

B. Tech. in SOFTWARE ENGINEERING

Syllabus of Paper – 1

DATA STRUCTURES

Introduction: Introduction to Algorithmic, Complexity- Time-Space Trade off. Introduction to abstract data types, design, implementation and applications. Introduction to List data structure. Arrays and Strings: Representation of Arrays in Memory: one dimensional, Two dimensional and Multidimensional, Accessing of elements of array, performing operations like Insertion, Deletion and Searching. Sorting elements of arrays. Strings and String Operations. Stacks and Queues: Introduction to data structures like Stacks and Queues. Operations on Stacks and Queues, Array representation of Stacks , Applications of Stacks : recursion, Polish expression and their compilation conversion of infix expression to prefix and postfix expression, Operations of Queues, Representations of Queues Applications of Queues, Priority queues. Linked Lists: Singly linked lists, Representation of linked list, Operations of Linked list such as Traversing, Insertion and Deletion, Searching, Applications of Linked List. Concepts of Circular linked list and Doubly linked list and their Applications. Stacks and Queues as linked list. Trees: Basic Terminology, Binary Trees and their representation, binary search trees, various operations on Binary search trees like traversing, searching, Insertion and Deletion , Applications of Binary search Trees , Complete Binary trees, Extended binary trees. General trees, AVL trees, Threaded trees, B- trees. Searching and Sorting: Linear Search, Binary search, Interpolation Search, Insertion Sort, Quick sort, Merge sort, Heap sort, sorting on different keys, External sorting. Graphs: Terminology and Representations, Graphs & Multi-graphs, Directed Graphs, Representation of graphs and their Transversal, Spanning trees, shortest path and Transitive Closure, Activity Networks, Topological Sort and Critical Paths. File Structure: File Organization, Indexing & Hashing, Hash Functions, Collision Resolution Techniques.

OBJECT ORIENTED PROGRAMMING

Object oriented paradigm & C++ at a glance: Evolution of programming paradigm, structured versus object-oriented development, elements of object-oriented programming, Objects, classes, methods, popular OOP languages, software reuse. Classes and objects: Introduction, Class revisited, constant objects and constructor, static data members with constructors and destructors, constructor overloading, nested classes, objects as arguments, returning objects , friend functions and friend classes, constant parameters and member functions, static data and member functions. Dynamic objects: Introduction, pointers to objects, array of objects, pointers to object members, this pointer, self-referential classes; Operator overloading and Inheritance: overloading of new and delete operators, conversion between objects and basic types, conversion between objects of different

classes, overloading with friend functions, abstract classes, inheritance types, virtual base classes, virtual functions, pointer to derived class objects, and base class objects, pure virtual functions, virtual destructors. Generic programming with templates: Introduction, function templates, overloaded function templates, class templates, inheritance of class template, class template containership, class template with overloaded operators. Introduction to byte code, security and portability, Data Types, variables, operators, arrays, type conversion and casting, type promotion, Control statements, standard input-output, Designing Classes, constructors, methods, access specifiers : public, private, protected, inheritance, packages and interfaces, Math, String, Vectors, and Array List classes, polymorphism: function and operator overloading, function overriding, abstract classes. Exception Handling: exception types, nested try-catch, throw, throws and finally statements, Multithread Programming: thread creation, synchronization and priorities. Input-output and file operations: Java.io, stream classes, Byte streams, character streams, serialization. Networking concepts: Client server and socket programming, TCP/IP client and server sockets. Applets and Java Swing: Applet design, AWT packages, Applet event handling, parameters to applets, AWT controls, layout manager, Frames, container classes, Introduction to Java Beans, Swing and Servlets.

WEB TECHNOLOGY

Inter-Networking: Internet, Growth of Internet, Owners of the Internet, Anatomy of Internet, APRANET and Internet history of the World Web, Basic Internet Terminology, Net etiquette. Working of Internet: Packet switching technology, Internet Protocols: TCP/IP, Router. Internet Addressing Scheme: Machine Addressing (IP address), E-mail Address, Resource Addresses. Internet Applications: E-mail, file transfer (FTP), telnet, usenet, Internet chat, Web. Evolution of Web: Web 1.0: Hypertext & linking documents, HTTP, Client-Server, peer-to-peer; Web Browser (Lynx, Mosaic, Netscape, Internet Explorer, Firefox, and Safari, the mobile web); Impact: Opportunities & Challenges. Web 2.0: From 1.0 to 2.0; Framework; Technologies: Client-side & server-side; Web 2.0 development technologies; Examples: social networking sites, blogs, wikis, video sharing sites, hosted services (web services, location-based services), web applications, mashups & folksonomies; Practical Usage. Web 3.0: From 2.0 to 3.0; Semantic Web: What, How, Why; From Web 3.0 to Web 4.0. Web Development: Phases; Web Page, Website, and Web Application: Example, Technology Framework for development. Client-side technology: HTML (HTML 5). Client-side scripting: JavaScript. Server-side technology: PHP. Server-side scripting: Server-side JavaScript. Web application development frameworks: Django & Ruby on Rails. Web Database: Database Connectivity: JDBC, ODBC; Database-to-web connectivity. Web Search and Mining: Web IR System: Search Engines, Web Crawling, Search Engine Optimization, Web Analytics, Web Mining Taxonomy; Web Mining Framework; Social Web Mining. Text Mining: Opinion Mining, Recommendation System, Topic Detection and Tracking.

