

Part B - Syllabus

COMPUTER SCIENCE AND ENGINEERING (CSE)

Programming and Data Structures: C and C++ programming: variables, data types, operators, control flow, functions, arrays, pointers, and strings; object-oriented programming: classes, inheritance, polymorphism, and encapsulation; data structures: arrays, linked lists (singly, doubly, circular), stacks, queues, trees (binary, BST, AVL, heap), graphs; recursion and dynamic programming; hashing: hash functions and collision resolution; sorting algorithms: bubble, selection, insertion, merge, quick, heap; searching: linear and binary search; time and space complexity: Big-O notation.

Algorithms: Algorithm design paradigms: divide and conquer, greedy, dynamic programming, and backtracking; graph algorithms: BFS, DFS, Dijkstra's, Bellman-Ford, Floyd-Warshall, Kruskal's, and Prim's; string algorithms: KMP, Rabin-Karp, and suffix arrays; computational complexity: P, NP, NP-hard, and NP-complete; reductions and Cook's theorem; approximation algorithms; randomised algorithms; amortised analysis; online algorithms; parallel algorithms; number theory algorithms.

Theory of Computation: Formal languages and automata: DFA, NFA, and equivalence; regular expressions and languages; properties of regular languages: pumping lemma and closure properties; context-free grammars and pushdown automata; Chomsky normal form; CFL pumping lemma; Turing machines: definitions, variants, and computational power; decidability and undecidability; halting problem; reducibility; Godel's incompleteness theorem; complexity theory: time and space complexity classes; Cook-Levin theorem.

Computer Organisation and Architecture: Digital logic: gates, Boolean algebra, and combinational circuits; ALU design; control unit: hardwired and micro-programmed; memory hierarchy: registers, cache (mapping policies), main memory, and virtual memory; pipeline: data hazards, control hazards, and forwarding; instruction set architectures: RISC and CISC; interrupts and DMA; I/O organisation; cache coherence; memory interleaving; Flynn's taxonomy; multi-core architectures; GPU computing basics.

Operating Systems: Process management: process states, PCB, and context switching; CPU scheduling algorithms: FCFS, SJF, priority, round-robin, and multilevel queue; synchronisation: critical section, semaphores, monitors, and classical problems (producer-consumer, readers-writers, dining philosophers); deadlock: conditions, prevention, avoidance (banker's algorithm), detection, and recovery; memory management: paging, segmentation, and virtual memory (page replacement: FIFO, LRU, optimal); file systems: file operations, directory structure, and disk scheduling; I/O management; system security.

Database Management Systems: Relational model: relations, attributes, and keys; SQL: DDL, DML, joins, aggregation, subqueries, and views; relational algebra and calculus; database design: ER model, normalisation (1NF, 2NF, 3NF, BCNF, 4NF, 5NF); functional dependencies and Armstrong's axioms; transaction management: ACID properties; concurrency control: locking protocols, serializability, and timestamp ordering; recovery: logging (undo, redo, ARIES); indexing: B-tree, B+ tree, and hashing; query optimisation; distributed databases; NoSQL databases: document, key-value, column-family, and graph databases.

Computer Networks: OSI and TCP/IP models; physical layer: data encoding, multiplexing, and transmission media; data link layer: framing, error control (CRC, Hamming), flow control, and MAC protocols (CSMA/CD, CSMA/CA); network layer: IP addressing (IPv4, IPv6), subnetting, routing algorithms (distance vector, link state), and NAT; transport layer: TCP (connection management, congestion control, and flow control) and UDP; application layer: HTTP, FTP, SMTP, DNS, and

NRIU Ph.D. ENTRANCE EXAMINATION : 2026-26

SNMP; socket programming; network security: SSL/TLS, firewalls, IPsec, and VPN; wireless networks: IEEE 802.11; software-defined networking (SDN).

Software Engineering: Software development life cycle models: waterfall, spiral, iterative, and agile (Scrum, Kanban); requirements engineering: elicitation, analysis, and specification; software design: architectural patterns, design principles (SOLID), and design patterns (creational, structural, behavioural); software testing: unit, integration, system, and acceptance testing; black-box and white-box testing; code coverage; CI/CD pipelines; software metrics: LOC, cyclomatic complexity, function points; version control: Git; software quality assurance; software maintenance and refactoring.

Artificial Intelligence and Machine Learning: Search strategies: uninformed (BFS, DFS) and informed (A*, heuristics); knowledge representation: propositional and first-order logic; inference and resolution; Bayesian networks; supervised learning: linear and logistic regression, decision trees, SVM, and k-NN; unsupervised learning: k-means, hierarchical clustering, and PCA; neural networks: perceptron, MLP, backpropagation; deep learning: CNN, RNN, LSTM, and transformer architecture; reinforcement learning: Q-learning and policy gradient; natural language processing: tokenisation, embeddings, and language models; evaluation metrics.

Compiler Design: Phases of compilation; lexical analysis: regular expressions and DFA-based scanning; syntax analysis: context-free grammars, top-down parsing (recursive descent, LL(1)), and bottom-up parsing (LR(0), SLR, LALR, CLR); syntax-directed definitions; semantic analysis: type checking and attribute grammars; intermediate code generation: three-address code, DAG; code optimisation: local, global, and loop optimisations; code generation: register allocation and instruction selection; linkers and loaders.

Cloud Computing and Emerging Technologies: Cloud computing models: IaaS, PaaS, SaaS; virtualisation: hypervisors, containers (Docker, Kubernetes); distributed systems: consistency models, CAP theorem, MapReduce, Hadoop, and Spark; microservices architecture; serverless computing; IoT: sensors, protocols (MQTT, CoAP), and edge computing; blockchain: distributed ledger, consensus algorithms, smart contracts; cybersecurity: cryptography (symmetric and asymmetric), PKI, digital signatures; big data: volume, velocity, variety, and veracity; data warehousing and OLAP; quantum computing basics.