Second Year (Lateral Entry) Curriculum

Admission Year 2026-27

Bachelor of Technology (Lateral Entry) Information Technology

Faculty of Engineering & Technology

Parul University

Vadodara, Gujarat, India

Semester 3

a. **Course Name:** Database Information System
b. **Course Code:** 03010803PC05
c. **Prerequisite:** Basic Computer Knowledge
d. **Rationale:** The course will enable students to understand the different issues involved in the design and implementation of a database system as well execute various database queries using SQL.

e. **Course Learning Objective:**

CLOBJ 1	To explain database management system concepts, ER modeling techniques, and relational database design principles for structured data management.
CLOBJ 2	To apply SQL and PL/SQL constructs for database creation, querying, transaction handling, and practical data manipulation tasks.
CLOBJ 3	To analyze normalization, concurrency control, recovery mechanisms, and query optimization techniques for efficient and reliable database systems.
CLOBJ 4	To evaluate database security practices, modern NoSQL databases, and big data technologies for scalable and distributed data management solutions.

f. **Course Learning Outcomes:**

CLO 1	Describe the basic concepts of Database.
CLO 2	Explain Relational Models and its importance.
CLO 3	Build a proper structured database for a given problem or application.
CLO 4	Learn how various transactions are managed in real-time scenarios.
CLO 5	Demonstrate the evaluation parameters of a query as well as security parameters of the database.
CLO 6	Implement SQL concepts to build dynamic database applications.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	**Total**
3	**-**	**-**	**3**	**20**	**20**	**-**	**60**	**-**	**100**

L- Lectures; T- Tutorial; P- Practical; C- Credit; MSE- Mid-Semester Evaluation, CE-Continuous Evaluation, ESE- End Semester Examination

h. **Course Content:**

Sr. No.	**Content**	**Weightage**	**Teaching Hours**
1	**Database Fundamentals & ER Modeling** IDBMS vs file systems, ANSI/SPARC 3-schema architecture, data independence, client-server/distributed architecture, database users and DBA roles. ER Model: entities, attributes (simple/composite/derived), relationships, cardinalities, participation constraints, weak entities, extended ER (specialization/generalization/aggregation), ER diagrams, ER to relational mapping.	14%	7
2	**SQL Mastery & Practical Implementation**	18%	9

	SQL on MySQL/PostgreSQL/Oracle/SQL Server. DDL (CREATE/ALTER/DROP), data types, constraints (PK/FK/UNIQUE/CHECK). DML (INSERT/UPDATE/DELETE), SELECT with WHERE/ORDER BY/LIMIT. SQL functions (aggregate/string/date), GROUP BY/HAVING. Joins (INNER/LEFT/RIGHT/FULL/SELF), subqueries, set operations (UNION/INTERSECT/EXCEPT). Views, indexes (B-tree/composite), transactions. PL/SQL (stored procedures/functions/triggers/cursors). Window functions, CTEs, JSON in SQL.		
3	**Relational Design & Normalization** Relational model (relations/tuples/attributes), integrity constraints, Codd's rules, keys (superkey/candidate/primary/foreign). Relational algebra (selection/projection/join/division). Functional dependencies, Armstrong's axioms, attribute closure. Normalization: 1NF (atomic values), 2NF (no partial dependencies), 3NF (no transitive dependencies), BCNF, 4NF, 5NF. Lossless decomposition, denormalization trade-offs, schema design process.	18%	9
4	**Transactions, Concurrency & Recovery** Transaction basics, ACID properties (Atomicity/Consistency/Isolation/Durability). Schedules (serial/concurrent), serializability (conflict/view), precedence graphs. Two-Phase Commit (2PC). Recovery: log-based (WAL/undo/redo), shadow paging, checkpoints, ARIES algorithm. Concurrency control: lock-based protocols (2PL/strict 2PL), intention locks, timestamp ordering, MVCC. Deadlocks: prevention (Wait-Die/Wound-Wait), detection (wait-for graph), recovery. Isolation levels.	18%	9
5	**Query Optimization & Database Security** Query processing steps (parsing/optimization/execution), cost measurement (I/O/CPU/network). Query optimization (CBO/RBO), selection/join algorithms (nested-loop/sort-merge/hash join). Execution plans (EXPLAIN), indexing strategies (B-tree/B+/hash/bitmap), index tuning. Database security: authentication (MFA/SSO), authorization (RBAC), access control (DAC/MAC/RBAC). SQL injection prevention, encryption (TDE/TLS), data masking, auditing (GDPR/HIPAA/PCI-DSS). Backup strategies (full/incremental/differential), high availability (replication/failover).	16%	8
6	NoSQL, Big Data & Modern Database Systems NoSQL fundamentals, CAP theorem, BASE properties. Document stores: MongoDB (CRUD/aggregation pipeline/indexing/sharding). Key-value: Redis (data structures/caching/pub-sub). Column-family: Cassandra (CQL/partitioning). Graph: Neo4j (Cypher/traversal). Time-series: InfluxDB/TimescaleDB.	16%	8