SOFTWARE ENGINEERING

Introduction to software Engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement, monitoring and control. Software development life-cycle, Water fall model, prototyping model, Incremental model, Iterative enhancement Model, Spiral model. Software Requirement Specification: Requirements Elicitation Techniques, Requirements analysis, Models for Requirements analysis, requirements specification, requirements validation. System Design: Design Principles: Problem partitioning, abstraction. Top down and bottom up – design, structured approach. Functional versus object oriented approach of design, design specification, Cohesiveness and Coupling. Overview of SA/SD Methodology, structured analysis, data flow diagrams, extending DFD to structure chart. Software project Management: Project planning and Project scheduling. Software Metrics: Size Metrics like LOC, Token Count, Function Count. Cost estimation using models like COCOMO. Risk management activities. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, reliability models, Software quality, ISO 9000 certification for software industry, SEI capability maturity model. Testing: Verification and validation, code inspection, test plan, test case specification. Level of testing: Unit, Integration Testing, Top down and bottom up integration testing, Alpha and Beta testing, System testing and debugging. functional testing, structural testing, Software testing strategies. Software Maintenance: Structured Vs unstructured maintenance, Maintenance Models, Configuration Management, Reverse Engineering, Software Re-engineering.

COMPUTER ORGANIZATION AND ARCHITECTURE

Digital computer generation, computer types and classifications, functional units and their interconnections, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer. Register Transfer Language: Data movement around registers. Data movement from/to memory, arithmetic and logic micro operations. Concept of bus and timing in register transfer. Control Unit: Instruction types, formats, instruction cycles and sub-cycles (fetch and execute etc.), micro-operations, execution of a complete instruction. Hardwired and microprogrammed control: microprogrammed sequencing, wide branch addressing, and micro-instruction with next address field, pre-fetching microinstructions, concept of horizontal and vertical microprogramming. Central Processing Unit: Addition and subtraction of signed numbers look ahead carry adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Processor organization, general register organization, stack organization and addressing modes. Input/Output organization: Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access. I/O channels and processors. Serial Communication: Synchronous & asynchronous communication, standard communication interfaces. Memory: Basic concept and hierarchy, Main memory, Auxiliary memory, Associative memory,

Cache memories: concept and design issues, associative mapping, direct mapping, set-associative mapping, cache writing and initialization.

DATABASE MANAGEMENT SYSTEMS

Database system concepts and its architecture, Data models schema and instances, Data independence and database language and interface, Data definition languages, DML. Overall database structure. Data modeling using Entity Relationship Model: E.R. model concept, notation for ER diagrams mapping constraints, Keys, Concept of super key, candidate key, primary key generalizations, Aggregation, reducing ER diagrams to tables, extended ER model. Relational Data Model and Language: Relational data model concepts, integrity constraints, Keys domain constraints, referential integrity, assertions, triggers, foreign key relational algebra, relational calculus, domain and tuple calculus, SQL data definition queries and updates in SQL. Data Base Design: Functional dependencies, normal forms, 1NF, 2NF, 3NF and BCNF, multi-valued dependencies fourth normal form, join dependencies and fifth normal form. Inclusion dependencies, lossless join decompositions, normalization using FD, MVD and JDs, alternatives approaches to database design. File Organization, Indexing and Hashing Overview of file Organization techniques, Indexing and Hashing- Basic concepts, Static Hashing, Dynamic Hashing, Ordered indices, Multi-level indexes, B-Tree index files, B+- Tree index files, Buffer management; Transaction processing concepts: Transaction processing system, schedule and recoverability, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recovery from transaction failures, deadlock handling. Concurrency Control Techniques: Locking Techniques for concurrency control, time stamping protocols for concurrency control, concurrency control in distributed systems. multiple granularities and multi-version schemes.

