**1.Database Systems**

INTRODUCTION:

Database-System Applications, Purpose of Database Systems, View of Data,

Database Languages, Relational Databases, Database Design, Data Storage and

Querying, Transaction Management, Database Architecture, Database Users and

Administrators, NoSQL, Sharding.

RELATIONAL MODEL:

Structure of Relational Databases, Database Schemas, Keys, Schema Diagrams,

Relational Query Languages, Relational Operations, Relational Algebra –

Fundamental Operations, Formal Definition of Relational Algebra, Extended

Relational Algebra Operations.

Module -2

STRUCTURED QUERY LANGUAGE:

SQL Data Definition, SQL Data Types and Schemas, Integrity Constraints, Basic

Structure of SQL Queries, Set Operations, Aggregate Functions, Nested

Subqueries, Additional Basic Operations Null Values, Modification of the

Database. Join Expressions, Views, Transactions.

Module – 3

DATABASE DESIGN USING E-R MODEL:

Overview of the Design Process, The Entity-Relationship Model, Constraints,

Removing Redundant Attributes in Entity Sets, Entity- Relationship Diagrams,

Entity-Relationship Design Issues, Extended E-R Features, Reduction to

Relational Schemas.

NORMALIZATION:

Features of Good Relational Design, Atomic Domains and First Normal Form,

Decomposition Using Functional Dependencies, Functional Dependency Theory,

Algorithms for Decomposition ,Decomposition Using Multivalued Dependencies.

Module-4

INDEXING AND HASHING:

File Organization, Organization of Records in Files, Basic concepts, Ordered

Indices, B+ Tree Index Files, B+ Tree Extensions, Multiple-Key Access, Static

Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing.,

Bitmap Indices

TRANSACTION MANAGEMENT:

Transaction Concept, A simple Transaction model, Transaction Atomicity and

Durability, Transaction Isolation , Serializability, Transaction Isolation and

Atomicity, Transaction Isolation Levels, Failure Classification, Storage, Recovery

and Atomicity, Recovery algorithm.

