COMPUTER NETWORKS

UNIT I

Introduction: Overview of Networks, Circuit switching to packet switching principles, Protocols, protocol architecture, Reference Models, TCP/IP Model. Design Issues for the layers.

UNIT II

Physical Layer:

Concepts of Frequency, Spectrum, bandwidth. Wireless and Wired Transmission, Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission: Microwave, satellite communication etc.

UNIT III

Data Link Layer

Data Encoding: Introduction, Block coding, cyclic codes, checksum, framing, Noiseless channels, noisy channels, Asynchronous and Synchronous transmission, Full and Half duplex, Encoding schemes : BCA (NRZ, Bipolar AMI, B8ZS, HDB3, ASK, FSK,PSK,PCM,AM,FM,PM),

Data Link Control: Flow control: Stop and Wait, Sliding window, Error detection: Parity

Check, CRC. Error control: Stop and Wait ARQ, Go back-N ARQ, Selective-Reject ARQ, Brief idea of HDLC and other Data Link control protocols.

UNIT IV

Network layer

Logical addressing, internetworking, address mapping, LAN Technology: LAN architecture, IEEE 802 standards, Ethernet (CSMA/CD): Medium Access Control, Ethernet, Fast Ethernet, Brief survey of other LAN systems (Token ring, FDDI, ATM, Fiber channel). Wireless LANS, Bridges, Latest trends in LAN technologies LAN Devices: Study of specifications of L2 and L3 switches. IPv4, IPv6, IP multicasting, Principles of routing. Routing protocols. Link-state and distance vector routing.

UNIT V

Transport Layer:

Process to process delivery, UDP and TCP protocols, SCTP, data traffic, congestion, congestion control,

Application Layer : Principles of Internetworking, connection less Internetworking, HTTP, WWW, FTP, SMTP, SNMP, and MIME POP3, DNS, Firewall and Gateways.

COMPUTER ARCHITECTURE

UNIT I

Processor Basics

Processor Basics: CPU Organization: Fundamentals, additional features. Data representation: Basic formats, fixed point numbers, floating-point numbers. Instruction sets: Instruction formats, instruction types, programming considerations.

UNIT II

Data path Design

Data path Design: Fixed point arithmetic- Addition and subtraction, multiplication, division. Arithmetic Logic Unit: Combinational ALUs, sequential ALUs. Advanced topics: Floating-point arithmetic, pipeline processing.

UNIT III

Control Design

Control Design: Basic concepts: Introduction, hardwired control, design examples. Micro- programmed control: Basic concepts, multiplier control unit, CPU control unit. Pipeline control: Instruction pipelines, pipeline performance, super-scalar processing.

UNIT IV

Memory Organization

Memory Organization: Memory technology: Memory device characteristics, random-access memories, serial-access memories. Memory systems: Multilevel memories, address translation, memory allocation. Caches: Main features, address mapping, structure versus performance.

UNIT V

System Organization

System Organization: IO and System Control: Programmed IO, DMA and interrupts, IO processors. Parallel processing: Processor-level parallelism, multiprocessors.

CLOUD COMPUTING

UNIT I

Introduction Cloud Computing: Definition, Types of Clouds, Layer & Services models, Deployment models, Cloud Computing Architecture & infrastructure: Cloud Reference Model, Virtualization: Definition, Types of virtualizations (Compute, Network, Storage), Types of Hypervisor.

UNIT II

Cloud Platforms in Industry: Major vendors and their offerings, Introduction to Microsoft Azure, Amazon web services (EC2, S3, Etc.), Google AppEngine. Integration of Private and Public Clouds

Cloud applications: Protein structure prediction, Data Analysis, Satellite Image Processing, CRM and ERP, Social networking. Cloud Application- Scientific Application, Business Application.

UNIT III

Advance Topic in Cloud Computing: Cloud Security, Risks and Approaches of Migration into Cloud. Federated Cloud/ Intercloud, Third Party Cloud Services, Business Continuity and Disaster Recovery, Service Level Agreement (SLA), Dynamic resource provisioning and management, Server consolidation and placement policies, Energy efficiency in data centers, Elastic Load Balancing and Auto Scaling.