i. Text Book and Reference Book:

1. Database System Concepts
   By Abraham Silberschatz, Henry Korth, S. Sudarshan | McGraw Hill International | 6th Edition
2. An Introduction to Database Systems
   By C. J. Date, A. Kannan, S. Swamynathan | Pearson Education | 8th Edition
3. SQL, PL/SQL – The Programming Language
   By Ivan Bayross | BPB Publications
4. The Definitive Guide to MongoDB", David Hows, Peter Membrey, Eelco Plugge, Tim Hawk, Apress, Second Edition.
5. Kevin Roebuck, "Storing and Managing Big Data - NoSQL, HADOOP and More", Emereopty Limited, ISBN: 1743045743, 9781743045749
6. Joy A. Kreibich, "Using SQLite", O'REILLY, ISBN: 13:978-93-5110-934-1

(2)

a. **Course Name:** Database Information System Laboratory
b. **Course Code:** 03010803PC06
c. **Prerequisite:** Basic Computer Knowledge
d. **Rationale:** The course will enable students to understand the different issues involved in the design and implementation of a database system as well execute various database queries using SQL.

e. **Course Learning Objective:**

CLOBJ 1	To create and manage relational database schemas using SQL commands, constraints, and normalization techniques.
CLOBJ 2	To apply SQL queries, joins, subqueries, aggregate functions, and data manipulation operations for efficient data retrieval and analysis.
CLOBJ 3	To demonstrate transaction control, access control, PL/SQL programming, and database management operations in practical scenarios.
CLOBJ 4	To develop problem-solving skills in database application development using filtering, grouping, multi-table operations, and procedural database programming concepts.

f. **Course Learning Outcomes:**

CLO 1	Describe the basic concepts of Database.
CLO 2	Explain Relational Models and its importance.
CLO 3	Build a proper structured database for a given problem or application.
CLO 4	Learn how various transactions are managed in real-time scenarios.
CLO5	Demonstrate the evaluation parameters of a query as well as security parameters of the database.
CLO6	Implement SQL concepts to build dynamic database applications.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
-	**-**	**2**	**1**	**-**	**-**	**20**	**-**	**30**	**50**

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**-Continuous Evaluation, **ESE-** End Semester Examination

h. **Course Content:**

Sr. No.	**Content**
1	**To study DDL-create and DML-insert commands.** **Schema** : DEPOSIT: (ACTNO, CNAME, BNAME, AMOUNT, ADATE) BRANCH: (BNAME, CITY) CUSTOMERS: (CNAME, CITY) BORROW: (LOANNO, CNAME, BNAME, AMOUNT)

