operating System Structures, Process Management, Memory management, I/O systems, Distributed Operating Systems

Text/References:

1. L. L. Beck, "Systems Software: An Introduction to Systems Programming", Addison-Wesley, 2001.
2. J. J. Donovan, "Systems Programming", Mc-Graw Hill, 1972.
3. D. M. Dhamdhere, "Introduction to Systems Software", Tata Mc-Graw Hill, 2000.
4. P. Glingaert, "Assembles Loaders and Compilers", Prentice Hall, 1972.
5. A. V. Aho and J. D. Ullman, "Principles of compiler Design", Addison Wesley/Narosa, 1985.

CSX-308 COMPUTER GRAPHICS AND ANIMATION [3 0 0 3]

Introduction: Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms

2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland – Hodgeman polygon clipping algorithm.

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods.

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.

Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

Text/References:

1. Donald Hearn and M. Pauline Baker, "Computer Graphics",PHI/Pearson Education, 2/e, 1994.
2. Zhigand Xiang and Roy Plastock, , "Computer Graphic", Schaum's outlines, Tata Mc- Graw hill, 2/e, 2000.
3. David F. Rogers, "Procedural elements for Computer Graphics", Tata McGraw hill, 2/e, 1997.
4. W. Neuman and R. F. Sproul , "Principles of Interactive Computer Graphics", Tata McGraw Hill, 1973.
5. John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, "Computer Graphics Principles & practice", Addison-Wesley, 3/e, 2013.

CSX- 324 INFORMATION SECURITY SYSTEMS LABORATORY [0 0 2 1]

Implementation of the followings in any High Level Programming Language:

1. Transposition Techniques, Stegnography.
2. Block Ciphers And The Data Encryption Standard
3. Random Number Generation.
4. Testing for Primality, The Chinese Remainder Theorem
5. The RSA Algorithm.
6. Elliptic Curve Cryptography.
7. Hash Algorithms: MD5 Message Digest Algorithm, Authentication Protocols.
8. System Security: Firewalls: Firewall Design Principles

CSX-326 SYSTEM PROGRAMMING LABORATORY [0 0 2 1]

1. Design and Implementation of an Editor in any language.
2. Design and Implementation of One Pass Assembler in any language.
3. Design and Implementation of Two Pass Assembler in any language.

4. Implementation of various search techniques: Linear and Binary Search.
5. Implementation of various sorting techniques: Bucket sort, Merge Sort, Heap Sort
6. Implementation of Lexical Analyzer.
7. Implementation of Top Down Parser.
8. Implementation of Bottom Up Parser.
9. Design and Implementation of Two Pass Macro- Processor.
10. Study of LEX and YACC.

CSX-328 COMPUTER GRAPHICS AND ANIMATION LABORATORY [0 0 2 1]

1. To draw a line using DDA Algorithm.
2. To draw a line using Bresenham's Algorithm.
3. To draw a circle using trigonometric Algorithm.
4. To draw a circle using Bresenham's Algorithm.
5. To draw a circle using Midpoint Algorithm.
6. To draw an ellipse using Trigonometric Algorithm.
7. To draw an ellipse using Midpoint Algorithm.
8. To translate an object with translation parameters in X and Y directions.
9. To scale an object with scaling factors along X and Y directions.
10. To rotate an object with a certain angle.
11. To perform composite transformations of an object.
12. To clip line segments against windows.
13. Demonstrate the properties of Bezier Curve.
14. Run a sample session on Microsoft Windows including the use of Paintbrush.
15. Implementation of simple graphics animation.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

DEPARTMENTAL ELECTIVE (DE)-I

CSX-332 ADVANCED OPERATING SYSTEM [3 0 0 3]

Process Synchronization: Concepts of processes, Concurrent processes, Threads, Overview of different classical synchronization problems, Monitors, Communicating Sequential processes (CSP)

Process deadlocks: Introduction, causes of deadlocks, Deadlock handling strategies, Models of deadlock

Distributed operating system: Architectures, Issues in Distributed operating systems, Limitations of Distributed Systems, Lamport's logical clock, Global states, Chandy-Lampert's global state recording algorithm,Basic concepts of Distributed Mutual Exclusion ,Lamport's Algorithm, Ricart-Agrawala Algorithm; Basic concepts of Distributed deadlock detection, Distributed File system, Architecture, Design issues, SUN Network File system Basic concepts of Distributed shared memory, Basic concepts of Distrubuted Scheduling, Load balancing, Load sharing.

Distributed OS Implementation: Models, Naming, Process migration, Remote Procedure Calls.

Multiprocessor System: Motivation, Classification, Multiprocessor Interconnections, Types, Multiprocessor OS functions & requirements; Design & Implementation Issue; Introduction to parallel programming; Multiprocessor Synchronization.

Performance, Coprocessors, RISC & data flow: Introduction, Necessity, Measures, Techniques, Bottlenecks & Saturation, Feedback loops, Coprocessors, RISC.

Analytic Modeling: Introductions, Queuing Theory, Markov Process.

Security & Protection: Security-threats & goals, Penetration attempts, Security Policies & mechanisms, Authentication, Protections & access control Formal models of protection, Cryptography, worms & viruses.

Text/References:

1. M. Milenkoviac, "Operating Systems Concepts and Design", Tata McGraw Hill, 2/e, 1992.
2. H. M. Deitel, "Operating System", Prentice Hall, 3/e, 2003.
3. Mukesh Singhal and Niranjan G. Shivaratri, "Advanced Concepts in operating Systems", Tata McGraw Hill, 2001.
4. M. J. Bach, "Design of the Unix Operating System", Prentice-Hall of India, 1986.
5. Charles Crowley, "Operating System: A Design-oriented Approach", Irwin Publishing, 1996.

CSX- 334 NETWORK PROGRAMMING [3 0 0 3]

Introduction to Network Programming: OSI model, UNIX standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application.

The UNIX Model: Introduction, Basic Definitions, Input and Output, Signal, Process Control, Daemon Processes.

IPC: Introduction, File and record locking, Pipes, FIFOs streams and messages, Name spaces, system IPC, Message queues, Semaphores. Shared Memory.

Sockets: Address structures, value – result arguments, Byte ordering and manipulation function and related functions Elementary TCP sockets – Socket, connect, bind, listen, accept, concurrent servers. Close function and related function.

TCP client server: Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.

I/O Multiplexing and socket options: I/O Models, select function, Batch input, shutdown function, poll function, TCP Echo server, getsockopt and setsockopt functions. Socket states, Generic socket option IPV6 socket option ICMPV6 socket option IPV6 socket option and TCP socket options.

Elementary UDP sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP, determining outgoing interface with UDP.

Elementary name and Address conversions: DNS, get host by Name function, Resolver option, Function and IPV6 support, uname function, other networking information.

Remote Login: Terminal line disciplines, Pseudo-Terminals, Terminal modes, Control Terminals, rlogin Overview, RPC Transparency Issues.

Text/References:

1. UNIX Network Programming, Vol. I, Sockets API, 2nd Edition. - W.Richard Stevens, Pearson Edn. Asia.
2. UNIX Network Programming, 1st Edition, - W.Richard Stevens. PHI.
3. UNIX SYSTEMS PROGRAMMING USING C++ T CHAN, PHI.
4. UNIX for programmers and Users, 3RD Edition, GRAHAM GLASS, KING ABLES, Pearson Education.
5. Advanced UNIX programming, 2nd edition, M J Rochkind pearson education.

