Computer Science

Unit 1. Discrete Structures

Sets, Relations, Functions, Pigeonhole Principle, Inclusion-Exclusion Principle. Equivalence and partial Orderings, Elementary Counting Techniques

Probability: Elements of probability, Bayes theorem.

Computability : Models of computation-Finite Automata, Pushdown Automata, Non-deterministic and NFA, DPDA and PDAs and Languages accepted by these structures. Grammars, Languages, Non-computability and Examples of non-computable problems.

Graph : Definition, walks, paths, trails, connected graphs, regular and bipartite graphs, cycles and circuits. Tree and rooted tree. Spanning trees, Eccentricity of a vertex radius and diameter of a graph. Central graphs. Centre(s) of a tree. Hamiltonian and Eulerian graphs. Planar graphs.

Groups : Finite fields and Error correcting/detecting codes.

Logic: Propositional logic, predicate logic, well formed formulae, Satisfiability and tautology.

Unit 2 Computer Organization and Architecture

Boolean algebra and Minimization of Boolean functions, Flip-flops-types, Race condition and comparison. Design of combinational and sequential circuits.

Representation of Integers : Octal, Hex, Decimal, and Binary. 2's complement and 1's complement arithmetic. Floating point representation.

Combinational Circuit Design, Sequential Circuit Design. Hardwired and Micro-programmed processor design, Instruction formats, Addressing modes, memory types and organizations, Interfacing peripheral devices, Interrupts.

Microprocessor architecture, Instruction set and Programming (8085, P-III/P-IV). Microprocessor applications.

Unit 3. Programming in C and C++

Programming language concepts, paradigms and models.

Programming in C: Elements of C-Tokens, identifiers, data types, operators in C. Control structures in C. Sequence, Selection and iterations (s). Structured data types in C-arrays, struct, union, String and pointers. I/O statements, User defined and built in functions, Parameter passing.

C++ Programming : Elements of C++- Tokens, identifiers, Variables and constants. Data types. Operators, Control statements, Functions parameter passing, Class and objects. Constructors and destructors. Overloading, Inheritance, Templates, Exception handling.

Object Oriented Programming Concepts : Class, Object, Instantiation, Inheritance, polymorphism and overloading, dynamic biding, reference semantics and their implementation.

Unit 4. Relational Database Design and SQL

Database, E-R diagram, Relational model, Relational Algebra, Relational Calculus, Relational design, Normalization, 1NF, 2NF, 3NF, BCNF and 4NF.

SQL: Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL) commands. Database objects like – Views, indexes, sequences, synonyms, data dictionary.

Unit 5. Data, File Structures

Data, Information, Definition of data structure. Arrays, stacks, Queues, Linked lists, Trees, Binary trees and traversal, Graphs, priority queues and heaps and assimilated algorithms.

File structures : Fields, Records and files. Sequential, Direct, index-sequential and relative files. Hashing, Inverted lists and multi-lists, B trees and B^+ trees.

Unit 6. Computer Networks

Network fundamentals : Local Area Networks (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN), Wireless Networks, Inter Networks.

Topologies, Networking Devices. The OSI model, TCP/IP model. Protocols for –(i) Data link layer, (ii) Network layer, and (iii) Transport layer, TCP/IP protocols, Networks security, Network administration.

Unit 7. System Software and Compilers

Assembly language fundamentals (8085 based assembly language programming). Assemblers -2-pass and single-pass. Macros and macroprocessors.

Loading, linking, relocation, program relocatability, linkage editing.

Text editors. Programming environments. Debuggers and program generators.

Compilation and Interpretation. Bootstrap compliers. Phases of compilation process. Lexical analysis. Lex package on Unix system.

Context free grammars. Parsing an parse trees. Representation of parse (derivation) trees as rightmost and leftmost derivations. Bottom up parsers –Shift-reduce, operator precedence, and LR. YACC package on Unix system.

Topdown parsers-left recursion and its removal. Recursive descent parser. Predictive parser. Intermediate codes-Quadruples, Triples, Intermediate code generation, Code generation, Code optimization.

Unit 8. Operating Systems (with case study of Unix)

Main features and functions of operating systems. Multiprogramming and Multiprocessing and multi tasking.

Memory Management : Virtual memory, paging, fragmentation.

Concurrent Processing : Mutual exclusion, Critical regions, Semaphores.

Scheduling : CPU scheduling, I/O scheduling, resource scheduling, Deadlock and scheduling algorithms. Banker's algorithm for deadlock handling.

The Unix System: File system, process management, bourne shell, shell variables, command line programming.

Filters and Commands : pr, head, tail, cut, paste, sort, uniq, tr, join, grap, egrap, fgrep, sed, awk, etc.

System Calls (like) : Creat, open, close read write, iseek, link, unlink, stat, fstat, umask, chmod, exec, fork, wait, system.

Unit 9. Software Engineering and Computer Graphics

System Development Life Cycle (SDLC) : Steps, Waterfall model, Prototypes, Spiral model.

Software Metrics : Software Project Management.

Software Design : System design, detailed design, function oriented design, object oriented design, user interface design, Design level metrics.

Coding and Testing: Testing Level Metrics. Software quality and reliability. Clean room approach, Software reengineering.

Concepts: Display systems, Storage Devices, Input devices, Output devices, 2D transformation, Windows and view port, Elements of 3D graphics, drawing, shading, clipping, drawing lines and shapes and algorithms, B-Spline curve, Bezier curves, Animation Graphics Standard, Fractals.

Unit 10. Algorithmics

Sorting and searching algorithms. Analysis of algorithms, Interpolation and Binary search, Asymptotic notations – big ohm, mega and theta, Average case analysis of simple programs like finding of a maximum of n elements. Recursion and its systematic removal. Quicksort-non-recursive implementation with minimal stack storage. Design of Algorithms (Divide and Conquer, Greedy method, Dynamic programming, Back tracking, Branch and Bound). Lower bound theory, non-deterministic algorithm,-non-deterministic programming constructs. NP-hard and NP-complete problems.