	From the above given schema perform the following queries: 1 . Define & Populate: Create the four tables using VARCHAR2, NUMBER, and DATE types and insert the specified records. 2 . Metadata Check: Use DESC to verify the structure of DEPOSIT and BORROW. 3 . Data Retrieval: List all records from CUSTOMERS and BRANCH, and specific columns (ACTNO, AMOUNT) from DEPOSIT. 4 . Conditional Filtering: Filter depositors by AMOUNT > 4000 and ADATE > '1-12-96'.
2	**To study various options of LIKE predicate and advanced filtering commands.** **Schema**: JOB: (JOB_ID, JOB_TITLE, MIN_SAL, MAX_SAL) EMPLOYEE: (EMP_NO, EMP_NAME, EMP_SAL, EMP_COMM, DEPT_NO) *(Note: This practical also utilizes the DEPOSIT table from Practical 1)* **From the above given schema perform the following queries:** 1 . Define & Populate: Create JOB and EMPLOYEE tables with specified constraints and insert the provided records. 2 . Data Retrieval: List all records from EMPLOYEE, JOB, and the existing DEPOSIT table. 3 . Advanced Filtering: Filter DEPOSIT using BETWEEN for dates, JOB where MIN_SAL > 4000, and EMPLOYEE using IN for departments (10, 20). 4 . Aliasing: Display employee name and salary for Department 20, assigning a custom Alias to the name column. 5 . Pattern Matching: Use the LIKE predicate to filter employees based on name patterns (e.g., starting with 'A', specific lengths, character positions) and handle NULL values. 6 . Escape Characters: Demonstrate the usage of the ESCAPE clause within a LIKE query to handle special symbols.
3	**To Perform various data manipulation commands, aggregate functions and sorting concepts on all created tables.** **Schema**: DEPOSIT: (ACTNO, CNAME, BNAME, AMOUNT, ADATE) BORROW: (LOANNO, CNAME, BNAME, AMOUNT) CUSTOMERS: (CNAME, CITY) EMPLOYEE: (EMP_NO, EMP_NAME, EMP_SAL, EMP_COMM, DEPT_NO) SUPPLIER: *(Created from EMPLOYEE)* SUP1/SUP2: *(Created/Modified during practical* **From the above given schema perform the following queries:** 1 . Aggregate Analysis: Calculate total SUM of deposits/loans, find MAX loan amounts, and COUNT total customers and unique cities using DISTINCT. 2 . Table Creation: Create new tables (SUPPLIER, SUP1, SUP2) from existing data using CREATE TABLE AS SELECT with specific column filters or empty structures. 3 . Data Insertion: Populate SUP2 from EMPLOYEE based on specific string patterns (length 5, 2nd char 'n'). 4 . Data Deletion: Perform DELETE operations to remove all rows from SUP1 or specific suppliers (ID 103).

	5 . Schema Modification: RENAME table SUP2 and DROP table SUP1 entirely. 6 . Data Updates: Update DEPT_NO and EMP_NAME based on pattern matching (2nd char 'm') or specific IDs.
	To study Single-row functions **Schema**: EMPLOYEE: (EMP_NO, EMP_NAME, JOB, SAL, COMM, HIREDATE) CUSTOMERS / CLIENTS: (CLIENT_NAME, CITY, STATE, PINCODE, BAL_DUE) BORROW: (LOANNO, BNAME) SAILORS / BOATS: (SNAME, BNAME) (Auxiliary tables for specific pattern queries) **From the above given schema perform the following queries:** 1 . System Date: Write a query to display the current date with the label "Date". 2 . Numeric Calculation: Calculate a 15% salary increase (rounded to whole number) for all employees, labeled "New Salary". 3 . Arithmetic Operation: Modify the previous query to display the difference between the new and old salary, labeled "Increase". 4 . String Formatting: Display names in Title Case (Initcap) and their lengths for employees whose names start with 'J', 'A', or 'M', sorted by name. 5 . Concatenation: Generate a report sentence for each employee: *"<Name> earns <Salary> monthly"*. 6 . Date Math: Display name, hire date, total months employed, and the specific day of the week the employee started (sorted by day). 7 . Date Formatting: Display hire dates in the specific text format: *'Seventh of June 1994 12:00:00 AM'*. 8 . NULL Handling: Calculate annual compensation (SAL + COMM), ensuring NULL values are handled correctly. 9 .Advanced Pattern Matching (LIKE): ● Prefix/Suffix: Names starting with 'M', ending with 'L', or addresses starting with 'S'. ● Position Specific: Names with 'h' as the 3rd character (__h%) or 'a' as the 2nd character (_a%). ● Length Constraints: Names with a minimum of 6 characters (_____%). Complex Patterns: States having 'a' as the 4th or 5th character, or Pincodes starting with '4'.
	Displaying data from Multiple Tables (join) **Schema**: EMPLOYEE: (EMP_NO, EMP_NAME, JOB, MGR, HIREDATE, DEPT_NO) DEPARTMENT: (DEPT_NO, DNAME, LOC) CUSTOMERS / DEPOSIT / BORROW: (From Practical 1) **From the above given schema perform the following queries:** 1 . Simple Filter: Retrieve all details for the customer 'ANIL'. 2 . Intersection (Join): List names of customers who are both depositors and borrowers and reside in the city 'Nagpur'. 3 . Equi-Join (Branch-City): List the cities of customers where the customer's city matches their branch's city. 4 . Department Join: Display LAST_NAME, DEPT_NO, and DNAME for all employees by joining the Employee and Department tables. 5 . Unique Listing: Create a unique list of jobs in Department 30, including the department's

