

Department of Electrical and Computer Science Engineering
PhD Admissions Spring Semester AY 2024-25 | Syllabus for written test
Discipline: Computer Science Engineering

Part-I [50% weight]

Research Methodology: Concepts and methods of engineering research, use of ICT and soft tools in research, research ethics, formulating the research problem, literature review, research design, collection-processing-analysis of data, reasoning, documentation-presentation-dissemination of research, intellectual property rights and scholarly publishing.

Part-II [50% weight]

- Basic Mathematics: linear algebra: vector and matrix properties and operations, solving systems of linear equations; probability and statistics, random variables, random processes
- Basic Programming Concepts: Iterative programming (for and while loop constructs), conditional executions, functions, pointers, recursions, file handling, procedural and object oriented programming concepts
- Data Structures: Linear data structures: arrays, stack, queue, linked lists; non-linear data structures: binary search tree, balanced binary search tree, heap tree, graphs; representation of data structures in computer memory; applications and complexity of operations on / using data structures
- Design and Analysis of Algorithms: Asymptotic notations, sorting and search- insertion sort, selection sort, merge sort, quicksort, binary search, design techniques: divide and conquer, greedy, dynamic programming, data structures: heaps, union of disjoint sets, search trees, algorithms on graphs: exploration, connectivity, shortest paths, directed acyclic graphs, spanning trees, Intractability: NP completeness, reductions.
- Computer networks: TCP/IP protocol stack and design of Internet, application layer: HTTP, FTP, DNS, P2P file sharing, transport layer: Issues related to process-to process communication and reliable data transfer, TCP and UDP operations; network layer: routing, addressing, QoS issues, IPv4 and IPv6 protocols; data link layer: wired and wireless local area networks and protocols
- Digital Logic Design: Boolean algebra, logic gates, design of combinational logic circuits - adder, subtractor, multiplier, comparator; sequential logic circuits - flip-flops, registers, counters Computer Organization: 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)
- 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

- Compiler Design: Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, Data flow analyses: constant propagation, liveness analysis, common subexpression elimination
- 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
- Operating Systems: Interprocess communication, deadlock, memory management, file system design, device/IO management