CSX-336 EMBEDDED SYSTEMS **[3 0 0 3]**

Introduction: Introduction to Embedded Systems, Classification of Embedded System, Concept of Embedded System Design, and Design challenges: Processor technology, IC technology, Design technology and Trade-offs.

Hardware and Software Co-Design in Embedded System: Buffers and latches, Reset circuit, Chip, Timers and counters and watch dog timers, Universal asynchronous receiver, transmitter (UART), Pulse width modulators, LCD controllers. Development of fixed ROM image, Code generation tools: Emulator, Simulator and Debugger.

Embedded software development environments: Challenges and issues in embedded software development, Device drivers, System calls and Programming languages: assembly languages, high level languages like C/C++, Source Code Engineering tool for Embedded C/C++. Introduction to Embedded Java.

Processor and memory Organization: Custom Single Purpose Processor Hardware, General-Purpose Processor: Introduction, Basic Architecture, Application Specific Instruction Set Processors (ASIPS), Microcontrollers and Digital Signal Processors. Memory writes ability, Storage performance, Tradeoff s, Memory hierarchy and cache.

Software Engineering in Embedded System: Software Engineering practice in the embedded Software development process. Software models used in designing, Unified Modeling language, Software maintenance.

Embedded Operating System: Operating system services, Embedded Operating system, Real Time Operating system, Interrupt latency and Response time, Interrupts Routines in RTOS, Introduction to VxWorks and Micro OS-II.

Text/References:

1. David E. Simon, “An Embedded Software Primer”, Pearson Education, 1999.
2. Raj Kamal, “Embedded Systems”, Tata McGraw-Hill, 2004.
3. Bruce Powel Douglass, “Real-Time UML: Developing Efficient Objects for Embedded Systems”, Addison Wesley, 2/e, 2004.
4. Muhammad Ali Mazidi and Janice Gillispie Mazidi, “The 8051 Micro controller & Embedded Systems”, Pearson Education, 2/e, 2005
5. Jonathan W. Valvano, “Embedded Microcomputer Systems: A real time interfacing”, Cengage Learning, 3/e, 2011.

CS- 338 AGILE SOFTWARE DEVELOPMENT [3 0 0]

Fundamentals of Agile:
The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools

Agile Scrum Framework:
Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management

Agile Testing:
The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester

Agile Software Design and Development:
Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control

Industry Trends:
Market scenario and adoption of Agile, Agile ALM, Roles in an Agile project, Agile applicability, Agile in Distributed teams, Business benefits, Challenges in Agile, Risks and Mitigation, Agile projects on Cloud, Balancing Agility with Discipline, Agile rapid development technologies

BOOKS RECOMMENDED:

1. Agile Software Development with Scrum By Ken Schawber, Mike Beedle, Pearson, 21 Mar 2008
2. Agile Software Development, Principles, Patterns and Practices By Robert C. Martin, Prentice Hall, 25 Oct 2002
3. Agile Testing: A Practical Guide for Testers and Agile Teams By Lisa Crispin, Janet Gregory, Addison Wesley, 30 Dec 2008
4. Agile Software Development: The Cooperative Game By Alistair Cockburn, Addison Wesley, 19 Oct 2006

CSX- 352 ADVANCED OPERATING SYSTEMS LABORATORY [0 0 2 1]

1. Semaphores Implementation using two producer and two consumer processes which are sharing a common stack.
2. Writing a Monitor Solution to implement the writer preference reader/writer problem.
3. Implementing weak reader preference problem using threads and monitors.
4. Write a Concurrent Time Server using Sockets and RPC Package.
5. Simulation of Lamport's timestamp algorithm & Vector timestamp algorithm.
6. Simulation of Ricart and Agarwala's algorithm.
7. Simulation of Raymond's algorithm for imposing a logical structure on a distributed computation.
8. Simulation of Bully Election Algorithm
9. Simulation of Invitation Election Algorithm.
10. Implementation of Group View Algorithm.
11. Implementation of Transis algorithm for reliable multicast.

12. Implementing Reconfiguration algorithm for a coordinator & participant in a group.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

CSX-354 NETWORK PROGRAMMING LAB [0 0 2 1]

Objectives:

To teach students various forms of IPC through UNIX and socket Programming

1. Write a program to implement fork, exec, wait and exit system call.
2. Implement the following forms of IPC.
   a) Pipes
   b) FIFO
3. Implement file transfer using Message Queue form of IPC
4. Write a programme to create an integer variable using shared memory concept and increment the variable simultaneously by two processes. Use semaphores to avoid race conditions
5. Design TCP iterative Client and server application to reverse the given input sentence
6. Design TCP client and server application to transfer file
7. Design a TCP concurrent server to convert a given text into upper case using multiplexing system call "select"
8. Design a TCP concurrent server to echo given set of sentences using poll functions
9. Design UDP Client and server application to reverse the given input sentence
10. Design UDP Client server to transfer a file
11. Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.
12. Design a RPC application to add and subtract a given pair of integers

* Students are advised to use **C/C++** programming language for above listed experiments.

This is only the suggested list of Practicals. Instructor may frame additional Practicals/ small project relevant to the course contents

CSX-356 EMBEDDED SYSTEMS LAB [0 0 2 1]

Instructor may frame Practicals relevant to the course contents

CS- 358 AGILE SOFTWARE DEVELOPMENT LABORATORY [0 0 2 1]

1. Understand a given business scenario and identify product backlog, user stories and sprint tasks.
2. Define user stories for a given feature.
3. Fill user stories, sprint schedule and sprint tasks in an Agile tool such as AgileFant.
4. Write unit tests aligned to xUnit framework for TDD.
5. Refactor a given design for next sprint requirement
6. Execute continuous integration using a tool such as Jenkins
7. Automate a set of given tests using Test automation tool.

Students should implement (and learn to use the tools to accomplish these tasks) the above during practical hours.

BOOKS RECOMMENDED:

1. Agile Software Development with Scrum By Ken Schawber, Mike Beedle, Pearson, 21 Mar 2008
2. Agile Software Development, Principles, Patterns and Practices By Robert C. Martin, Prentice Hall, 25 Oct 2002
3. Agile Testing: A Practical Guide for Testers and Agile Teams By Lisa Crispin, Janet Gregory, Addison Wesley, 30 Dec 2008
4. Agile Software Development: The Cooperative Game By Alistair Cockburn, Addison Wesley, 19 Oct 2006

7TH SEMESTER:

CSX-401 WIRELESS NETWORKS **[3 0 0 3]**

Networking Fundamentals: Analog vs. digital data, Data representation, Open Systems Interconnectivity (OSI), Elements of networks (hardware & software), Network topology, Network protocols, Performance Metric

Basis of Wireless Communications: What is mobile computing?, The driving forces to wireless, Advantages vs. disadvantages of wireless, Key elements of wireless networks or systems, Spectrum of mobile technologies – mobile phone 2/3/4 G; palm; pocket PC; tablet, Potential applications of mobile computing, Mobile challenges and limitations, Determinants of successful applications.