	location in the output. 6 . Location Filter: Display EMP_NAME, DEPT_NO, and DNAME for all employees working in 'NEW YORK'. 7 . Self-Join (Hierarchy): Display the employee's name and number alongside their manager's name and number. Label columns as *Employee*, *Emp#*, *Manager*, and *Mgr#*. 8 . Conditional Self-Join: Display the name and hire date of all employees hired after the employee 'SCOTT'.
	To apply the concept of Aggregating Data using Group functions. **Schema**: DEPOSIT: (ACTNO, CNAME, BNAME, AMOUNT, ADATE) EMPLOYEE: (EMP_NO, EMP_NAME, JOB, SAL, HIREDATE, DEPT_NO) BRANCH: (BNAME, CITY) **From the above given schema perform the following queries:** 1 . Totals: Calculate total deposits for customers with accounts opened after '1-JAN-96' or living in 'Nagpur'. 2 . Max Value: Find the maximum deposit amount for customers in 'Bombay'. 3 . Salary Stats: Display Max, Min, Sum, and Avg salary for all employees (rounded). 4 . Salary Range: Calculate the difference between the highest and lowest salaries. 5 . Pivot/Count: Count total employees and specific counts for those hired in 1995, 1996, 1997, and 1998. 6 . Group Averages: Find the average salary for each department where the average > 2000. 7 . Job Totals: Display total salary paid to each Job Title within every Department. 8 . Filter Groups: List Job Titles with total salary > 3000 (excluding 'PRESIDENT'), sorted by amount. 9 . Branch Sums: List 'Bombay' branches with total deposits exceeding 5000.
	To solve queries using the concept of sub query. 1 . Department Match: Find employees working in the same department as 'SCOTT' (excluding Scott). 2 . Location Match: List customers who are depositors and live in the same city as 'Sunil' or 'Pramod'. 3 . Above Average: List employees earning more than the company's average salary. 4 . Complex Filter: Find depositors living in the same city as 'Anil' with a deposit > 2000. 5 . Hierarchy: Display name and salary of all employees reporting to manager 'FORD'. 6 . Dept Lookup: Display details for all employees in the 'ACCOUNTING' department (using Dept Name). 7 . Max Aggregates: ● Find the branch with the highest number of depositors. ● Find the city with the maximum number of branches. Find customers in the city with the maximum number of depositors.
	Manipulating Data 1 . Interest Updates: Update deposits with 10% interest for: ● All depositors. ● 'VRCE' branch depositors only.

