Jyoti Nivas College Autonomous

Post Graduate Centre
Hosur Road, Bangalore -95

Phone No: 255012143 Fax: 25533886
e-mail:jncpgcourses@gmail.com
Website: www.jyotinivas.org

Proposed Scheme & Syllabus

M. Sc in Computer Science

w. e. f. 2023-25 Batch onwards

CONTENTS

MSc programme in Computer Science is designed to provide an insight into computing through advanced concepts, principles, strategies and skills supplemented with practical knowledge to effectively develop and work with a range of technologies to build systems and applications that help apply in real-time computing environments. The program combines strong fundamentals, projects, and team-oriented activities, leading to a holistic education.

PROGRAMME OBJECTIVES:

- Provide a strong foundation in core areas of Computer Science
- Impart knowledge of latest developments in the field of IT
- Identify research gaps and provide innovative solutions
- Adhere to social, ethical, and cyber values

PROGRAMME SPECIFIC OUTCOMES:

In the span of four semesters, the students are well equipped with programming skills, analytical ability, research skills and strong knowledge in the domain. The students will be able to:

- Analytical and critical thinking abilities to solve real time problems
- Design, develop and maintain software applications for different verticals
- Ability to communicate ideas and solutions and work in a collaborative manner
- Amenable to life-long learning of new tools and technology

PROGRAMME STRUCTURE

- The M.Sc programme has a curriculum with syllabi consisting of (i) Core courses, (Theory & Practical) (ii) Skill Enhancement Courses, (iii) Discipline specific Elective Courses (iv) Open Electives and (v) Research Project work.
- In addition, add-on courses, group activities, soft skills, communication skills and value added training are provided based on the need.
- The total number of credits assigned to the course is 98 and the credits per paper are distributed as follows:
    - 4 credits per core theory courses with 4 hours of lecture/tutorial per week
    - 2 credits per practical courses with 4 hours of practical/project per week
    - 4 credits for discipline specific elective courses with 4 hours of lecture/tutorial per week
    - 4 credits per open elective paper with 4 lecture hours per week
    - 12 credits for Research Project

- Core courses are compulsory subjects offered by the department. Total credits for Core Courses including theory, practical papers, skill enhancement courses and projects are 82.
- Elective courses may be chosen by the student from the list of Electives offered by the department. Total credits for Elective Courses are 12.
- Open Elective has to be taken from other post graduate disciplines. Credit for open Elective course is 4.

REGULATION AND SCHEME

1. **Eligibility:**
    a. B.Sc. (Computer Science) or BCA with Mathematics as one of the subject and at least 50% aggregate marks of all optional subjects (throughout 3 years B.Sc. / BCA course), B.Sc. PCM with PG Diploma / Certificate in Computer Science of duration one year.
    b. The minimum requirement for SC / ST candidates are relaxed in accordance with University regulations.

2. **Duration of the Course: 2** years (**4** semesters)

3. **Medium of Instruction:** The medium of Instruction and Examination shall be in English.

4. **Proposed Intake:** 45

5. **Mode of Admission:** A category wise merit list will be prepared with marks obtained in all optional in all the three years

6. **Evaluation Procedure:**

    a. Continuous Internal Assessment for theory (CIA): 30 Marks

Component	**Marks**
Mid Term Examination	15
Assignment / Presentation / Case Study / Mini Project	10
Group Activity	5
Total	**30**

    b. End Semester Examination for theory (ESE) : 70 Marks

    c. Continuous Internal Assessment for Practical (CIA): 30 Marks

Component	**Marks**
Mid Term Examination	15
Review of the work done	10
Documentation	5
Total	**30**

    d. End Semester Examination for Practical : 35 Marks

    e. Students should secure a paper minimum of 40% each in CIA, end semester theory / practical examination and an aggregate of 50% in a semester.

GRADUATE ATTRIBUTES

1. Domain proficiency

Generally defined, domain proficiency implies knowledge and understanding of the essential aspects of a specific field of inquiry. Domain proficiency is used to evaluate the inputs, guide the process, and evaluate the end products within the context of value and validity.

2. Applied knowledge

It's about the description and demonstration of procedures and tasks that are learnt in the classroom. The knowledge could be applied to develop or revise procedures or algorithms to address a problem / situation.

3. Joint Effort

Joint effort is the collaborative effort of a group to achieve a common goal or to complete a task in the most effective and efficient way. This concept is seen within the greater framework of a team, which is a group of interdependent individuals who work together towards a common goal.