Overview of Mobile Technology: Types of wireless transmission, Basic components – filter, mixer, amplifier, antenna, Infrared light transmission, Radio frequency transmission – AM / FM / PM, Factors impact radio transmission, Bluetooth technology, IEEE 802.11 a/b/g technology, Comparison and selection of technology

Comparison of Mobile Technologies: Spread spectrum transmission - FHSS (Frequency Hopping Spread Spectrum) - DSSS (Direct Sequence Spread Spectrum), FDMA - Frequency Division Multiple Access, TDMA – Time Division Multiple Access, CDMA – Code Division Multiple Access, Comparison of wireless technology – AMPS, TDMA, GSM, GPRS, UMTS, etc.

Wireless Application Protocols (WAP) (Brief):, Bearsers – SMS, USSD, CSD, IS-136, CDMA, CDPD, PDC, etc.,WPD – wireless datagram protocol, WTLS – wireless transport layer security, WTP – wireless session protocol, WSP – Wireless session protocol, WAE - Wireless Application environment, Versions of WAP – WAP 1.1, WAP 1.2, WAP 2.0, WAP network architecture MAC- Media Access Control, LLC – Logical Link Control PHY – Physical Layer, IrDA standards and protocol, Bluetooth standards and protocol, 802.11x standards and protocol

Life Cycle of Wireless Network Design Life Cycle of Network Design – Planning, Analysis, Design, Implementation, Planning – wireless strategic planning, Planning – challenges, threats, and trends, Analysis – current network / systems status (strengths and weaknesses), Analysis – market gap analysis, Analysis – requirements analysis, Analysis – costs / benefits analysis, Implementation– project management, Implementation– change management

Peer to Peer (Ad Hoc) Network Design Type of wireless network, P-P network topology, IrDA network design and configuration, Bluetooth network design and configuration, 802.11x network design and configuration, Comparison of P-P network, Implementation related issues

Infrastructure Network Design Mobile wave propagation, Factors impact wave propagation, Propagation models, Site surveying techniques, Optimal network design (number and location of AP), 802.11x network design and configuration, Implementation related issues

Wireless Wide Area Network Design (Brief) Design of mesh network, Digital cellular telephony, Mobile gateway, Mobile bridge, Fixed wireless, VPN – virtual private network

Text/References:

1. K. Pahlavan and P. Krishnamurthy, “Principles of Wireless Networks”, Prentice Hall, 2002.
2. J. Schiller, “Mobile Communications”, Addison-Wesley, 2/e, 2003.
3. Jerry D. Gibson, “The Mobile Communications Handbook”, CRC Press, 3/e, 2012.
4. G. Held, “Data over Wireless Networks”, McGraw-Hill, 2001.
5. Roy Blake, “Wireless Communication Systems”, Cengage Learning, 2000.

CSX-403 COMPILER DESIGN **[3 0 0 3]**

Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

Lexical analysis: interface with input, parser and symbol table, token, lexeme and patterns. Difficulties in lexical analysis. Error reporting. Implementation. Regular definition, Transition diagrams, LEX.

Syntax analysis: CFGs, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.

Syntax directed definitions: inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.

Type checking: type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

Run time system: storage organization, activation tree, activation record, parameter passing, symbol table, dynamic storage allocation.

Intermediate code generation: intermediate representations, translation of declarations, assignments, control flow, Boolean expressions and procedure calls, Implementation issues.

Code generation and instruction selection: issues, basic blocks and flow graphs, register allocation, code generation, dag representation of programs, code generation from dags, peep hole optimization, code generator generators, specifications of machine.

Text/References:

1. V. Aho, R. Sethi, and J. D. Ullman, "Compilers: Principles, Techniques and Tools", Addison-Wesley, 1988.
2. Fischer and R. LeBlanc, "Crafting a Compiler", Benjamin Cummings, 1991.
3. C. Holub, "Compiler Design in C", Prentice-Hall, 1993.
4. Andrew W. Appel, "Modern Compiler Implementation in C: Basic Design", Cambridge Press, 2/e, 2001.
5. Christopher W. Fraser and David R. Hanson, "A Retargetable C Compiler: Design and Implementation", Addison-Wesley, 1995.

CSX-405 DATA MINING AND WAREHOUSING **[3 0 0 3]**

Introduction: Data Mining Concepts, Input, Instances, Attributes and Output, Knowledge Representation & Review of Graph Theory, Lattices, Probability & Statistics

Machine learning concepts and approaches: Supervised Learning Framework, concepts & hypothesis, Training & Learning, Boolean functions and formulae, Monomials, Disjunctive Normal Form & Conjunctive Normal Form, A learning algorithm for monomials

Data Preparation: Data Cleaning, Data Integration & Transformation, Data Reduction

Mining Association Rules: Associations, Maximal Frequent & Closed Frequent item sets, Covering Algorithms & Association Rules, Linear Models & Instance-Based Learning, Mining Association Rules from Transactional databases, Mining Association Rules from Relational databases & Warehouses, Correlation analysis & Constraint-based Association Mining.

Classification and Prediction: Issues regarding Classification & Prediction, Classification by Decision Tree induction, Bayesian classification, Classification by Back Propagation, k-Nearest Neighbor Classifiers, Genetic algorithms, Rough Set & Fuzzy Set approaches

Cluster Analysis: Types of data in Clustering Analysis, Categorization of Major Clustering methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based Clustering methods

Mining Complex Types of Data: Multidimensional analysis & Descriptive mining of Complex data objects, Mining Spatial Databases, Mining Multimedia Databases, Mining Time-series & Sequence data, Mining Text databases, Mining World -Wide Web

Data Mining Applications and Trends in Data Mining: Massive Datasets/Text mining, Agent-Based Mining

Text/References:

1. Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Morgan Kauffmann Publishers, 3/e, 2011.
2. Ian H. Witten and Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java implementations", Morgan Kaufmann Publishers, 1999.
3. Dorian Pyle, "Data Preparation for Data Mining", Morgan Kaufmann, 1999.
4. Korth, Silbertz and Sudarshan, "Database Concepts", McGraw Hill, 5/e, 2005.
5. Elmasri and Navathe, "Fundamentals Of Database Systems", Addision Wesley, 6/e, 2010.

CSX-421 Wireless Networks Lab. [0 0 2 1]

1. Design an 802.11 network of mesh topology, using set of suitable inputs check the performance parameters like: Battery Energy consumed, Bit error Rate, Busy, Signal to Noise ratio, Throughput, Utilization.
2. Design Wireless network using Carrier Sensing Multiple Access Technique, Check the performance parameters like: Channel Throughput, Signal to Noise Ratio etc.
3. Design a Project having two scenarios: (a) Star Topology Wireless Network using rapid configuration method. (b) Ring Topology Wireless network also using rapid configuration method, Compare the performance parameters like: End to End Delay for data, Traffic Received, Queue size etc.
4. Design a Star shaped Wireless network, and suggest a way to configure a Physical layer of selected nodes.
5. Design a Project having two scenarios: (a) Bus Topology Wireless Network (b) Ring Topology Wireless network, make use of the Web Reporting to compare the result of two different scenarios.
6. Design a Wireless model having four networks which are ten meters apart from each other, connected to each other wirelessly and are susceptible to delays etc.
7. Create a radio network and observe variations in the quality of received signal that results from radio noise at the receiving node in a dynamic network topology.
8. Designs a Star shaped Wireless topology and suggest a suitable way to import traffic.
9. Performance analysis of wireless mesh backhaul network with 802.11 a/b/g technologies using OPNET.
10. Performance analysis of wireless mesh backhaul network with 802.11 a/p technologies using OPNET.
11. Development of a new CDMA based MAC on top of 802.11p Physical layer

