# Syllabus – PhD Entrance Examination

## Computer Science & Engineering

**Data Structures and Algorithms:** Asymptotic worst-case time and space complexity, Recursion, Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs. Divide and Conquer methods – searching and sorting, Greedy methods, Dynamic Programming, Backtracking, NP-hard and NP-complete problems.

**Operating System:** Processes, threads, Deadlocks, CPU scheduling, Memory management and virtual memory, File systems.

**Computer Networks and Network Security:** Concept of layering. LAN technologies (Ethernet). Flow and error control techniques, switching. IPv4/IPv6, routers and routing algorithms (distance vector, link state). TCP/UDP and sockets, congestion control. Application layer protocols (DNS, SMTP, POP, FTP, HTTP). Basics of Wi-Fi, authentication, basics of public key and private key cryptography, digital signatures and certificates, firewalls.

**Database Management System**: Relational Database Design, Relational model, Relational Algebra, Relational Calculus, Relational design, Normalization, Indexing and Hashing, Storage and File Structures, Transaction Management, Concurrency Control. SQL: Joins, SQL/PL-SQL. Data mining and its algorithms, OLAP, OLTP.

**Software Engineering:** Software life cycle models, Requirement Engineering, feasibility analysis, Domain specific modelling, Software Design, UML and design patterns, Software Coding, Testing, Testing strategies, Software reliability and Advanced testing techniques, Software Quality management, Software Metrics

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

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

**Computer Architecture and Organization:** Instruction cycle and addressing modes. ALU, data-path and control unit. Memory hierarchy, interrupt and DMA, parallel processing

**Theory of Computation:** Regular expressions and finite automata. Context-free grammars and pushdown automata. Regular and context-free languages, pumping lemma. Turing machines and undecidability.

**Compiler Design:** Lexical analysis, top-down parser, bottom-up parser, syntax-directed translation. 3 - address code, Runtime environments. Intermediate code generation.