4. Problem solving

Problem solving is the act of defining a problem; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution.

5. Design and development

The design and development process should examine all the potential challenges and hurdles the students will need to overcome to develop an effective application. It involves defining the specification and the design that is simple yet creative and user friendly.

6. Research skills

Research skills refer to the ability to search for, locate, extract, organise, evaluate and use or present information that is relevant to a particular topic. It involves intensive search, investigation, and critical analysis, usually in response to a specific research question or hypothesis.

7. Application Governance.

Application Governance refers to the branch of project management dedicated to the planning, scheduling, resource allocation, execution, tracking and delivery of software and web projects.

8. Usage of modern tools.

Introducing the latest and open source tools that are available to handle the application development.

9. Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

SEMESTER-WISE SCHEDULE

Semester	Course Code	Course Title	Hours / Week (32)			Credits	Marks		
			L	T	P		CIA	ESE	Total
I	22MCS101	Modern Operating Systems	4	–	–	4	30	70	100
	22MCS102	Advanced Computer Networks	4	–	–	4	30	70	100
	22MCS103	Mathematical Foundations for Computer Science	4	–	–	4	30	70	100
	22MCS104	Theory of Computation	4	–	–	4	30	70	100
	22MCS105	Data Structures	4	–	4	6	30+15	70+35	150
	22MCS106	Advanced Database Management System	4	–	4	6	30+15	70+35	150
		TOTAL	**24**	**0**	**8**	**26**	**210**	**490**	**700**
II	22MCS201	Advanced Algorithms	4	–	–	4	30	70	100
	22MCS202	Artificial Intelligence	4	–	–	4	30	70	100
	22MCS203	Probability and Statistics	4	–	–	4	30	70	100
	22MCS204	Data Analytics	4	–	–	4	30	70	100
	22MCS205	Web Technology	4	–	4	6	30+15	70+35	150
	22MCS206	Exploring Python	4	–	4	6	30+15	70+35	150
		TOTAL	**24**	**0**	**8**	**26**	**210**	**490**	**700**
III	22MCS301	Cloud Computing	4	–	–	4	30	70	100
	22MCS302	Research Methodology	4	–	–	4	30	70	100
	22MCS303(A-D)	Elective - I	4	–	–	4	30	70	100
	22MCS304	Open Elective	4	–	–	4	30	70	100
	22MCS305	Machine Learning	4	–	4	6	30+15	70+35	150
	22MCS306	Mobile Application Development	4	–	4	6	30+15	70+35	150
		TOTAL	**24**	**0**	**8**	**26**	**210**	**490**	**700**
IV	22MCS401	Advanced Machine Learning	4	–	–	4	30	70	100
	22MCS402	Data Visualization	4	–	–	4	30	70	100
	22MCS403(A-D)	Elective II	4	–	–	4	30	70	100
	22MCS404	Research Project	–	–	16	8	50	150	200
		TOTAL	**12**	**0**	**16**	**20**	**140**	**360**	**500**

Total Credits (1- 4 semesters): 98

List of Discipline Specific Electives

Elective I	
22MCA302A	Digital Image Processing
22MCA302B	Software Engineering Methodologies
22MCA302C	Database and Application Security
22MCA302D	Internet of Things (IoT)

Elective II	
22MCA303A	Natural Language Processing
22MCA303B	Software Testing
22MCA303C	Ethical Hacking
22MCA303D	Blockchain Technology

List of Open Elective Courses available for MCA Students

- Personal Wealth Management
- Approaches to Texts
- Life Style Management
- Chemistry in Daily Life

Open Elective offered to other Discipline Students

- Cyber Security

Add-on Certificate Courses

- Logic N Life
- Soft Skills & Communication Skills
- Academic Writing
- Recent Technologies

Semester I

22MCS101 MODERN OPERATING SYSTEMS

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: This course provides the basics of Operating System, key concepts from Distributed Operating System, real time and mobile operating systems. The students will be equipped with the knowledge about the usage of Operating Systems at different levels.

Learning Outcomes

After completion of this course, the student will be able to:

- Discuss the various synchronization, scheduling and memory management issues
- Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
- Identify the different features of real time and mobile operating systems
- Gain knowledge on Operating Systems used in embedded systems.

Unit I: Introduction: Types of Operating Systems, Operating System Structures, Components & Services, Process Management: Process Concept, Process Scheduling, Co – Operating process, Threads, Inter process communication. Process Synchronization and deadlocks: The Critical Section Problem, Synchronization hardware, Semaphores, Classical problems of synchronization. Dead locks – system model, Characterization, Dead lock prevention, avoidance and detection, Recovery from dead lock.