	● 'Nagpur' residents with 'Bombay' branches. 2 . Dept Transfer: Change Department ID of employees matching 7788's job to 7844's current department. 3 . Money Transfer: Transfer 10 Rs from Anil to Sunil if they share the same branch. 4 . Bonus: Add 100 Rs to depositors holding the maximum amount in their respective branch. 5 . Conditional Deletes: ● Delete depositors from branches with 1–3 customers. ● Delete all deposits for customer 'VIJAY'. ● Delete borrowers from branches with average loan < 1000.
	TCL and DCL Commands in SQL 1 . TCL: Demonstrate COMMIT (save), ROLLBACK (undo), and SAVEPOINT (partial undo). 2 . DCL: Demonstrate GRANT (give permission) and REVOKE (remove permission) for a user.
	PL/SQL Block: 1 . Arithmetic: Add two numbers using variables. 2 . Geometry: Calculate Area of Rectangle, Triangle, and Square. 3 . Max Value: Find the Maximum of 3 numbers. 4 . Looping: Calculate the Sum of N numbers using a FOR loop. 5 . Sequence: Generate the Fibonacci series up to N terms.

(3)

a. **Course Name:** Data Structure and Algorithms
b. **Course Code:** 03010803PC01
c. **Prerequisite:** Computer Programming and Basic Syntax
d. **Rationale:** Data structure is a subject of primary importance in Information and Communication Technology. Organizing or structuring data is important for implementation of efficient algorithms and program development. Efficient problem solving needs the application of appropriate data structure during program development.
e. **Course Learning Objective:**

CLOBJ 1	To explain fundamental data structures, dynamic memory allocation concepts, and algorithm performance analysis techniques.
CLOBJ 2	To apply linked lists, stacks, queues, recursion, and hashing techniques for solving computational problems efficiently.
CLOBJ 3	To analyze searching and sorting algorithms based on their operational behavior and time-space complexity.
CLOBJ 4	To implement tree and graph data structures along with traversal techniques for efficient data representation and problem solving.

f. **Course Learning Outcomes:**

CLO 1	Summarize the basic concepts of data structures.
CLO 2	Explain the linear & dynamic allocation of memory using linked list and its operations.
CLO 3	Illustrate stacks, queues, and recursion represented in memory for real-time applications.
CLO 4	Discuss static and dynamic hashing functions and sorting and searching.
CLO5	Analyze the concepts of B-trees, AVL tree and Red-Black tree.
CLO6	Describe the shortest path algorithms on graph data structure.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
3	-	-	3	20	20	-	60	-	100

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

h. **Course Content:**

Sr. No.	**Content**	**Weightage**	**Teaching Hours**
1. 1	**Introduction:** Techniques of Problem Solving, Data Structures, Types of Data Structures (Primitive, Non-Primitive, Linear & Non- Linear), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, dynamically allocated arrays. Performance analysis of an algorithm and space and time complexities. Real time examples.	**15%**	**8**

2. 2	**Linked Lists:** Definition, Representation of linked lists in Memory. Linked list operations: Traversing, Searching, Insertion and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists.	**12%**	**6**
3. 3	**Stack, Recursion and Queue:** **Stack:** Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. **Recursion:** Factorial, GCD, Fibonacci Sequence, Tower of Hanoi **Queue:** Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeue, Priority Queues and its problems.	**25%**	**12**
4. 4	**Hashing, Searching and Sorting:** **Hashing:** Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. **Searching and Sorting:** Interpolation Search, Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, Radix Sort.	**10%**	**3**
5. 3	**Trees:** Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals: In-Order, Post-Order, Pre-Order; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Red Black Trees and AVL Trees.	**25%**	**12**
6.	**Graphs:** Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search; Selection of Data Structures.	**13%**	**4**

i. Text Book and Reference Book:

1. **An Introduction to Data Structures with Applications (TextBook)**
   By Jean-Paul Tremblay, Paul G. Sorenson | Tata McGraw-Hill | 2nd Edition, (2007)
2. **C & Data Structures - P. Padmanabham, Third Edition, B.S Publications (Text Book)**
3. **Advanced data structure**
   By Peter Brass | Cambridge University Press
4. **Classic Data Structures**
   By Samanta, D. | PHI Learning, New Delhi
5. **Computer Programming & Data Structures - E. Balaguruswamy,4th Edition TMH (Text Book)**
6. **Data Structures and Algorithm Analysis in C++**
   By Mark Allen Weiss | Pearson | 2nd Edition, Pub. Year 2004