CSX-425 Data Mining and Data Warehousing Lab. [0 0 2 1]

Students are required to perform practical's in Oracle/MS SQL Server and *STATISTICA Data Miner*

- Building a Database Design using ER Modeling and Normalization Techniques
- Implementation of functions ,Procedures, Triggers and Cursors
- Load Data from heterogenous sources including text files into a predefined warehouse schema.
- Design a data mart for a bank to store the credit history of customers in a bank .Use this credit profiling to process future loan applications.
- Feature Selection and Variable Filtering (for very large data sets)
- Association Mining in large data sets
- Interactive Drill-Down, Roll up, Slice and Dice operations
- Generalized EM & *k*-Means Cluster Analysis
- Generalized Additive Models (GAM)
- General Classification and Regression Trees (GTrees)
- General CHAID (Chi-square Automatic Interaction Detection) Models
- Interactive Classification and Regression Trees
- Goodness of Fit Computations

CSX-400 MAJOR PROJECT (Phase-I)

This is project work (phase-I) to be done by the students in the seventh semester. The evaluation committee of the department shall evaluate the project for 2 credits assigned for the project. A report of the project work carried out during the semester shall be submitted at the end of the semester approved by the project guide and HOD.

DEPARTMENTAL ELECTIVE (DE)-III

CSX-431 NATURAL LANGUAGE PROCESSING [3 0 0 3]

Introduction to Natural Language Understanding: The study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems, Linguistic Background: An outline of English syntax. Introduction to semantics and knowledge representation, Some applications like machine translation, database interface.

Grammars and Parsing: Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Transition Network Grammars, Top-Down Chart Parsing. Feature Systems and Augmented Grammars: Basic Feature system for English, Morphological Analysis and the Lexicon, Parsing with Features, Augmented Transition Networks.

Grammars for Natural Language: Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language, Handling questions in Context-Free Grammars. Human preferences in Parsing, Encoding uncertainty, Deterministic Parser.

Computational morphology: Lemmatization, Part-of-Speech Tagging, Finite-State Analysis.

Ambiguity Resolution: Statistical Methods, Probabilistic Language Processing, Estimating Probabilities, Part-of-Speech tagging, Obtaining Lexical Probabilities, Probabilistic Context-Free Grammars, Best First Parsing. Semantics and Logical Form, Word senses and Ambiguity, Encoding Ambiguity in Logical Form.

Application of NLP: Intelligent Work Processors: Machine Translation; User Interfaces; Man-Machine Interfaces: Natural language Querying Tutoring and Authoring Systems. Speech Recognition Commercial use of NLP.

Text/References:

1. Akshar Bharti, Vineet Chaitanya and Rajeev Sangal, "NLP: A Paninian Perspective", Prentice Hall, New Delhi, 2004.
2. James Allen, "Natural Language Understanding", 2/e, Pearson Education, 1994.
3. D. Jurafsky and J. H. Martin, "Speech and Language Processing", Prentice Hall, 2/e, 2008.
4. L.M. Ivansca and S. C. Shapiro, "Natural Language Processing and Language Representation", AAAI Press, 2000.
5. T. Winograd, "Language as a Cognitive Process", Addison-Wesley, 1982.

CSX-433 CLOUD COMPUTING [3 0 0 3]

Overview of Distributed Computing: Trends of computing; Introduction to distributed computing; Next big thing: cloud computing.

Introduction to Cloud Computing: What's cloud computing; Properties & Characteristics; Service models; Deployment models.

Infrastructure as a Service (IaaS): Introduction to IaaS; Resource Virtualization (Server, Storage, Network); Case studies.

Platform as a Service (PaaS): Introduction to PaaS; Cloud platforms & Management (Computation, Storage); Case studies.

Software as a Service (SaaS): Introduction to SaaS; Web services; Web 2.0; Web OS; Case studies.

Cloud issues and challenges: Cloud provider Lock-in; Security

Text/References:

1) Barrie Sosinsk, "Cloud Computing Bible", Wiley, 2011.
2) John Rhoton, "Cloud Computing Explained: Implementation Handbook for Enterprises", Recursive Press, 2009.
3) George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud", O'Reilly, 2009.
4) Tim Mather, Subra Kumaraswamy and Shahed Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Media, 2009.

CSX-435 WEB TECHNOLOGIES **[3 0 0 3]**

Introduction to HTML: HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;

Introduction to JavaScript: Scripts, Objects in Java Script, Dynamic HTML with Java Script

XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's

Web Servers and Servlets: Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues,

Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

JSP Application Development: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations

Database Access : Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

Text/References:

1. Chris Bates, "Web Programming, building internet applications", John Wiley and Sons, 3/e, 2006.
2. Patrick Naughton and Herbert Schildt, "The complete Reference Java 2", McGraw Hill, 5/e, 2002.
3. Hans Bergsten, "Java Server Pages", O'Reilly, 3/e, 2003.
4. Harvey M. Deitel, Paul J. Deitel and T. R. Nieto, "Internet and World Wide Web", Pearson Education,1999.
5. Don Gosselin, Joel Sklar, Matt Slaybaugh, Ruth Guthrie and Louise Soe, "Web Warrier guide to web design technologies", Cengage Learning, 2003.

DEPARTMENTAL ELECTIVE (DE)-IV

CSX-441 SOFT COMPUTING **[3 0 0 3]**

Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen's self organizing networks - Hopfield network.

Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition -Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees -Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls -Simulated annealing – Evolutionary computation.

Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction -Rank method - Rank space method.

Softcomputing And Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

Text/References:

1. J. S. R. Jang, C. T. Sun and E. Mizutani, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998.
2. Laurene Fausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
3. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, 1995.
4. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998.
5. D. E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, 1989.

CSX-443 SYMBOLIC LOGIC AND LOGIC PROGRAMMING **[3 0 0 3]**

Prepositional logic: syntax and semantics: Validity and consequence. Normal forms. Representing world knowledge using prepositional logic.

First order logic: World knowledge representation and the need for quantifiers. Syntax, semantics validity consequence clause normal from.

Introduction to prolog: Syntax of prolog, Structured data representation. Execution model Introduction to Programming in Prolog, Illustrative examples. The connection between logic and logic programming interpreting logic programs in terms of Horn clauses Deduction from clause form formulas resolution for prepositional logic Ground resolution. Unification and first order resolution SLD resolution; the computation and search rules. SLD trees and interpretation of non-declarative features of Prolog.

Advanced prolog features: programming techniques: Structural Induction and Recursion, Extra Logical features: Cut and Negation Case studies.

Introduction to Fuzzy logic neural networks

Text/References:

1. Robert R. Stoll, "Set Theory and logic", Dover publishers, 1979.
2. W. F. Clocksin, and C. S. Mellish, "Programming in Prolog", Springer, 5/e, 2013.
3. D. Gries, "The Science of Programming", Narosa Publishers, 1985
4. R. O' Keefe, "The Craft of Prolog", The MIT Press, 1991.
5. J. W. Lloyd, "Foundation of Logic Programming", Springer, 2/e, 1987.

