

## **Syllabus System Discipline**

### **Section 1: Engineering Mathematics**

Discrete Mathematics: Propositional and first order logic. Sets, relations, functions, partial orders and lattices. Monoids, Groups. Graphs: connectivity, matching, colouring. Combinatorics: counting, recurrence relations, generating functions.

Linear Algebra: Matrices, determinants, system of linear equations, eigenvalues and eigenvectors, LU decomposition.

Calculus: Limits, continuity and differentiability, Maxima and minima, Mean value theorem, Integration.

Probability and Statistics: Random variables, Uniform, normal, exponential, Poisson and binomial distributions. Mean, median, mode and standard deviation. Conditional probability and Bayes theorem.

Numerical Methods: Numerical solutions of linear and non-linear algebraic equations; integration by trapezoidal and Simpson's rules; single and multi-step methods for differential equations.

### **Section 2: Digital Logic**

Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).

### **Section 3: Computer Organization and Architecture**

Machine instructions and addressing modes. ALU, data-path and control unit. Instruction pipelining, pipeline hazards. Memory hierarchy: cache, main memory and secondary storage; I/O interface (interrupt and DMA mode).

### **Section 4: Programming and Data Structures**

Programming in C. Recursion. Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.

### **Section 5: Algorithms**

Searching, sorting, hashing. Asymptotic worst case time and space complexity. Algorithm design techniques: greedy, dynamic programming and divide-and-conquer. Graph traversals, minimum spanning trees, shortest paths.

### **Section 6: Theory of Computation**

Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and context-free languages, pumping lemma. Turing machines and undecidability.

**Section 7:** Compiler Design Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimization, Data flow analyses: constant propagation, liveness analysis, common sub expression elimination.

## **Section 8: Operating System**

System calls, processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU and I/O scheduling. Memory management and virtual memory. File systems.

## **Section 9: Databases**

ER-model. Relational model: relational algebra, tuple calculus, SQL. Integrity constraints, normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control

## **Section 10: Computer Networks**

Concept of layering: OSI and TCP/IP Protocol Stacks; Basics of packet, circuit and virtual circuit switching; Data link layer: framing, error detection, Medium Access Control, Ethernet bridging; Routing protocols: shortest path, flooding, distance vector and link state routing; Fragmentation and IP addressing, IPv4, CIDR notation, Basics of IP support protocols (ARP, DHCP, ICMP), Network Address Translation (NAT); Transport layer: flow control and congestion control, UDP, TCP, sockets; Application layer protocols: DNS, SMTP, HTTP, FTP, Email.

**\*Note: The above syllabus is indicative in nature and not exhaustive.**