DISCRETE STRUCTURES

Formal Logic: Statement, Symbolic Representation and Tautologies, Quantifiers, Predicate and validity, Normal form, Propositional Logic, Predicate Logic, Logic Programming and Proof of correctors. Proof, Relation and Analysis of Algorithm: Technique for theorem proving : Direct Proof, Proof by Contra position, proof by exhausting cases and proof by contradiction, Principle of mathematical induction, principle of complete induction, recursive definition, solution methods for linear, first-order recurrence relations with constant coefficients, analysis of algorithms involving recurrence relations-recursive selection sort, binary search, quick sort, solution method for a divide-and-conquer recurrence relation. Sets and Combinations: Sets, Subsets, powersets, binary and unary operations on a set, set operations/set identities, fundamental counting principles, principle of inclusion, exclusion and pigeonhole, permutation and combination, pascal's triangles, binomial theorem, representation of discrete structures. Relation/function and matrices: Rotation, properties of binary rotation, operation on binary rotation, closures, partial ordering, equivalence relation, properties of function, composition of function, inverse,

binary and n-ary operations, characteristics for, permutation function, composition of cycles, Boolean matrices, Boolean matrices multiplication. Lattices & Boolean Algebra; Lattices: definition, sublattices, direct product, homomorphism Boolean algebra: Definition, properties, isomorphic structures (in particular, structures with binary operations) subalgebra, direct product and homo-morphism, Boolean function, Boolean expression, representation & minimization of Boolean function. Graph Theory: Terminology, isomorphic graphs, Euler's formula (Proof) four color problem and the chromatic number of a graph, five color theorem. Trees terminology, directed graphs, Computer representation of graphs, Warshall's algorithms, Decision Trees, Euler path & Hamiltonian circuits, Shortest path & minimal spanning trees, Depth-first and breadth first searches, analysis of research algorithm, trees associated with DFS & BFS Connected components, in order, preorder & post order trees traversal algorithms.

Syllabus of Paper - 2

OBJECT ORIENTED SOFTWARE ENGINEERING

Object Oriented system concepts and Principles, Object Oriented system development, Component reuse, The common process framework for Object Oriented processes, System Development and Methodologies, object oriented software estimation. System development: System as model building, model architecture, The importance of modeling, principle of modeling, object oriented modeling, Introduction to Object-oriented Methodologies such as Unified Modeling Language, Overview of UML, conceptual model of UML, architecture, software development lifecycle using Rational Unified Process. Object Oriented Analysis: requirement model, analysis model, Object oriented analysis using methods of Rumbaugh. Software Design: Software design Models, Object oriented methodologies of Booch, design model, System development using various UML Diagrams. UML Methodology: Detailed study of various UML Diagrams, System Analysis using UML Diagrams. Object Oriented Testing and metrics: Path Testing, State based testing, Class Testing, object oriented metrics. Applications & Tools: A complete case study of Software development using above Methodologies, Concepts of Computer-Aided Software Engineering and knowledge about current CASE tools use in the industry.

ALGORITHM DESIGN AND ANALYSIS

Concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations. Growth of Functions, Master's Theorem. Searching and Sorting: Structure of divide-and-conquer algorithms; examples: binary search, quick sort, Stassen Multiplication; merge sort, heap sort and Analysis of divide and conquer run time recurrence relations. Greedy Method: Overview of the greedy paradigm examples of exact optimization solution: minimum cost spanning tree, approximate solutions: Knapsack problem, Kruskal's algorithm and Prim's algorithm for finding Minimum cost Spanning Trees, Dijkstra's and Bellman Ford Algorithm for finding Single source shortest paths, Huffman coding, Activity Selection Problem. Dynamic programming: Principles of dynamic programming. Applications: Rod cutting problem, Floyd-Warshall algorithm for all pair shortest paths. Matrix multiplication, Travelling salesman Problem, Longest Common sequence. Back tracking: Overview, 8-queen problem, and Knapsack problem, Traveling Salesman problem; Branch and bound: LC searching Bounding, FIFO branch and bound, LC branch and bound application: 0/1 Knapsack problem. Computational Complexity: Complexity measures, Polynomial Vs non-polynomial time complexity; NP-hard and NP-complete classes, examples: Circuit Satisfiability, Vertex cover, Subset Sum problem, Randomized Algorithms, String Matching, NP-Hard and NP-Completeness, Approximation Algorithms, Sorting Network, Matrix Operations, Polynomials and FFT, Number Theoretic Algorithms.

SOFTWARE TESTING

Introductory concepts: Verification & Validation Terminologies like Goals, Role, Objectives, Limitations, Approaches & Applicability. Software Testing: Testing Process, Limitations of Testing, Testing activities. Levels of Testing: Unit Testing, Integration Testing, System Testing, Debugging, Domain Testing, Regression Testing, Stress Testing, Slice based testing. Verification Testing: Verification Methods, SRS Verification, Software Design Document Verification, Code Reviews, User Documentation Verification, Software Project Audits. Functional Testing techniques: Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique. Structural Testing: Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing. Object Oriented Testing: Class Testing, GUI Testing. Testing Activities: Unit Testing, Levels of Testing, Integration Testing, System Testing, Debugging Software Testing Tools Taxonomy: Methodology to evaluate automated testing. Using tools: Load Runner, Win runner and Rational Testing Tools, Java Testing Tools, JMetra, JUNIT Cactus and other recent tools. Advanced Topics on Testing: Prioritizing the Test-cases, Testing Web Applications, Automated Test Data Generation.