CSX-445 MULTIMEDIA DATABASES **[3 0 0 3]**

Introduction: An introduction to Object-oriented Databases; Multidimensional Data Structures: k-d Trees, Point Quadtrees, The MX-Quadtree, R-Trees, comparison of Different Data Structures

Image Databases: Raw Images, Compressed Image Representations, Image Processing: Segmentation, Similarity-Based Retrieval, Alternative Image DB Paradigms, Representing Image DBs with Relations, Representing Image DBs with R-Trees, Retrieving Images By Spatial Layout, Implementations

Text/Document Databases: Precision and Recall, Stop Lists, Word Stems, and Frequency Tables, Latent Semantic Indexing, TV-Trees, Other Retrieval Techniques

Video Databases: Organizing Content of a Single Video, Querying Content of Video Libraries, Video Segmentation, video Standards

Audio Databases: A General Model of Audio Data, Capturing Audio Content through Discrete Transformation, Indexing Audio Data

Multimedia Databases: Design and Architecture of a Multimedia Database, Organizing Multimedia Data Based on The Principle of Uniformity, Media Abstractions, Query Languages for Retrieving Multimedia Data, Indexing SMDSs with Enhanced Inverted Indices, Query Relaxation/Expansion

Creating Distributed Multimedia Presentations: Objects in Multimedia Presentations, Specifying Multimedia Documents with Temporal Constraints, Efficient Solution of Temporal Presentation Constraints, Spatial Constraints.

Spatial Concepts and Data Models: Models of spatial information, Design extending the ER model with spatial concepts, Extending the ER model pictograms, Object oriented data model with UML.

Spatial Query Languages: Extending the SQL for spatial data, Examples of queries that emphasis spatial data, Object relational schema examples querries.

Text/References:

1. V.S. Subrahmanian, "Principles of Multimedia Database Systems", Morgan Kauffman, 1998.
2. S. Shekhar and S. Chawla, "Spatial Databases", Pearson Education, 2003.
3. Lynne Dunckley, "Multimedia Databases: An object relational approach", Pearson Education, 2003.
4. B. Prabhakaram, "Multimedia Database Systems", Springer, 1997.
5. S. Maheshwari and R. Jain, "DBMS: Complete Practical Approach", Firewall Media, 2006.

OPEN ELECTIVES (OE) -I

CSX- 001 DATABASE MANAGEMENT SYSTEM **[3 0 0 3]**

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model:
ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

Data Base Design & Normalization:
Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

Crash Recovery: Failure classification, recovery concepts based on deferred update, recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Client/Server Databases: Client/Server concepts, approach, Client/Server environments, characterization of Client/Server computing, application partitioning, the two-layer, and the Three layer architecture, Client/Server communication, APIs in Client/Server computing, middleware technology, application developments, design concepts, Client application development tools, and database servers.

Integrity, Security and Repositories: Needs for database integrity, integrity constraints, non-procedural integrity constraints, integrity constraints specifications in SQL, introduction to database security mechanism, security specification in SQL, system catalogues

Case Studies:
Oracle: Database Design and Querying Tools; SQL Variations and Extensions; Storage and Indexing; Query Processing and Optimization; Concurrency Control and Recovery; System Architecture; Replication, Distribution and External Data; Database Administration Tools.
IBM DB2: Universal database; Database Design and Querying Tools; SQL Variations and Extensions Storage and Indexing; Query Processing and Optimization; Concurrency Control and Recovery; System Architecture; Replication, Distribution and External Data; Database Administration Tools.

Text/References:

1. C. J. Date, "An Introduction to Database System", Addision Wesley, 8/e, 2003.
2. Korth, Silbertz and Sudarshan, "Database Concepts", McGraw Hill, 5/e, 2005.
3. Elmasri and Navathe, "Fundamentals Of Database Systems", Addision Wesley, 6/e, 2010.
4. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication, 1990.
5. Rob and Coronel, "Database Systems", Cengage Learning, 7/e, 2006.

CSX- 003 EMBEDDED SYSTEMS **[3 0 0 3]**

Introduction: Introduction to Embedded Systems, Classification of Embedded System, Concept of Embedded System Design, and Design challenges: Processor technology, IC technology, Design technology and Trade-offs.

Hardware and Software Co-Design in Embedded System: Buffers and latches, Reset circuit, Chip, Timers and counters and watch dog timers, Universal asynchronous receiver, transmitter (UART), Pulse width modulators, LCD controllers. Development of fixed ROM image, Code generation tools: Emulator, Simulator and Debugger.

Embedded software development environments: Challenges and issues in embedded software development, Device drivers, System calls and Programming languages: assembly languages, high level languages like C/C++, Source Code Engineering tool for Embedded C/C++. Introduction to Embedded Java.

Processor and memory Organization: Custom Single Purpose Processor Hardware, General-Purpose Processor: Introduction, Basic Architecture, Application Specific Instruction Set Processors (ASIPS), Microcontrollers and Digital Signal Processors. Memory writes ability, Storage performance, Tradeoff s, Memory hierarchy and cache.

Software Engineering in Embedded System: Software Engineering practice in the embedded Software development process. Software models used in designing, Unified Modeling language, Software maintenance.

Embedded Operating System: Operating system services, Embedded Operating system, Real Time Operating system, Interrupt latency and Response time, Interrupts Routines in RTOS, Introduction to VxWorks and Micro OS-II.

Text/References:

1. David E. Simon, "An Embedded Software Primer", Pearson Education, 1999.
2. Raj Kamal, "Embedded Systems", Tata McGraw-Hill, 2004.
3. Bruce Powel Douglass, "Real-Time UML: Developing Efficient Objects for Embedded Systems", Addison Wesley, 2/e, 2004.
4. Muhammad Ali Mazidi and Janice Gillispie Mazidi, "The 8051 Micro controller & Embedded Systems", Pearson Education, 2/e, 2005
5. Jonathan W. Valvano, "Embedded Microcomputer Systems: A real time interfacing", Cengage Learning, 3/e, 2011.

CSX-005 BIOINFORMATICS **[3 0 0 3]**

Fundamentals of Bioinformatics and Information Technology: Introduction to bioinformatics, Experimental sources of biological data, publicly available databases, Operating systems - including Windows and UNIX, Networks - including the Intranets and the Internet

Analytical Science and Bioinformatics: High throughput sequencing, Experimental determination of protein structures, Gene expression monitoring, Proteomics, Metabolomics

Statistical Methods in Bioinformatics: Basic mathematics, Vectors and matrices, Multivariate statistics - particularly exploratory methods and pattern recognition

Bioinformatics Algorithms and Tools: Visualization of sequence data, Sequence alignment, Homology searching - including BLAST, Gene expression informatics, Introduction to gene finding

Applications and Commercial Aspects of Bioinformatics: Visualization of sequence data, Drug discovery, Genetic basis of disease, Personalized medicine and gene-based diagnostics, Legal, ethical and commercial ramifications of bioinformatics

Bioinformatics: The Business of Research: Research methodology (focusing on computer-based research), Case studies of areas of current bioinformatics research Routes to research funding (academic and commercial), Bioinformatics business models, Intellectual property rights

Software Engineering in Bioinformatics: Advanced programming using Java and BioJava, Advanced database work using SQL, Interfacings programs with databases. Data interoperability using XML

Principles of Programming and Databases using Java and SQL: Fundamental principles of programming, Object-oriented programming using Java, Introduction to databases using Oracle.

