IV SEMESTER SYLLABUS

CSE 221 PROBABILITY & QUEUEING THEORY

Permutations, combinations, counting, summation, generating function, recurrence relations, asymptotic. Sample space and events- Probability- The axioms of probability- Some Elementary theorems- Conditional probability- Baye's theorem- Random variable- Discrete and continuous- Distribution- Distribution function, Distribution, Binomial and poison distribution Normal distribution- related properties.

Queuing theory- Classification, stationary process, markov process, Binomial process, Poisson process, Birth and death process, Markov chain.

Textbook:

1. Probability and Statistics with reliability, Queuing and Computer Science Trivedi K.S.

References:

1. Reliability Engineering Balagurusamy .E
2. Fundamental of Queuing Theory Gross D,and Harris C.M.
3. Probability Statistics and Queuing Theory Allen, A.O.

Course Outcomes:

- Acquainted with basic concepts of probability theory, permutations, combinations, Sample space, axioms of probability.
- Introduced to the techniques of developing discrete & continuous probability distributions and its applications.
- Able to describe a random process in terms of its mean and correlation functions.
- Aware with the special processes like Poisson, Renewal processes, birth & death process and derive the backwards & forwards equation.
- Able to specify a given discrete and Markov chain in terms of a transition diagram.
- To understand basic characteristic features of a queuing system and acquire skills in analyzing queuing models.

CSE 222 COMPUTER ARCHITECTURE

Central processor organizations: basic building blocks, bus organized computer memory, address structure, register transfer languages, instruction formats, expanding op-codes and addressing modes. Control unit organization: hardwired control & micro-programmed control organization, control memory, address sequencing micro-instruction formats, micro-program sequencer, micro-programming. Arithmetic processor design: addition and subtraction algorithm, multiplication algorithm, division algorithm, processor configuration, and floating point arithmetic. Input-Output organization: Asynchronous Data Transfer, Asynchronous Communication Interface, Modes of Transfer: Interrupt-Initiated, Direct Memory Access (DMA). Memory Organization: Main Memory, Auxiliary Memory, Associative Memory: Hardware Organization, Cache Memory: Mapping Schemes, Virtual Memory: Address Space and Memory Space, Address Mapping. Structure of multiprocessors, Introduction to parallel processing, Flynn's classification, pipeline processing, pipeline hazards.

Textbook:

1. Computer Architecture Morris Mano,

References:

1. Computer Organization and architecture William Stallings
2. Computer Organisation& Architecture T.K. Ghosh,

Course Outcomes:

- Learn about functioning of computer.
- Learn about structure of computer.
- Learn about various components used in computers like CPU, Memory and I/O.
- Learn about computer design (ALU and CU design).

CSE 223 THEORY OF COMPUTATION

Finite State Systems, Regular Expressions, Output machines, Regular sets, Context Free Grammar (CFG), simplification of CFG, normalization of CFG, Push Down Automata, CFL, CSL and LBA, Turing Machine, Recursive and RE sets, undecidability, Chomsky Hierarchy.

Text Book:

1. Introduction to automata theory, language & computations Hopcroaft& O.D. Ullman, R Mothwani,

Reference Books:

1. Theory of Computer Sc. (Automata, Languages and computation): K.L.P. Mishra & N. Chandrasekaran,
2. Introduction to formal Languages & Automata Peter Linz,
3. Fundamentals of the Theory of Computation- Principles and Practice RamondGreenlaw and H. James Hoover
4. Introduction to the Theory of Computation Michael Sipser

Course Outcomes:

- Able to construct deterministic and nondeterministic finite state automata (DFA and NFA) for solving simple decision problems.
- Able to evaluate the computational complexity of algorithms involved in the decision problems of finite state automata, and using pumping lemma to demonstrate the non-regularity of languages.
- Able to compute the minimal state machine corresponding to a DFA.
- Understand and construct context-free grammars (CFG) for formal definitions involving recursion such as regular expressions; and also understand the fundamental role played by CFG in designing formal computer languages with simple examples.
- Understand basic properties of Turing machines and computing with Turing machines.
- Able to evaluate the computational complexity of algorithms involved in the decision problems of context free grammars and push down automaton, classify and simplify grammars into their useful canonical forms.

CSE 224 DATABASEMANAGEMENT SYSTEM

Fundamentals of DBMS, different data models. Relational database systems. ER modelling, Enhanced ER Model, ER to Relational Mapping. Relational Database Design, integrity constraints, functional dependency constraints, assertions, triggers, Normalization in relational approach. Relational algebra and calculus. SQL, overview of query processing and cost estimation, Query Optimization, Transaction processing and concurrency control. Data storage and indexing, B-Trees and B+ Trees, Overview of advanced databases.

Textbook:

1. Fundamentals of Database Systems Elmasri&Navathe

References:

1. Database System Concepts Silberschatz, Korth&Sudershan .
2. An Introduction to Database Systems C. J. Date

Course Outcomes:

- Learn about the concept of data, databases and database management systems.
- Learn about how to handle databases.

CSE 225 ANALYSIS & DESIGN OF ALGORITHMS

Fundamentals of algorithm, asymptotic complexity, recursive algorithms, recurrence relation, heap, priority queue and heap sort. Algorithm Design Techniques their control abstractions and related problems: Divide and conquer, Greedy strategy, Dynamic programming, Backtracking, Branch and bound, least cost search. Introduction to lower-bound theory, Search Trees: BST, AVL, B and B+ trees. Introduction to NP-Complete and NP Hard problems.

Textbook:

1. Computer Algorithms Horowitz and Sahani.

References:

1. Introduction to Algorithms Cormen and Rivest
2. An Introduction to Algorithms Thomas H Cormen,Ronald L. Rivest

Course Outcomes:

- Learn different algorithm design techniques and study related problems.
- Learn to determine algorithm correctness and its efficiency.
- Learn various searching, sorting and graph traversal algorithms.
- How search trees help in searching effectively.
- Learn to use various techniques for efficient algorithm design like divide and conquer, greedy and dynamic algorithms.
- Understand NP completeness and identify different NP complete problems.

CSE 226 SOFTWARE ENGINEERING

Introduction to software engineering, software process & process models, Software metrics and measurements, software project management, software project planning, scheduling and tracking, cost estimation methods. Requirements analysis: Principles, complexity, methods, structured analysis, SRS Documentation. Design principles: abstraction, refinement, modularity, control hierarchy, structured partitioning, design types and methods. Software coding: coding style, coding efficiency, capability maturity model (CMM), Software quality assurance, Software testing: Software testing techniques, choice and classification of test data, verification & validation methods. Software maintenance, configuration management, system documentation, software reusability.

Textbook:

1. An Integrated Approach to Software Engineering Pankaj Jalote,
2. Software Engineering: A Practitioner's Approach R S. Pressman.

References:

1. Pearson Edu, “Software Engineering by Ian sommerville”, 9th edition, 2010

Course Outcomes: -

- The main purpose of this course is to impart knowledge on the basic principles of software development life cycle.
- Understand the software life cycle models.
- Understand the importance of the software development process.
- Understand the importance of modeling and modeling languages.
- Design and develop correct and robust software products.

CSE 311 COMPILER DESIGN

Compilers and translators, structure of compiler its different phases, Compiler construction tools. Lexical analyzer, Specification and recognition of tokens, input buffering.Syntax analyzer, top down and bottom up parsing. Syntax directed definition, syntax directed translation scheme, intermediate codes: syntax tree, post fixed expressions,three address code, quadruples and triples. Code optimization, DAG, Code generation,Symbol table implementation, Error handling