Birla Institute of Technology & Science, Pilani

Pilani | Goa| Hyderabad | Dubai

# PhD ADMISSION BROCHURE

## Computer Science and Information Systems

| Subject | Content | Reference Books |
|---|---|---|
| **Data Structures and Algorithms** | Introduction: Data Abstraction, Data Modeling, Data Representation, Abstract Data Types, Algorithm Analysis and Order Notation, Time and Space requirements, Recursion and Iteration<br>Linear Structures: Lists, Random vs. Sequential Access, Restricted Access Lists.<br>Dictionaries-Searching and Ranking: Sorting Algorithms, Searching, Hashing, Hash Tables, Bloom Filters, Non-Linear Data Structures: Binary Trees, Binary Search Trees (BST), B-Trees, Tree Traversals, Heaps and Tries, Application of Trees, Graph Traversals-Representation, Connectivity, Paths, Connected Components<br>Weighted Graphs- Modelling, Shortest Path, Minimal Spanning Tree. | Goodrich, Michael T., and Roberto Tamassia. Algorithm design: foundation, analysis and internet examples. John Wiley & Sons, 2006.<br>Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to algorithms 3rd Edition, MIT press, 2018. |
| **Operating Systems** | Introduction to OS: OS architecture and its components, Special purpose systems, System structures<br>Process: Process Concept, Process scheduling, operations on processes, Inter Process Communication (IPC)<br>Process scheduling algorithms, Process Synchronization Critical section problem, Semaphores, Monitors, Synchronization Hardware<br>Deadlock: Characterization, Deadlock detection, Prevention, Avoidance, Recovery from deadlocks<br>Memory management, Allocation, Concept of segmentation & H/W support in Intel Processors.<br>Concept of Paging & H/W support in Intel Processors, Virtual Memory management, Page replacement algorithms, Frame allocation and | Stallings, William. Operating systems: internals and design principles. Boston: Prentice Hall, 2012.<br>Silberschatz, Abraham, Peter Baer Galvin, and Greg Gagne. Operating system principles. John Wiley & Sons, 2006. |

| | | |
|---|---|---|
| | Thrashing<br>Secondary Storage Structures & Management, File Systems | |
| **Computer Organization & Architecture** | Introduction, MIPS Architecture & Instruction Set, Computer Arithmetic, Floating Point Arithmetic, Role of Performance, Data path Design, Control Hardware, Exceptions & Microprogramming, Memory Organization- Introduction, Cache Memory Organization, Cache Performance, I/O Organization, Pipelining – Design Issues, Data Hazards, Control Hazards, Static Branch Prediction, Dynamic Branch Prediction, Advanced Concepts in pipelining, Modern Processors | Patterson, David A & J L Hennenssy, Computer Organisation & Design, Elsevier, 4th Ed., 2009.<br>W. Stallings, Computer Organisation & Architecture, PHI, 9th ed., 2012 |
| **Database Systems** | Introduction to Database Systems, Data Modelling: ER Modelling, Relational Modelling: ER to Relational Model, Database Design through functional dependencies, Normalization: 1NF, 2NF, 3NF, BCNF, Multi-valued dependencies.<br>Query Languages: Relational Algebra, SQL<br>Data Storage Indexing: File Organizations, RAID, Indexing Structures (tree-based, hash-based)<br>Query Processing and Optimization: Cost-Based Optimization and Heuristic-Based Optimization.<br>Transaction Management: Serial Schedule & Serializability, Recoverability & Cascadeless Schedules<br>Concurrency Control and Crash Recovery: Locking, Time-Stamping, Log-Based, Shadow Paging. | Ramakrishna R. & Gehrke J, Database Management Systems, 3e, Mc-Graw Hill.<br>Hector G Molina, Jeffrey D.Ullman and Jennifer Widom, Database Systems – The Complete Book, Pearson Education, 2e. |
| **Software Engineering** | Introduction: Evolving role of S/W, Software Myths, different development philosophies: sequential versus iterative, overview of various SDLC models/methodologies. Software Lifecycle Models: Build and Fix Model, Waterfall Model, Increment Process Model, Rapid Application Development (RAD) Model, Evolutionary Process Models, Unified Process, Selection of a Life Cycle Model<br>Requirement Specifications: Algebraic Specifications, Functional and non-functional requirements, SRS, requirements engineering processes, requirements elicitation and analysis, | Sommerville I, Software Engineering, Pearson Education, 10th Edition, 2017.<br>Pressman, R.S., Software Engineering: A Practitioner's Approach, 7th (Alternate) Edition, McGraw Hill International Edition, 2010 |

| | | |
|---|---|---|
| | requirements validation, management.<br><br>Software Design: Modularity, high-level and detailed design, layered design, cohesion and coupling, function-oriented design, SA/SD (structured analysis/structured design), data flow diagrams (DFDs), constructing DFDs, structure chart, object-oriented analysis and design (OOAD), UML Concept, (rational) unified process, patterns.<br><br>Software Project Planning: Size Estimation: Lines of Code (LOC), Function count, Cost estimation, Models: Constructive Cost Model (COCOMO), COCOMO II, Putnam resource allocation model, Halstead's software science, Software risk management.<br><br>Implementation and Testing: Rationale between requirements and testing, verification versus validation, black box and white box testing techniques.<br><br>Software Testing Methodologies: Functional Testing: Boundary value analysis, equivalence class testing, Cause Effect Graphing Technique and other topics based on students' interest. | |