PERL programming: Data manipulation, File maintenance, Pipelining Packaging and interfacing system facilities

Text/References:

1. Jean-Michel Claverie and Cedric Notredame, "Bioinformatics for Dummies", John Wiley & Sons, 2003.
2. Bryan P. Bergeron, "Bioinformatics Computing", Prentice Hall, 2002.
3. Teresa Attwood and David Parry-Smith, "Introduction to Bioinformatics", Prentice Hall, 2001.
4. James Tisdall, "Beginning Perl for Bioinformatics", O'reilly, 2001.
5. Cynthia Gibas and Per Jambeck, "Developing Computer Skills", O'reilly, 2001.

8TH SEMESTER:

CSX-402: OBJECT-ORIENTED ANALYSIS AND DESIGN [3 0 0 3]

Introduction:Introduction to the Design Process Improvement Model ,Six-Level Improvement Process

UML Structural Modeling Techniques:Basic Building Blocks -- objects and classes,Structural Composition Techniques,Design Scaling Issues

UML Behavioral Modeling Techniques :Use Case Diagrams,Interaction Diagrams,Event State Diagrams,Action Matrices,Business Lifecycle Diagrams,Activity Diagrams,Collaboration Diagrams,Rule Specification Techniques,Behavioral Model-Based Reference Architecture for Component Specification

Design Standards :Architectural Patterns ,Design Patterns,Program Patterns,Behavioral Design Units,Component-Based Specification Techniques

DPIM - Level One:Requirements Analysis Techniques,Ad Hoc Approach to Design

DPIM - Levels Two, Three and Four:Design Methodology Deployment

Design Quality Control Properties and Analysis Techniques:Automatic Convertability,Traceability,Standardizability (Design Units/Reusable Patterns) Modularity,Changeability (Change Management),Scalability of Design,Reliability

DPIM - Levels Five and Six:Design Process Management and Optimization,Design Metric Models,Testing Maturity Model,Extended V-Model ,Testing Techniques

Text/References:

1. Grady Booch, J. Rambaugh and Ivar Jacobson, "The UML Users guide", Addison-Wesely, 2/e, 2005.
2. Edward Yourdon and Carl Argila, "Case Studies in Object-Oriented Analysis and Design", Prentice Hall PTR, 1996.
3. Ali Bahrami, "Object Oriented System Development", McGraw Hill, 1999.
4. J. Rambaugh and M. R. Blaha, "Object Oriented Modeling and Design", Prentice Hall, 2/e, 2004.
5. Andrew Haigh, "Object Oriented Analysis and Design", Tata McGrawHill, 2001.

CSX-404 SYSTEM SIMULATION AND MODELING [3 0 0 3]

Introduction to Simulation: System & System Environment, Components of a System, Discrete and Continuous Systems, Model of a System and Types of Models,. Discrete Event System Simulation, Advantages and Disadvantages of Simulation, Areas of Application

Techniques of Simulation: Monte Carlo Method, Types of System Simulations, Real Time Simulation, Stochastic Variables, Discrete Probability Functions

General Principles: Concepts in Discrete Event Simulation, Event Scheduling /Time Advance Algorithm, List Processing, Using Dynamic Allocation & Linked List

Simulation Software: History of Simulation Software, Selection of Simulation Software, Simulation in C++, GPSS, Simulations Packages, Trends in simulation Software.

Statistical Models in Simulation: Useful Statistical Models, Discrete Distribution s, Continuous Distributions, Poisson Process, Empirical Distributions

Queuing Models: Characteristics of Queuing systems, Queuing Notation, Long Run Measures of performance of Queuing Systems, Steady State Behavior of infinite Population Markovian Models, Steady State Behavior of finite Population Models, Networks of Queues

Random Number Generation: Properties of Random Numbers, Generation of Pseudo-Random Numbers, Techniques for Generating Random Numbers, Tests for Random Numbers, Inverse transform Techniques, Convolution Methods, and Acceptance –Rejection Techniques

Input Modeling: Data Collection, Identifying the Distribution with Data, Parameter Estimation, Chi – Square Test, Selecting Input Models with Data

Verification & Validation of simulation Modeling: Model Building, Verification & Validation, Verification of simulation Models, Calibration & Validation of Models**.**

Text/References:

1) G. Gordon, "System Simulation", Prentice Hall, 2/e, 1998.
2) Deo Narsingh, "System Simulation with Digital Computers", Prentice Hall, 1993.
3) K. S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Application", Wiley, 2001.
4) K. R. V. Subranranian and Sudaresan R. Kadayam, "System simulation: Introduction to GPSS", CBS, 1993.
5) W. Feller, "An introduction to Probability Theory and its Applications", Wiley Eastern Ltd, 1971.

CSX-406 ARTIFICIAL INTELLIGENCE [3 0 0 3]

Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas

Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic

Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation & Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Machine Learning: Concept of learning, Concept creation, learning automation, supervised and Unsupervised Learning, learning tasks & learning strategies, single layer & multiplayer Perceptions, Back propagation, learning by inductions, Competitive Learning, Hebbian Coincidence Learning, Attractor Networks Samuel's checkers algorithm. Hopfield nets, Adaptive resonance theory

Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Text/References:

1. Rich and K. Knight, "Artificial Intelligence", Tata McGraw Hill, 2/e, 1990.
2. George F. Luger, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving" Pearson Education, 6/e, 2008.
3. Stuart Russell and Peter Norvig, "Artificial Intelligence: A modern approach", Pearson Education, 3/e, 2009.

4. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, Prentice Hall, 1990.
5. Eugene Charniak and D. McDermott, “Introduction to Artificial Intelligence”, Addison-Wesley, 1985.
6. Nils J. Nilson, “Principles of Artificial Intelligence”, Morgan Kaufmann, 1982.

CSX-422: OBJECT-ORIENTED ANALYSIS AND DESIGN LABORATORY [0 0 2 1]

1. Understanding Problem Statements
2. Preparing Software Requirement Specification Document
3. Introduction to CASE tool: Rational Rose
4. Preliminary Use Case Diagrams
5. Detailed Use Case Diagrams
6. Class Diagrams
7. Object Diagrams
8. Activity Diagrams
9. Sequence Diagrams
10. Collaboration Diagrams
11. Deployment Diagrams
12. Practicing Analysis and Design Case Tools

Text/References:

1. Martin Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, Addison Wesley Publications, 5/e.
2. Henriette Baumann, Patrick Grassle and Philippe Baumann, “UML 2.0 in Action: A Project Based Tutorial”, Packet Publishing Limited, 2005.
3. Russell Miles and Kim Hamilton, “Learning UML 2.0”, Addison Wesley, 4/e, 2006.

Instructor may frame Practicals relevant to the course contents

CSX-424: SYSTEM SIMULATION AND MODELING LABORATORY [0 0 2 1]

Implementation of the followings Simulation problems in GPSS or any High Level Programming Language

1. Computer Generation of Random Numbers.
2. Testing Random Number Generators.
3. Monte-Carlo Simulation.
4. Simulation of Single Server Queuing System.
5. Simulation of Two-Server Queuing System.
6. Simulation of Inventory System.
7. Simulation of Telephone System.