(12)

Unit II: Distributed Operating System: Issues in Distributed Operating System – Architecture – Communication Primitives – Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, and Election Algorithms. Lamport's Logical clocks – Agreement Protocols.

(12)

Unit III: Distributed Shared Memory: Introduction, General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, and Advantages of DSM.

(12)

Unit IV: Mobile and Real Time Operating Systems: Basic Model of Real Time Systems – Characteristics – Applications of Real Time Systems – Real Time Task Scheduling – Handling Resource Sharing. Mobile Operating Systems – Architecture – Layers – Microkernel Design – Kernel Extensions – Processes and Threads – Memory Management – File system – Android – iOS.

(12)

Unit V: Embedded Operating Systems :Operating Systems for Wireless Sensor Networks – Introduction - Operating System Design Issues - Examples of Operating Systems – TinyOS – Mate – MagnetOS – MANTIS - OSPM - EYES OS – SenOS – EMERALDS – PicOS – Introduction to Tiny OS – NesC – Interfaces and Modules- Configurations and Wiring - Generic Components -Programming in Tiny OS using NesC, Emulator TOSSIM. **(12)**

Reference Books:

1. Abraham Silberschatz and Peter Baer Galvin, "Operating System Concepts", 7th Edition, Pearson Education, 2002.
2. Mukesh Singhal and Niranjan G. Shivaratri, "Advanced Concepts in Operating Systems – Distributed, Database, and Multiprocessor Operating Systems", Tata McGraw-Hill, 2001.
3. Distributed Operating Systems Andrew S.Tanenbaum Pearson Education 2013
4. Rajib Mall, Real-Time Systems: Theory and Practice, Prentice Hall, 2006.
5. TinyOS Programming - Philip Levis, 2009

22MCS102 ADVANCED COMPUTER NETWORKS

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: This course provides the broad coverage of data communication system and computer networks. The students will be equipped with the knowledge about communication system components, internetworking, network topology, protocols, and algorithms.

Learning Outcomes

After completion of this course, the student will be able to:

- Understand the concept of networks, different topologies and network devices
- Discuss the objectives and functionalities of different layers.
- Describe how the available methods and algorithms are implemented in the real-time networks such as Ethernet, Bluetooth and internet protocols.
- Understand the working of few application protocols such as SMTP, POP and HTTP.

Unit I: Introduction - Protocol and Standards - Hierarchies, Network Models, Layered Tasks - OSI Reference Model, Introduction to internetworking, TCP/IP Model.

Data Communication: Analog and Digital Signals, Digital Transmission – Line coding schemes- unipolar and bipolar, Block coding- 4B/5B, - Sampling and Transmission Mode, Modulation Techniques, Networking Devices – hubs, switches, bridges, routers and gateways, Bandwidth utilization - Multiplexing and Spreading, Switching Techniques-Packet switching. **(12)**

Unit II: Optical Networking – Introduction, SONET / SDH Standard, WAN protocol – Introduction to ATM, basic concepts of ATM Networking, ATM cells.
LAN Protocol Architecture – Wireless LAN's, Bluetooth, and High-Speed LAN's - Need for High speed LAN’s – Ethernet. **(12)**

Unit III: Internet and Transport Protocols - Internet basics, IP Protocol, ICMP — The Internet Control Message Protocol, ARP—The Address Resolution Protocol. Introduction to Intra-domain and inter-domain routings, Unicast Routing Protocols, Multicast Routing Protocols, Overview of RIP, OSPF, MOSPF and BGP protocols, Transport protocols - TCP, UDP protocols. **(12)**

Unit IV: Software Defined Networking: Evolution of Switches and Control Planes, cost, Fundamental characteristics, operations, SDN devices, SDN controller, Openflow Specification: Openflow overview, OpenFlow 1.0 and 1.1, Introduction to Network Functions Virtualization, SDN vs NFV. **(12)**

Unit V: Network Security: Security Services, Message Confidentiality – Symmetric Key Cryptography, AES algorithm, Public Key cryptography, RSA algorithm, Message Integrity – hashing, SHA algorithm, MAC, Digital Signature, Entity Authentication, Key Management – KDC, Diffie Hellman and Kerberos. **(12)**

Reference Books:

1. Andrew S. Tanenbaum, "Computer Networks", Fifth Edition, Pearson Education, 2013.
2. Behrouz A. Forouzan, "Data communications and Networking", Tata McGraw-Hill, Fourth Edition, 2017.
3. Stallings William, "Data & Computer Communications", Pearson Education Asia, Tenth Edition, 2017.
4. Software Defined Networks, "A Comprehensive Approach", Paul Göransson, Chuck Black, 2014, Elsevier, ISBN: 978-0-12-416675-2
5. Dayanand Ambawade, Deven Shah, Mahendra Mehra, Mayank Agarwal, "Advanced Computer Network", Wiley Publications, 2011.