OPERATING SYSTEM

Operating system and function, Evolution of operating system, Batch, Interactive, Time Sharing and Real Time System, System protection. Operating System Structure: System Components, System structure, Operating System Services. Concurrent Processes: Process concept, Principle of Concurrency, Producer Consumer Problem, Critical Section problem, Semaphores, Classical problems in Concurrency, Inter Process Communication, Process Generation, Process Scheduling. CPU Scheduling: Scheduling Concept, Performance Criteria Scheduling Algorithm, Evolution, Multiprocessor Scheduling. Deadlock: System Model, Deadlock Characterization, Prevention, Avoidance and Detection, Recovery from deadlock combined approach. Memory Management: Base machine, Resident monitor, Multiprogramming with fixed partition, Multiprogramming with variable partition, Multiple base register, Paging, Segmentation, Virtual memory concept, Demand paging, Performance, Paged replaced algorithm, Allocation of frames, Thrashing, Cache memory, Organization, Impact on performance. I/O Management & Disk Scheduling: I/O devices and organization of I/O function, I/O Buffering, DISK I/O, Operating System Design Issues. File System: File Concept, File Organization and Access Mechanism, File Directories, File Sharing, Implementation Issues.

COMPILER DESIGN

Definition, Phases and Passes, FSM & RE's and their application to Lexical Analysis, Implementation of Lexical Analyzers, Lexical- Analyzer Generator, Lex – Compiler. Syntax Analysis: Formal Grammar and their application to Syntax Analysis, BNF Notation,. The Syntactic specification of Languages: CFG, Derivation and Parse Trees, Shift Reduce Parsing, Operator precedence parsing, top down Parsing, Predictive Parsers. LR Parsers, the

canonical collection of LR(0) items, constructing SLR Parsing Tables, Constructing canonical LR Parsing tables and LALR parsing tables, An Automatic Parser Generator, YACC. Syntax Directed Translation: Syntax directed Translation Schemes, Implementation of Syntax directed translators, Intermediate Code, Postfix notation, Parse Trees and Syntax Trees, Three address Code, Quadruple & Triples, Translation of Assignment Statements, Boolean expressions, Control Statements, Array references in Arithmetic expressions, Procedure Calls, Declarations and Case statements Translations. Symbol Tables: Data Structure for Symbol Tables, representing scope information. Run Time Administration: Implementation of simple Stack allocation scheme, storage allocation in block structured language. Error detection and Recovery: Lexical phase errors, syntax phase errors, semantic errors and Error recovery techniques. Code Optimization: Loop optimization, the DAG representation of basic blocks, value numbers and Algebraic Laws, Global Data – Flow Analysis and Code generation.

SOFTWARE PROJECT MANAGEMENT

Project Management concepts, Process Framework, Project Planning Software Life Cycle Models, Artifacts of the Project Management Process. Cost and Scheduling Estimation Models: Various Levels of COCOMO for Cost, Effort, Schedule and Productivity Estimation. Approaches to Effort, Cost Estimation, and Schedule Estimation factors through COCOMO II, Putnam Estimation Model, Algorithmic models. Project Management Techniques: Project Organizations and Responsibilities, Establishing Project Environment, Risk Management Process, Project Tracking and Control Defect Tracking Concepts such as Process monitoring and audit, Reviews, Inspections and Walkthroughs. Project Closure: Project Closure Analysis, Role of Closure Analysis in a project, Performing Closure Analysis, Closure Analysis Report. Software Project Management Renaissance: Conventional Software Management, Evolution of Software Economics, Improving Software Economics, The old way and the new way.

COMPUTER NETWORKS

Concepts: Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design - Delay Analysis, Physical Layer Transmission Media, Switching methods, ISDN. Medium Access sub layer: Medium Access sub layer - Channel Allocations, LAN protocols -ALOHA protocols - Overview of IEEE standards - FDDI. Data Link Layer - Elementary Data Link Protocols, Sliding Window protocols, Error Handling. Network Layer: Network Layer - Point - to - Point Networks, routing, Congestion control, Internetworking -TCP / IP, IP packet, IP address, IPv6. Transport Layer: Transport Layer - Design issues, connection management, session Layer-Design issues, remote procedure call. Presentation Layer- Data compression techniques, cryptography. Application Layer: Application Layer: File Transfer, Access and Management, Electronic mail, Virtual Terminals, Internet and Public Networks.