DEPARTMENTAL ELECTIVE (DE)-V

CSX-432 MODELING AND SIMULATION OF NETWORKS [3 0 0 3]

Delay Models in Data Networks: Queuing Models, M/M/1, M/M/m, M/M/□ □ M/M/m/m and other Markov System, M/G/1 System, Networks of Transmission Lines, Time Reversibility, Networks of Queues.
Multi-access Communication: Slotted Multi-access and the Aloha System, Splitting Algorithms, Carrier Sensing, Multi-access Reservations, Packet Radio Networks.

Routing in Data Networks: Introduction, Network Algorithms and Shortest Path Routing, Broadcasting Routing Information: Coping with Link Failures, Flow models, Optimal Routing, and Topological Design, Characterization of Optimal Routing, Feasible Direction Methods for Optimal Routing, Projection Methods for Optimum Routing, Routing in the Codex Network.

Flow Control: Introduction, Window Flow Control, Rate Control Schemes, Overview of Flow Control Practice, Rate Adjustment Algorithms.

Text/References:

1. Dimitri Bertsekas and Robert Gallager, "Data Networks", Prentice Hall of India, 2/e, 1992.
2. William Stallings, "High-Speed Networks and Internets", Prentice Hall, 2/e, 2001.
3. J. Walrand and P. Varaya, "High Performance Communication Networks", Morgan Kaufman, 2/e, 1999.
4. Jean Walrand, Kallol Bagchi and George W. Zobrist, "Network performance modeling and simulation", Gordon and Breach Science Publishers, 1998.
5. Nader F. Mir, "Computer and Communication", Prentice Hall, 2006.

CSX-434 DISTRIBUTED SYSTEMS [3 0 03]

Characterization of Distributed Systems: Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges.

System Models: Architectural models, Fundamental Models

Theoretical Foundation for Distributed System: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport's & vectors logical clocks, Causal ordering of messages, global state, and termination detection.

Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non token based algorithms, performance metric for distributed mutual exclusion algorithms.

Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms.

Agreement Protocols: Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem, Atomic Commit in Distributed Database system.

Distributed Objects and Remote Invocation: Communication between distributed objects, Remote procedure call, Events and notifications, Java RMI case study.

Security: Overview of security techniques, Cryptographic algorithms, Digital signatures Cryptography pragmatics, Case studies: Needham Schroeder, Kerberos, SSL & Millicent.

Distributed File Systems: File service architecture, Sun Network File System, The Andrew File System, Recent advances.

Transactions and Concurrency Control: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control.

Distributed Transactions: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault -tolerant services, highly available services, Transactions with replicated data.

Distributed Algorithms: Introduction to communication protocols, Balanced sliding window protocol, Routing algorithms, Destination based routing, APP problem, Deadlock free Packet switching, Introduction to wave & traversal algorithms, Election algorithm.

CORBA Case Study: CORBA RMI, CORBA services.

Text/References:

1. Mukesh Singhal and Niranjan G. Shivaratri, "Advanced Concepts in operating Systems", Tata McGraw Hill, 2001.
2. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, "Distributed System: Concepts and Design", Pearson Education, 5/e, 2002.
3. Gerald Tel, "Introduction to Distributed Algorithms", Cambridge Iniversity Press, 2/e, 2000.
4. Nancy Lynch, "Distributed Algorithms", Morgan Kaufmann, 1996.
5. Andrew S. Tanenbaum, "Distributed Operating Systems", ACM Press, 1994.

CSX-436 DATABASE ADMINISTRATION **[3 0 0 3]**

Introduction to Database and SQL Server 2000: Client/Server Concept, Types of Databases, Relational Vs. Flat File Database. Background of SQL Server, Versions of SQL Server and Clients Supported by SQL Server. Installation & Configuring SQL Server: Installing SQL Server 2000, Unattended Installations, SQL Server Services. Configuring SQL Server Network Protocol Settings. Installing SQL Server Clients.

SQL Server Tools and Utilities: Managing SQL Server with Enterprise Manager, Query Analyser, SQL Server Groups. Tools Menu, Action Menu. Introduction to Transact – SQL(T-SQL)

Managing Database: Creating Database, Database File Placement(RAID 0, RAID 1 RAID 5), Creating Database using T-SQL and Enterprise Manager. Altering, Renaming, Dropping Database. Creating Objects in Database: Tables, Views, Constraints, Indexes.

Managing Security: Understanding Security Modes, Windows Authentication Modes, Mixed Mode, SQL Server Logins, Windows Logins, Fixed Server Logins, Creating Users, Database Roles, (Grant,Revoke ,Deny) N-Tier Security. Database Backups and Restore: Copying Database with Copy Database Wizard. SQL Database Backup Modes(Full, Differential, Transactional Log Backup). Backing Up of the Database. Restoring Database. DTS: Its meaning, DTS Packages. DTS Storage and Designer.

SQL Server Agent: Configuring Understanding Alerts, Jobs and Events. Creating Jobs: Multi Server Jobs, Creating, Editing and Deleting of Jobs. SQL Server and IIS. Understanding the Static Page and Dynamic Pages of the Internet. Internet Database Connector. Replication and Performance Optimization: Overview of Replication. Installing. Types of Replication : Merge Replication, Snapshot Replication, Transactional Replication. Using Windows System Monitor: Monitor with SQL Profiler and Query Analyser. Optimization Techniques: Queries and Stored Procedure, Proper Indexing, Locks and Defragmentation.

Text/References:

1. David C. Kreines, Brian Laskey, "Oracle Database Administration", Oreilly Media, 1999.

2. Craig S. Mullins, "Database Administration: The Complete Guide to Practices and Procedures", Addisn Wisley, 2002.
3. Claire Rajan, "Oracle 10g Database Administrator II: Backup/recovery & Network Administration", Course Technology, 2/e, 2006.
4. Sam R. Alapati, "Expert Oracle9*i* Database Administration", Apress, 2003.
5. Dan wood, "Beginining SQL Server 2005 Administration", Wrox Publication, 2006.

DEPARTMENTAL ELECTIVE (DE)-VI

CSX-442 IMAGE PROCESSING **[3 0 0 3]**

Fundamentals: Introduction, Origin, Areas of Image Processing, steps in Digital Image Processing, Components of Image Processing System, Image Sensing, Sampling and Quantization, Neighboring of Pixels, Mathematical and perceptual preliminaries, human visual system model, image signal representation, imaging system specification building image quality, role of computers, image date formats.

Image Enhancement and Restoration: Enhancement: Spatial Filtering, Introduction to Fourier Transformation.

Restoration: A model of the Image Degradation/ Restoration Process.

Hardware architecture for image processing: Color image signal representation, color system transformations, extension of processing techniques to color domain.
Wavelets: Wavelet functions, Wavelet transformations in one and two dimensions, fast wavelet transform.

Image Compression: Image compression models, Error free compression, Lossy compression. Image segmentation: Line detection, Edge Detection, Edge linking and Boundary Detection, and Region-based segmentation

Object Recognition: Pattern and pattern classes, Recognition based on Decision Theoretic Methods, Structural Methods.