UNIT IV

Storage Network Design: Architecture of storage, analysis and planning. Storage network design considerations; NAS and FC SANs, hybrid storage networking technologies (iSCSI, FCIP, FCoE), design for storage virtualization in cloud computing, host system design considerations. Techniques for Big data processing (Google GFS, BigTable, and Map-Reduce Hadoop Distributed File System (HDFS), HIVE).

UNIT V

Consensus in Cloud Computing: Issues in consensus, Consensus in synchronous and asynchronous system, Byzantine Agreement: Agreement, Faults, Tolerance, Measuring Reliability and Performance, SLIs, SLOs, SLAs, TLAs, Byzantine failure, Byzantine Generals Problem, Failures & Recovery Approaches in Distributed Systems, Check pointing.

DESIGN & ANALYSIS OF ALGORITHMS

UNIT-I

Algorithms Analysis: Algorithms and structured programming. Analyzing algorithms, asymptotic behavior of an algorithm, recurrence relation, Order notations, time and space complexities, average and worst case analysis, lower and upper bounds.

UNIT-II

Algorithm design strategies: Divide and conquer (Merge sort, Quick sort, matrix multiplication), Greedy method (knapsack problem, minimum spanning trees).
Basic search & Traversal Techniques (Breadth first and Depth first traversals of Graphs).

UNIT-III

Dynamic programming: 0/1 knapsack, Travelling salesman problem Backtracking: 8-queen problem, sum of subsets, 0/1 Knapsack
Branch & Bound: 0/1 knapsack, Travelling salesman.

UNIT-IV

Matrix algorithms: Basics of matrices, Strassen's matrix-multiplication algorithm
Data structures for set manipulation problems: Fundamental operation on sets, a simple disjoint- set union algorithm, tree structures for UNION-FIND problem, applications and extensions of the UNION-FIND algorithm.

UNIT-V

Finite automata and regular expression, recognition of regular expression, patterns, recognition of substrings, Conversion from NFA to DFA
Complexity Theory: Overview, Turing machine, polynomial and non-polynomial problems, deterministic and non-deterministic algorithms, Idea of problem Classes: P class, NP class & NP complete problems.

JAVA PROGRAMMING

UNIT-I

OOPS Concepts: Encapsulation, Inheritance and polymorphism, Classes and data abstraction, constructors and destructors.

UNIT-II

Packages and Interfaces, Access Control, Method Overriding, Garbage Collection, Abstract Classes.
Exceptions handling, throwing an exception, try block, catching an exception, Multithreading, Synchronization.

UNIT-III

J2EE Platform: Enterprise architecture style (2 tier, 3 tier, N tier), J2EE run time, J2EE APIs, J2EE technology, web components
JDBC Overview, JDBC implementation, Connection class, Statements, Types of statement objects (Statement, Prepared Statement and Callable Statement), and Types of result set.

UNIT-IV

Servlet: Servlet API, Overview of Servlet, Servlet Life Cycle, HTTP Methods, Attributes in Servlet, Request Dispatcher interface.

UNIT-V

Java Server Pages: JSP Overview, Problem with Servlet, Life Cycle of JSP Page, JSP Processing, JSP Application Design with MVC, JSP Directives, JSP Action, JSP Implicit objects, JSP Session and Cookies Handling.

SOFTWARE ENGINEERING

UNIT-I

Software Engineering Fundamentals: Definition of Software, Software characteristics, Software Applications.

Software Process:

Software Process Models - Waterfall model, prototyping model, spiral model, incremental model, concurrent development model.

Project management Concepts: The Management Spectrum - The People, The Product, The Process, The Project.

UNIT –II

Software Process and Project Metrics: Measures, Metrics and Indicators, Software measurement: Size - Oriented Metrics, Function - Oriented Metrics, Extended Function point metrics.

Software Project Planning: Project Planning Objectives, Software Project Estimation, Decomposition Techniques - Problem Based Estimation, Process Based Estimation, Empirical Estimation Models- The COCOMO Model.

Risk Analysis and Management: Software risks, Risk identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring and Management.

UNIT –III

Software Quality Assurance: Basic concepts- Quality, Quality Control, Quality Assurance, Cost of Quality, Software Quality Assurance (SQA), Formal Technical Review.