(4)

a. **Course Name:** Data Structure and Algorithms Laboratory
b. **Course Code:** 03010803PC02
c. **Prerequisite:** Basic knowledge of Data Structures
d. **Rationale:** This course provides a broad introduction to Data Structures. The various Data structures and its analysis of working design and development.

e. **Course Learning Objective:**

CLOBJ 1	To implement linear data structures such as stacks, queues, and linked lists along with their fundamental operations.
CLOBJ 2	To apply stack-based techniques for expression conversion, expression evaluation, and recursive problem solving.
CLOBJ 3	To develop searching and sorting algorithms for efficient data organization and retrieval.
CLOBJ 4	To implement tree and graph data structures with traversal algorithms for solving complex computational problems.

f. **Course Learning Outcomes:**

CLO 1	Compare single and double linked list operations.
CLO 2	Apply stack and queue for real-world problems.
CLO 3	Analyze binary search tree, traversals pre-order, in-order, post-order and its operations.
CLO 4	Illustrate Bubble sort, selection sort, Insertion sort, quick sort, and merge sort.
CLO 5	Demonstrate graphs, adjacency list, adjacency matrix and basic operations with traversals.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
0	-	2	1	-	-	20	-	30	50

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

h. **List of Practical:**

Exp. No.	**Name of the Experiment**
1	**Implement Stack and its operations like (creation push pop traverse peek search) using linear data structure**
2	**Implement Infix to Postfix Expression Conversion using Stack**
3	**Implement Postfix evaluation using Stack.**
4	**Implement Towers of Hanoi using Stack.**
5	**Implement queue and its operations like enqueue, dequeue, traverse, search.**
6	**Implement Single Linked lists and its operations (creation insertion deletion traversal search reverse)**
7	**Implement circular Linked lists and its operations (creation insertion deletion traversal search reverse)**
8	**Implement Doubly linked lists and its operations (creation insertion deletion traversal search reverse)**
9	**Implement binary search and interpolation search.**
10	**Implement Bubble sort, selection sort, Insertion sort, quick sort, merge sort.**
11	**Implement Binary search Tree and its operations (creation, insertion, deletion).**
12	**Implement Traversals Preorder, Inorder, Postorder on BST.**
13	**Implement Graphs and represent using adjaceny list and adjacency matrix and implement basic operations with traversals (BFS and DFS).**

(5)

a. **Course Name:** Discrete Mathematics
b. **Course Code:** 03019103BS01
c. **Prerequisite:** Basic set theory and elementary programming logic.
d. **Rationale:**

e. **Course Learning Outcomes:**

CLO 1	Apply set theory, relations and functions to model computational problems.
CLO 2	Use induction, recursion and counting techniques to analyze algorithms.
CLO 3	Construct valid logical arguments and mathematical proofs.
CLO 4	Identify and apply algebraic and Boolean structures in computing systems.
CLO 5	Analyze graphs and trees to solve problems related to connectivity and optimization

f. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
3	-	-	3	20	20		60	-	100

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

g. **Course Content:**

Sr. No.	**Content**	**Weightage**	**Teaching Hours**
1	**Sets, Relations and Functions** Sets and Operations on Sets Finite and Infinite Sets Countable and Uncountable Sets Cantor's Diagonal Argument Power Set Theorem Cartesian Products Relations: Reflexive, Symmetric, Transitive Equivalence Relations and Partial Order Relations Functions: One–One, Onto and Bijective Schroeder–Bernstein Theorem (statement and intuition)	**20%**	**9**
2	**Mathematical Induction, Recursion and Counting Techniques** Principle of Mathematical Induction Well Ordering Principle Recursive Definitions Recurrence Relations (linear recurrences, basic solutions) Division Algorithm Prime Numbers Euclidean Algorithm for GCD Fundamental Theorem of Arithmetic Counting Techniques: Inclusion–Exclusion Principle	**20%**	**9**