Applications of Image processing: Picture data archival, machine vision, medical image processing.

Text/References:

1. William K. Pratt, "Digital Image Processing", Wiley-Interscience, 4/e, 2007.
2. Anil K. Jain, "Fundamentals of Digital Image Processing", Pearson Education, 1988
3. A. Rosenfield and A. C. Kak, "Picture Processing", Academic Press, 2/e, 1982
4. Rafael C. Gonzales and Richard E. Woods, "Digital Image Processing", Pearson Education, 3/e, 2007.
5. Kenneth R. Castleman, "Digital Image Processing", Prentice Hall, 1995.

CSX-444 Wireless Sensor Networks **[3 0 0 3]**

Introduction
Introduction to Wireless sensor networks, Single-sink single-hop WSN, Single-sink multi-hop WSN, Multi-sink multi-hop WSN, Advantages of ad-hoc/sensor networks, Node and Network Architectures, Wireless Sensor Device Architecture, Network Architectures, Main features of WSANs, Current and future research on WSANs

Applications of WSNs
Positioning and animals tracking, Entertainment, Logistics, Transportation, Industrial Control and Monitoring, Home Automation and Consumer Electronics, Security and Military Sensing, Asset Tracking and Supply Chain Management, Intelligent Agriculture and Environmental monitoring, Health Monitoring.

Technologies for WSNs
ZigBee technology, Ultrawide bandwidth technology, Bluetooth technology, Comparison among technologies

WSN Propagation Models
Introduction, Wireless Propagation Models: The Free Space Propagation Model, The Two-Ray Ground Model, The Log-Distance Path Model, Energy Dissipation Model, Error Models: The Independent Error Model, the Two-State Markov Error Model, Sensing Models: The Binary Sensing Model, the Probabilistic Sensing Model

Communication protocols for WSNs
MAC protocols: Scheduled protocols, LEACH protocol, Guo protocol, TRAMA protocol, Contention-based protocols, Zhong protocol, DMAC protocol, PAMAS protocol, SMAC protocol
Routing protocols: Issues in designing routing protocols, Classification of routing protocols, Flat routing, Flooding and gossiping, SPIN protocol, Directed diffusion protocol, Rumour routing, Gradient-based routing, Hierarchical routing, LEACH protocol, PEGASIS protocol, TEEN protocol, MECN protocol, SPAN protocol, Location-based routing protocols, GAF protocol, GEAR protocol, GeRaF protocol, Rugin protocol

Text/References:

1. Roberto Verdone, Davide Dardari, Gianluca Mazzini and Andrea Conti, "Wireless Sensor and Actuator Networks: Technologies, Analysis and Design", Academic Press, 2008.
2. Miguel A. Labrador and Pedro M. Wightman, "Topology Control in Wireless Sensor Networks-with a companion simulation tool for teaching and research", Springer Science, 2009.
3. Edgar H. Callaway, "Wireless Sensor Networks: Architectures and Protocols", CRC Press, 2004.
4. Xian-Yang Li, "Wireless Ad Hoc and Sensor Networks: Theory and Applications", Cambridge University Press 2008.
5. Feng Zhao and Leonidas J. Guibas, "Wireless Sensor Networks: An Information Processing Approach", Morgan Kaufmann Publishers, 2008.

CSX- 446 SOFTWARE PROJECT MANAGEMENT [3 0 0 3]

Conventional Software Management: The waterfall model, conventional software Management performance.

Evolution of Software Economics: Software Economics, pragmatic software cost estimation.

Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.

The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.

Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

Model based software architectures: A Management perspective and technical perspective.

Work Flows of the process: Software process workflows, Iteration workflows,

Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.

Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation: Automation Building blocks, The Project Environment.

Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.

Tailoring the Process: Process discriminants.

Standards: Introduction to standards - ISO 9002 and ISO 9003 - Quality system development, SO 9000 standard for software, Understanding ISO 900-3 clauses, SEI model – capability Maturity model - Five levels Bootstrap method, Implementing ISO 9000, Analysis the Quality system, Documenting & Auditing quality system, ISO 9000 registration process & Accreditation System, Total Quality Management

Future Software Project Management: Modern Project Profiles, Next generation Software economics, modern process transitions.

Case Study: The command Center Processing and Display system- Replacement (CCPDS-R)

Text/References:

1. Walker Royce, "Software Project Management", Pearson Education, 2005.
2. Ian Sommerville, "Software Engineering, Addison Wesley, 9/e, 2010.
3. M. Cottrell and B. Hughes, "Software Project Management", McGraw-Hill, 5/e, 2009.
4. D. J. Henry, "Software Project Management – A Real-World Guide to Success", Addison-Wesley, 2003.
5. Pankaj Jalote, "Software Project Management in practice", Pearson Education, 2002.

OPEN_ELECTIVE 11

CSX- 002 SOFTWARE ENGINEERING [3 0 0 3]

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

Design Engineering: Design process and Design quality, Design concepts, the design model.

Creating an architectural design: Software architecture, Data design, Architectural styles and patterns, Architectural Design.

Object-Oriented Design: Objects and object classes, An Object-Oriented design process, Design evolution.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

Metrics for Process and Products: Software Measurement, Metrics for software quality.

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

Quality Management: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modelling Language (UML)

Text/References:

1. K. K. Agarwal and Yogesh Singh, "Software Engineering", New Age International Publishers, 2008.
2. James F. Peters and Witold Pedrycz, "Software Engineering, an Engineering approach", John Wiely, 1999.
3. Waman S. Jawadekar, "Software Engineering principles and practice", McGraw-Hill Companies, 2004.
4. Roger S. Pressman, "Software Engineering, A practitioner's Approach", McGrawHill, 7/e, 2009.
5. Ian Sommerville, "Software Engineering", Addison Wesley, 7/e, 2004.

CSX-004 SOFT COMPUTING [3 0 0 3]

Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen's self organizing networks - Hopfield network.

Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition -Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees -Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls -Simulated annealing – Evolutionary computation.

Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction -Rank method - Rank space method.

Softcomputing And Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

Text/References:

1. J. S. R. Jang, C. T. Sun and E. Mizutani, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998.
2. Laurene Fausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
3. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, 1995.
4. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998.
5. D. E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, 1989.

CSX- 006 ARTIFICIAL INTELLIGENCE [3 0 0 3]

Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas

Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic

Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation & Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Machine Learning: Concept of learning, Concept creation, learning automation, supervised and Unsupervised Learning, learning tasks & learning strategies, single layer & multiplayer Perceptions, Back propagation, learning by inductions, Competitive Learning, Hebbian Coincidence Learning, Attractor Networks Samuel's checkers algorithm. Hopfield nets, Adaptive resonance theory

Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Text/References:

1. E. Rich and K. Knight, "Artificial Intelligence", Tata McGraw Hill, 1990.
2. George F. Luger, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Pearson Education, 6/e, 2008.
3. S. Russell & P. Norvig, "Artificial Intelligence 'a Modern Approach", Pearson Education, 3/e, 2009.
4. Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", PHI, 1990.
5. E. Charnaik and D. McDermott, "Introduction to Artificial Intelligence", Addison-Wesley, 1985.
6. Nils J. Nilson, "Principles of Artificial Intelligence", Morgan Kaufmann, 1982.