22MCS103 MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: The aim of this course is to provide the grounding knowledge of various mathematical concepts and its application in computer science domain.

Course Outcome:

- Understand the concept and applications of vector spaces, subspaces and linear independence.
- Understand various inner products and able to perform various inner product operations.
- Explore the applicability of general Linear Transformations, Linear operators, Composition of operators and linear transformations.
    - Understand the basics of Lattices and Boolean algebra.

Unit-I: Set Theory and Matrices
Sets, Operations on sets, Cardinality of sets, inclusion-exclusion principle, pigeonhole principle, matrices, finding Eigen values and Eigen vectors. **(12)**

Unit-II: Vector Spaces: General Vector Spaces, Subspaces, Linear Independence, Basis and Dimension, Span, Fundamental Theorems, Row Space, Column Space, Nullspace, Rank and Nullity, Four Fundamental Spaces. Inner Product Spaces: General Inner Products, Euclidean and Weighted Inner Product, Length, Distance, Norm, Angle and Orthogonality in Inner Product Spaces, Caucy-Schwarz Inequality, Orthogonal Complement, Orthonormal Bases, Gram-Schmidt Procedure. **(12)**

Unit-III: Linear Transformations: General Liner Transformations, Linear operators, Composition of operators and linear transformations, Kernel and Range, Dimension theorem for Linear Transformation, Inverse Linear Transformations, Matrices of General Linear Transformations. **(12)**

Unit-IV: Lattices and Boolean algebra: Partially ordered set, relations, maps, order relations, recursion and Introduction, Poset, closures, orderings, total orderings, Hasse diagrams. Includes important theorems on lattices. **(12)**

Unit V: Graphs: Introduction, Graphs and Graph models, Graph Terminology and Special types of Graphs , Representing Graphs & Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest Path Problems, Planar Graphs, Graph Coloring -Chromatic Polynomials.
Trees: Introduction, Properties-Rooted Trees- Spanning Trees, Minimum Spanning Trees. **(12)**

Reference Books:

1. Howard Anton and Chris Rorres, "Elementary Linear Algebra", John Wiley and Sons, 9th Edition, 2008.
2. Douglas C. Montgomery and George C. Runger, "Applied Statistics and Probability for Engineers", John Wiley and Sons, 3rd Edition, 2003.
3. Ronald E Walpole, Raymond H Myers, Sharon L Myers, “Probability & Statistics for engineers and scientists”, sixth edition, Prentice Hall.
4. Trivedi et.al, “Probability and statistics with computer applications”, Tata McGraw Hill.
5. Gupta S.C & Kapoor V.K, Fundamentals of Mathematical statistics, Sultan Chand & sons, 2009.
6. “Kenneth H.Rosen, Discrete Mathematics and Its Applications”, Tata McGraw-Hill, Fifth Edition 2003.

22MCS104 THEORY OF COMPUTATION

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: This course provides the broad coverage of automata and regular expressions. The students will be equipped with the knowledge about grammars, push down automata and Turing machines.

Learning Outcomes:
After completion of this course students will be able to:

1) Implement programming techniques for Turing Machine
2) Understand language of Pushdown Automata
3) Draw parse trees.
4) Understand computational complexity

UNIT I: Automata and Regular Expressions
Introduction to formal proof – Additional forms of proof – Inductive proofs –Finite Automata (FA) – Deterministic Finite Automata (DFA) – Non-deterministic Finite Automata (NDFA) – Finite Automata with Epsilon transitions.
Regular Expression – FA and Regular Expressions – Proving languages not to be regular – Closure properties of regular languages – Equivalence and minimization of Automata. **(12)**

Unit II: Grammars
Grammar Introduction– Types of Grammar - Context Free Grammars and Languages–– Parse Trees – Ambiguity in grammars and languages – Simplification of CFG – Elimination of Useless symbols - Unit productions - Null productions – Normal forms for CFG- Greiback Normal form – Chomsky normal form – Pumping Lemma for CFL. **(12)**