Software Configuration Management: Baselines, Software Configuration Items, The SCM Process, Version Control, Change Control, Configuration Audit, Status Reporting.

Analysis Concepts and Principles: Requirements Elicitation for Software, Analysis Principles - The Information Domain, Modeling, Partitioning, Essential and Implementation Views, Specification: Specification Principles, Representation, The Software Requirement Specification (SRS).

UNIT –IV

Design Concepts and Principles: Design Principles, Design Concepts, Abstraction, Refinement, Modularity, Software Architecture, Control Hierarchy, Structural Partitioning, Data Structure, Software Procedure, Information Hiding, Effective Modular Design- Cohesion, Coupling.

UNIT –V

Software Testing: Testing Objectives & principles, Unit Testing, Integration Testing (Top Down Integration, Bottom Up Integration, Regression Testing, Smoke Testing), Validation Testing (Alpha and Beta Testing), System Testing (Recovery Testing, Security Testing, Stress Testing, Performance Testing).

Reengineering: Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering CASE Tools: What is CASE, Building Blocks of CASE, A Taxonomy of CASE Tools, Integrated CASE Environments, The Integration Architecture, The CASE Repository.

DESIGN & ANALYSIS OF ALGORITHMS LAB

Prerequisite:

1. Proficiency in a C & C++ programming language, basic program design concepts (e.g, pseudo code), proof techniques, familiarity with trees and graph data structures, familiarity with basic algorithms such as those for searching, and sorting

Objectives of the Course:

1. To identify and analyze worst-case running times of algorithms.
2. To model a given engineering problem using graphs and trees and write the corresponding algorithm to solve the problems.
3. To strengthen the ability to identify and apply the suitable algorithm for the given real world problem.

Learning Outcomes:

1. Design an algorithm in a effective manner
2. Apply iterative and recursive algorithms.
3. Design iterative and recursive algorithms
4. Implement optimization algorithms for specific applications.
5. Design optimization algorithms for specific applications

List of Some Programs for reference:

1. Implement Merge Sort
2. Implement Quick Sort
3. Implement Binary Search
4. Implement DFS and BFS traversals
5. Implement knapsack problem
6. Implement N Queens Problem
7. Implement minimum cost spanning tree
8. Implement all pair shortest path
9. Implement Strassen Multiplication

Suggested E-Resources:

Online virtual lab

Scheme of EoSE :

(i) Exercise(s)/ Experiment(s) :50 (ii)Viva Voce:20

(iii)Evaluation of record book: 10

JAVA PROGRAMMING LAB

Prerequisite:

1. The students should have basic knowledge of programming languages like C.
2. Prior knowledge of Object-Oriented Language like C++ is helpful but not mandatory.

Objectives:

1. The learning objective of this course are:
2. The course aims to strengthen the conceptual knowledge of Object Oriented Programming Language JAVA at core level and advance level.
3. Lay foundation for further learning of the subject C++ and Advance Java Programming Language which is useful for the design of desktop and web applications in Computer Science

Learning Outcomes:

After successful completion of the course, the students are able to:

1. The student will be able to learn conceptual and advanced level of programming with a modern programming language, Java.
2. The student gets an understanding of how to install and use a good Java development environment.
3. The student learn how to produce robust programs in Java using :Variables, Expressions, Looping,
   Branching, Methods / Functions, Classes, Interfaces and Inheritance, Package, Exceptions,
   Multithreading,
4. The student will be able to learn connectivity of the Java programming language with databases using JDBC.
5. Design web related applications which mimic the real world scenarios using servlets and JSP.

List of Sample programs for reference:

1. Program to develop an in depth understanding of Java: data types, variables, operators, operator precedence,
2. Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
3. Write object oriented programs in Java: objects, classes, constructors and destructors.
4. Write object oriented programs of inheritance, using super, final with inheritance, overloading and overriding
5. Programs on access control, static member, abstract classes and variables.
6. Program to develop understanding of packages & Interfaces in Java
7. Write a program using exception handling mechanisms.
8. Programs using JDBC.
9. Programs using Servlets.
10. Programs using JSP for web applications.