Ph. D Written Test Format and Syllabus

Computer Science, Faculty of Science

Ph.D. Admission Test Format

The written test consists of two parts:

1. **Part A:** Research Methodology 25 questions.
2. **Part B**: Computer Science 25 questions.

Part A: Research Methodology Syllabus:

Research Fundamentals:

Meaning of research; objectives of research; characteristics of good research, Research problem: Identification, selection, and techniques for defining research problem, Research process, Research outcomes, Review of Literature, Hypothesis: Definition and Types

Types of Research:

Types of research, fundamental and applied research, qualitative and quantitative. Research Design: Types of research design – Exploratory, Descriptive, Casual Analytical

Sampling, Data Collection and analysis:

Types and sources of data: Primary and secondary, Methods of collecting data: questionnaire, interview, observation, case study, experiments etc., Sampling and sampling methods, characteristics of good sample, sampling techniques, Statistical Methods for Data Analysis: measures of central tendency and dispersion

Research Report:

Main body of report, abstract and keywords, Referencing styles and bibliography. Journal and author indexing

Ethics in Research:

Biasing: Definition and Types, Plagiarism -Definition and forms, IPR, copyright infringement, AI Generated Content

Part B: Computer Science Syllabus

Mathematical Foundations and Optimization:

Mathematical logic, propositional and predicate logic, propositional equivalences, normal forms, predicates, quantifiers, and rules of inference form the core logical foundations. Discrete mathematics including set operations, relations, functions, recurrence relations, posets, lattices, combinatorics, counting principles, pigeonhole principle, mathematical induction, and Boolean algebra. Graph theory basics, paths, circuits, trees, traversals, groups, subgroups, rings, and fields. Optimization techniques incorporating linear programming mathematical models, graphical solutions, simplex and dual simplex methods, transportation and assignment models, along with project scheduling methodologies like PERT and CPM diagram representations, critical path calculations, and resource leveling.

Hardware Architecture and Operating Systems:

Digital logic covering gates, map simplifications, combinational circuits (adders, subtractors, multiplexers), and sequential circuits (latches, flip-flops, registers, counters). Computer architecture encompassing data representation, register transfer micro-operations, central processing unit organization, instruction formats, addressing modes, pipelining, and vector processing. 8085/8086 microprocessor architecture and instruction sets. Operating systems fundamentals featuring process management, threads, multicore programming, CPU and I/O scheduling algorithms, concurrent processing emphasizing mutual exclusion and critical regions, deadlocks handling via Banker's algorithm, memory management covering virtual memory, paging, fragmentation, thrashing, mass-storage structure, RAID, file system implementations, and security protection.

Data Structures, Algorithms, and Computation Theory:

Core data structures including definition, implementation, and applications of arrays, sparse matrices, stacks, queues, dequeues, linked lists, trees (binary, AVL, B, B+ trees), graphs, hashing functions, collision resolution techniques, and standard template libraries. Algorithm analysis relies on time and space complexities, asymptotic notations (Big-Oh, Theta), running time factors, Master Theorem, and recursion. Design paradigms incorporating divide-and-conquer, dynamic programming, greedy algorithms, backtracking, branch and bound, searching (sequential, binary), sorting (bubble, selection, insertion, merge, quick, heap, radix), and graph algorithms like BFS, DFS, and topological sorting. Computation theory covering finite automata, context-free grammars, pushdown automata, Turing machines, and compiler construction phases.

Database Technologies and Advanced Data Systems:

Database system concepts involving architectures, data independence, data models, schemas, ER and EER modeling with cardinality constraints, relational algebra, SQL (DDL, DML, DCL queries), and PL/SQL programming with stored procedures. Relational database design requiring functional dependencies, normalization (1NF, 2NF, 3NF), eliminating anomalies, de-normalization, transaction management, concurrency control, database recovery, query processing, query optimization, indexing (B/B+ trees), and file organizations. Advanced data systems spanning object-oriented databases, data warehousing, OLAP, data mining techniques (association rules, classification, clustering), Big Data architectures (Map-Reduce and Hadoop), HDFS, and NoSQL product management.

Networking, Software Engineering, and Artificial Intelligence:

Computer networks covering LAN, MAN, WAN, wireless networks, OSI and TCP/IP models, data communication transmission media, multiplexing, routing algorithms, congestion control, and application layers like DNS, Email, and WWW. Software engineering focuses on SDLC models (Waterfall, Prototype, Spiral, Agile), requirements elicitation, SRS validation, software design principles (cohesion, coupling, UML), metrics, cost estimation (COCOMO), quality assurance, and verification/validation testing strategies. Artificial intelligence incorporating heuristic state-space searches, knowledge representation, expert systems, planning, natural language processing, fuzzy sets, genetic algorithms, artificial neural networks, supervised/unsupervised machine learning, and data science cleansing using R.