Unit III: Push Down Automata
Definition of the Pushdown automata – Languages of a Pushdown Automata – Equivalence of Pushdown automata and CFG– Deterministic Pushdown Automata. **(12)**

Unit IV: Turing Machines
Turing Machines, Introduction – Formal definition of Turing machines – Instantaneous descriptions- Turing Machine as Acceptors – Turing Machine as Transducers- Programming Techniques for Turing Machines– Modifications of Turing Machines. **(12)**

Unit V: Computational Complexity
Undecidability- Basic definitions- Decidable and undecidable problems - Properties of Recursive and Recursively enumerable languages – Undecidable problems about Turing Machine – Introduction to NP-Hardness and NP-Completeness. **(12)**

Reference Books:

1. Hopcroft J.E., Motwani R. and Ullman J.D, "Introduction to Automata Theory, Languages and Computations", Third Edition, Pearson Education, 2008.

2. H.R. Lewis and C.H. Papadimitriou, "Elements of the theory of Computation", SecondEdition, Pearson Education,2003.

3. Thomas A. Sudkamp," An Introduction to the Theory of Computer Science, Languages and Machines", Third Edition, Pearson Education, 2007.

4. Raymond Greenlaw an H.James Hoover, "Fundamentals of Theory of Computation, Principles and Practice", Morgan Kaufmann Publishers, 1998.

5. Micheal Sipser,"Introduction of the Theory of Computation", Thomson Brokecole,1997.

6. J. Martin, "Introduction to Languages and the Theory of computation" Third Edition, Tata Mc Graw Hill, 2007.

22MCS105 DATA STRUCTURES

Total No. of Hours: 60 +60 **Hours per week: L: T: P (4:0:4)**

Course Objectives: Computer science is primarily concerned with the study of data. It is important to introduce the student to these aspects of data and data structures which are required in modular programming. The basic algorithms related to handle data like stack, queue, tree and graphs are introduced in this subject.

Learning Outcomes:

After completion of this course, the student will be able to:

- Understand basic data structures such as arrays, linked lists, stacks and queues.
- Ability to program data structures and use them in implementations of abstract data types.
- Describe the hash function and concepts of collision and its resolution methods
- Solve problem involving graphs, trees
- Apply algorithm for solving problems like sorting, searching, insertion and deletion of data
- Ability to sensibly select appropriate data structures for problems and to justify that choice.

Unit I: Introduction: Definition of: Data, data objects, data types, data structure. Purpose of a data structure and implementation of data structure. Introduction to algorithms, properties of algorithms.
Arrays: Array as data structure, Storage representation of arrays, polynomial representation using arrays, sparse matrix representation. Applications of arrays. **(12)**

Unit II: Searching: Introduction, Searching Techniques: linear and binary search. Hashing, hash functions, collision and collision resolution.
Sorting: Introduction, insertion sorts, selection sorts, bubble sorts, quick sort, radix sort, enumeration sort and merge sort. **(12)**

Unit III: Lists: Linear list concepts-single and doubly linked list, circular lists, applications, operations on linked list.
Stacks: Introduction, operations on stack, static and dynamic implementation, application of stack: recursion, prefix, infix, and postfix expressions. **(12)**

Unit IV: Queues: Introduction, operations on Queue, static and dynamic implementation, Types of Queue: circular, priority, dequeue. Application of Queue: job scheduling.
Trees: Tree terminology, Binary tree, Binary tree representation, introduction to binary search tree (BST). **(12)**

Unit V: Operation on BST: creation, traversal, insertion and deletion in binary tree, binary tree sort. Balanced Tree: Introduction, AVL tree, height balance in AVL trees. B-tree, insertion and deletion into a B-tree.

Graphs: Terminology, representation, traversal, operations and applications: single source shortest path problem, minimum spanning tree. **(12)**

Suggested List of Lab Exercises:

1. Sorting programs: Bubble sort, Merge sort, Insertion sort, Selection sort, and Quick sort.
2. Searching programs: Linear Search, Binary Search.
3. Array implementation of Stack, Queue, Circular Queue, Linked List.
4. Implementation of Stack, Queue, Circular Queue, Linked List using dynamic memory allocation.
5. Evaluation of arithmetic expression
6. Implementation of Binary tree.
7. Program for Tree Traversals (preorder, inorder, postorder).
8. Program for graph traversal (BFS, DFS).
9. Write a program to perform the following operations:
   a) Insert an element into a AVL tree.
   b) Delete an element from a AVL tree.
   c) Search for a key element in a AVL tree.