**2.Formal Languages and Automata Theory**

1. Introductory class (Introduction between teacher & students. Overview of the subject).

2. INTRODUCTION TO THE THEORY OF COMPUTATION AND FINITEAUTOMATA

3. Three basic concepts: Languages, Grammars and Automata

4. Grammars

5. Some applications

6. Deterministic Finite Accepter

7. Nondeterministic Finite Accepters

8. Equivalence of Deterministic and Nondeterministic Finite Accepters

9. Reduction of the Number of States in Finite Automata

10. REGULAR LANGUAGES, REGULAR GRAMMARS AND PROPERTIESOF REGULAR LANGUAGES

11. Regular Expressions

12. Connection between Regular Expressions and Regular Languages

13. Connection between Regular Expressions and Regular Languages (Contd.)

14. Regular Grammars

15. Closure Properties of Regular Languages

16. Identifying Non-regular Languages

17. CONTEXT-FREE LANGUAGES AND SIMPLIFICATION OFCONTEXT-FREE GRAMMARS AND NORMAL FORMS

18. Context-Free grammars

19. Parsing and Ambiguity

20. Context-Free Grammars and programming language

21. Methods for Transforming Grammars

22. Methods for Transforming Grammars (Contd.)

23. Two important Normal Forms

24. PUSHDOWN AUTOMATA AND PROPERTIES OF CONTEXT-FREE LANGUAGES

25. Nondeterministic Pushdown Automata

26. Pushdown Automata and Context–Free Languages

27. Pushdown Automata and Context–Free Languages (Contd.)

28. Deterministic Pushdown Automata and Deterministic Context-Free Languages

29. Two Pumping Lemma

30. Two Pumping Lemma (Contd.)

31. Closure properties of Context-Free Language

32. TURING MACHINES AND OTHER MODELS OF TURING MACHINES

33. The Standard Turing Machines

34. Non-deterministic Turing Machines, Linear Bounded Automata

35. A HIERARCHY OF FORMAL LANGUAGES & AUTOMATA

36. Recursive and Recursively Enumerable Languages, Unrestricted grammars

37. Context-Sensitive Grammars and Languages, The Chomsky Hierarchy

**3.Design and Analysis of Algorithm**

1. Introduction to course; What is an Algorithm?
2. Fundamentals of Algorithmic Problem Solving, Important Problem Types
3. Fundamental Data Structures, Analysis Framework
4. Tutorial-1 : Examples on Data structure, Asymptotic notations
5. Asymptotic Notations and Basic Efficiency Classes
6. Mathematical Analysis of Non-recursive Algorithms and Recursive Algorithms
7. Selection Sort and Bubble Sort
8. Tutorial-2 : Example on Non-recursive, Recursive Algorithms and recurrence
9. relation solution
10. Sequential Search, Brute-Force String Matching
11. Exhaustive Search
12. Depth First Search Breadth First Search
13. Tutorial-3 : Examples on Exhaustive Search DFS and BFS
14. Insertion Sort
15. Topological Sorting
16. Binary Search
17. Tutorial-4 : Examples on Binary Search Insertion sort, topological sorting
18. Merge Sort
19. Quick Sort
20. Binary tree traversals and related properties, Multiplication of large integers
21. Tutorial-5 : Examples on Merge Sort, Quick Sort, Multiplication of large integers
22. Stassen’s Matrix Multiplication
23. Presorting, Balanced Search Trees — AVL trees
24. Balanced Search Trees — 2-3 trees
25. Tutorial-6 Examples on AVL tree, 2-3 tree
26. Heaps and Heapsort
27. Problem Reduction, Sorting by Counting
28. Input Enhancement in String Matching - Horspool algorithm
29. Tutorial-7 : Examples on Heaps and Heapsort , Horspool algorithm
30. Boyer-Moore Algorithm
31. Hashing
32. Computing a Binomial Coefficient, Warshall's algorithm
33. Tutorial-8 : Examples on Boyer-Moore Algorithm, Hashing, Warshall's algorithm
34. Floyd's Algorithm
35. Knapsack Problem Bottom-up
36. Knapsack-Memory Functions
37. Tutorial-9 : Examples on Floyd's Algorithm, Knapsack Problem Bottom-up &
38. Memory Functions
39. Prim's Algorithm
40. Kruskal's Algorithm
41. Dijkstra’s Algorithm
42. Tutorial-10 : Examples on Prim's, Kruskal's & Dijkstra’s Algorithm
43. Huffman Trees
44. Backtracking: n — Queen's problem
45. Backtracking: Hamiltonian Circuit Problem, Subset-Sum Problem
46. Tutorial-11: Examples on Huffman Trees, Backtracking
47. Branch-and-Bound: Assignment Problem, Branch-and-Bound: Knapsack Problem
48. Traveling Salesperson
49. P, NP and NP Complete Problems : Definition
50. Tutorial-12 : Examples on Branch-and-Bound , P, NP problems

**4.Engineering Maths-IV**

1 Introduction to the course, Definition and Axioms of probability.

2 Addition rule , independent events, problems.

3 ditional probability, problems.

4 Total probability, problems.

5 Baye’s Theorem with proof, problems.

6 Tutorial

7 One dimensional random variables, CDF, Mode, Median, problems.

8 Mean and Variance of one dimensional random variables, Chebyshev’s inequality

without proof, Problems.

9 Tutorial

10 Two dimensional Random variables, Marginal Pdf’s, problems.

11 Mean and variance of discrete and tinuous random variables, ditional

probability function and conditional pdf’s, Problems.

12 ariance and relation efficient, Properties, Problems.

13 Tutorial

14 Probability distributions: Binomial distribution, mean and variance with problems

15 Poisson’s distribution - mean and variance, Uniform distribution - mean and

variance.

16 Tutorial

17 Normal distribution, mean and variance , problems

18 Problems on Normal distribution.

19 Gamma, Exponential and Chi- Square Distribution -- mean and variance.

20 Gamma, Exponential and Chi- Square Distribution : Problems.

21 Functions of one dimensional random variables, Problems.

22 Functions of two dimensional random variables, Problems.

23 F and t- distribution (Definition only) and Problems

24 Tutorial

25 Moment generating functions (mgf), Problems

26 Problems related to mgf of both tinuos and discrete random variables.

27 Introduction to Sampling Theory and related problems

28 Central limit theorem with proof, Problems.

29 Problems on Central limit theorem.

30 Point estimation, problems.

31 Maximum Likelihood estimator (MLE) , problems.

32 Significance level, critical region and power of the test, Problems.

33 Testing of Hypothesis with problems

34 Chi-square test, problems

35 Best critical region, Neyman-Pearson lemma, Problems

36 Tutorial

**5.Embedded ARM Systems**

1. Introduction to Embedded Systems, Microprocessors and Microcontrollers, An overview of ARM-Cortex- M Architecture: General purpose registers

2. Load store instructions in ARM, ARM CPSR, ARM Data format

3. Pseudo instructions and Directives

4. Introduction to ARM Assembly Programming, The Program Counter and Program Memory space in the Arm

5. Data Transfer Programming

6. ARM Addressing modes, RISC Architecture in ARM

7. Arithmetic Instructions-addition and subtraction

8. Programs on addition and subtraction

9. Multiplication instructions and programming

10. Division Programming

11. Logic Instructions, Rotate and Barrel Shifter, Shift and Rotate Instructions

12. BCD and ASCII conversion

13. Looping and Branch Instructions, Calling Subroutine and Return

14. Conditional Execution

15. Programming using logical instructions-BCD and ASCII concersion

16. Recursion

17. ARM Memory Map, Memory Access and Stack

18. Stack and Stack usage in ARM

19. Different types of Stack Programming

20. Programming using advanced addressing mode and stack

21. ARM Bit Addressable Memory Region

22. Address Calculation

23. Pin connect block, Pin function select registers

24. General Purpose Input and Output (GPIO) registers, GPIO configuration

25. GPIO programming using ARM C language

26. Interfacing: LEDs

27. Seven Segment Display

28. LCD

29. Keyboard

30. DC motor, Stepper motor

31. Timer versus counter

32. Timer Registers

33. Timer architecture and operation

34. PWM timer and architecture

35. Timer/counter programming

36. PWM Programming

37. General introduction to serial interfacing, RS232, MAX 232

38. UART, UART programming

39. Data acquisition system, Analog to Digital Converter (ADC), ADC registers

40. Digital to Analog converter (DAC), DAC registers

41. ADC programming

42. DAC programming

43. Hardware and software synchronization, multithreading

44. Nested Vectored Interrupt Controller (NVIC), external hardware interrupts

45. IO interrupts, SysTick interrupts

46. Timer/counter interrupts

47. ADC and DAC interrupts, UART interrupts

48. Interrupt programming