Reference Books:

1. Robert L. Kruse, Bruce P. Leung, Clovis L. Tondo, *Data structures and program design in C*, BPB Publications, 2010.
2. Mark Allen Weiss, *Data structures and Algorithm Analysis in C*, Tata McGraw hill Publishing, 2nd edition, 2006.
3. Jean Paul Trembly and Soberson, *Introduction to Data Structures*, Tata McGraw hill Publishing, 2001.
4. Gilberg R.F, Forouzan B.A and Cengage, *Data Structures: A pseudocode approach withc++*, 2nd edition, 2005.
5. Dinesh P. Mehta, Sartaj Sahni, *Handbook of Data Structures and Applications*, Second Edition, CRC Press, 2018.

22MCS106 ADVANCED DATABASE MANAGEMENT SYSTEM

Total No. of Hours: 60 + 60 **Hours per week: L:T:P (4:0:4)**

Course Objectives:

- To provide overview of advancement in SQL and Database technology.
- To impart knowledge of query processing and optimization.
- To introduce the concept of document-oriented database.
- Understand the usage of advanced data models for real life application.

Learning Outcomes

On completion of this course, the student will be able to:

- Discuss new developments in database technology.
- Measure query cost and optimize query execution.
- Demonstrate the understanding of the concepts of document-oriented databases.
- Implement advanced data models for real life applications.

Unit I: Introduction: Purpose of Database Systems - View of Data - Database Languages - Data Storage and Querying - Database Users and Administrators. Database System Architecture: Database System Architectures, Distributed Databases, Parallel Databases. Data Models: Entity-Relationship Model, Relational Model.

Relational Databases: Introduction to the Relational Model - Structure of Relational Databases - Database Schema - Keys-Schema Diagrams - Functional Dependency – Normalization. Relational Query Languages - Relational operations. Advanced SQL - Accessing SQL from a Programming Language – Triggers. **(12)**

Unit II: Database Storage: File organization, Organization of records in files, Data Dictionary storage.
Indexing and Hashing: Basic Concepts, Ordered Indices, B+-Tree Index Files, Static Hashing, Dynamic Hashing.
Transaction Processing and Concurrency Control: Definition of Transaction and ACID properties; Concurrency Control Techniques: Lock based Concurrency control -Optimistic Concurrency Control – Timestamp based Concurrency Control, Deadlock Handling.
(14)

Unit III: Object-Based Databases: Object-Oriented Databases – Need for complex Data Types, Object-Oriented Data Model, Object identity OODBMS architecture and storage issues - Querying persistent objects.
XML Data Model: Structured, Semi structured, and Unstructured Data, XML Hierarchical Tree Data Model, XML Documents, DTD, and XML Schema, Storing and Extracting XML Documents from Databases - XML Languages, Extracting XML Documents from Relational Databases. **(12)**

Unit IV: Document Oriented Database: Need of Document Oriented database, Types of encoding - XML, JSON, BSON, Representation of JSON Objects.
Introduction to MongoDB, Storing data and accessing data from MongoDB, Querying MongoDB, Document store internals, MongoDB reliability and durability, Horizontal scaling, CRUD operations in MongoDB, Creating and using indexes in MongoDB.

(12)

Unit V: Advanced data models: Temporal data models: Aspects of valid time, bi-temporal time and bi-temporal time with examples of each.
Spatial model: Types of spatial data models - Raster, Vector and Image models.

(10)

Suggested List of Lab Exercises:

- Data modelling - E-R & Relational model
- Relational data model – working with SQL: DDL & DML commands
- Aggregate functions, Sub queries and Joins
- Concept of a view – Creation of views, operations
- Working with Stored Procedures, Triggers, Functions, and Cursors.
- Simulation of Query optimization
- Query execution on XML database
- Data handing using JSON
- Processing of Spatial and temporal data
- CRUD operations in MongoDB
- Implementation of Unstructured data like images and videos in MongoDB

Reference Books:

1. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Addison Wesley, Pearson Education, Seventh Edition, 2021
2. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Tata McGraw Hill, Sixth Edition, 2013.
3. Jeff A Hoffer, V Ramesh, HeikkiTopi, "Modern Database Management", Global Education, Twelfth Edition, 2016.
4. Kristina Chodorow, MongoDB, "The definitive Guide", O'Reilly, Third Edition, 2019.
5. Raghu Ramkrishnan, Johannes Gehrke, "Database Management System" Tata McGraw Hill. Third edition, 2020.