	Pigeonhole Principle Permutations and Combinations		
3	**Propositional Logic and Proof Techniques** Propositional Logic: Syntax and Semantics Truth Tables and Logical Connectives Validity and Satisfiability Logical Equivalence and Laws of Logic Rules of Inference Predicate Logic and Quantifiers Proof Techniques: Direct Proof Proof by Contradiction Proof by Contraposition Proof of Necessity and Sufficiency Logic in Program Reasoning: Preconditions and Postconditions Introduction to Loop Invariants	**20%**	**9**
4	**Algebraic Structures and Boolean Algebra** Algebraic Structures with One Binary Operation: Semigroups Monoids Groups (basic properties) Subgroups and Normal Subgroups Cyclic and Permutation Groups (introductory) Congruence Relations Lattices: Definition, Properties, Hasse Diagrams Boolean Algebra and Boolean Rings Identities and Principle of Duality Boolean Functions Disjunctive Normal Form (DNF) Conjunctive Normal Form (CNF)	**18%**	**8**
5	**Graph Theory and Trees** Graphs: Definitions and Properties Degree, Paths, Cycles, Connectivity Subgraphs and Graph Isomorphism Eulerian and Hamiltonian Graphs Graph Coloring: Vertex Coloring Edge Coloring Map Coloring Planar Graphs and Perfect Graphs Trees and Rooted Trees Weighted Trees Prefix Codes (Huffman Coding) Graph Algorithms: Breadth First Search (BFS) Depth First Search (DFS) Shortest Path Concepts	**22%**	**10**

h. Text Book and Reference Book:

1. Discrete Mathematics and its Applications (TextBook)
   By Kenneth H. Rosen | Tata McGraw – Hill
2. Discrete Mathematical Structure and It's Application to Computer Science (TextBook)
   By J.P. Tremblay and R. Manohar | TataMcgraw-Hill | TMG
3. Graph Theory with Applications to Engineering and Computer Science (TextBook)
   By Narsingh Deo | PHI
4. Elements of Discrete Mathematics A Computer Oriented Approach
   By C. L. Liu and D P Mohapatra | Tata McGraw – Hill | 3
5. Discrete Mathematics with Applications
   By Susanna S. Epp | Wadsworth Publishing Co. Inc. | 4

(6)

a. **Course Name:** Python Programming
b. **Course Code:** 03010803PC03
c. **Prerequisite:** Basic knowledge of Computer and programming.
d. **Rationale:** To introduce students to Python programming concepts from basic to advanced. To develop problem solving skills using Python programming constructs. To implement object-oriented programming, file handling, and exception handling. To familiarize with data processing using libraries like NumPy, Pandas, and Matplotlib.

e. **Course Learning Objective:**

CLOBJ 1	To explain Python programming fundamentals, control structures, operators, functions, and data types for problem solving.
CLOBJ 2	To apply Python data structures such as strings, lists, tuples, sets, and dictionaries in program development.
CLOBJ 3	To develop object-oriented Python programs using classes, inheritance, polymorphism, and operator overloading concepts.
CLOBJ 4	To implement file handling, exception handling, and Python libraries such as NumPy, Pandas, and Matplotlib for data processing and visualization.

f. **Course Learning Outcomes:**

CLO 1	Explain the basic concepts of Python programming, including data types, operators, control structures, and syntax rules.
CLO 2	Develop modular and efficient programs using functions, recursion, and built-in data structures like lists, tuples, sets, and dictionaries.
CLO 3	Apply object-oriented programming concepts such as classes, objects, inheritance, and polymorphism to design structured applications.
CLO 4	Implement file handling mechanisms and apply exception handling techniques to build robust and error-free programs.
CLO 5	Analyze and solve data processing problems using Python libraries such as NumPy, Pandas, and Matplotlib for scientific computing and data visualization.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
3	**-**	**-**	**3**	**20**	**20**	**-**	**60**	**-**	**100**

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**-Continuous Evaluation, **ESE-** End Semester Examination

h. Course Content:

Sr. No.	Content	Weightage	Teaching Hours
1	**Introduction to Python** Python environment setup, IDEs (IDLE, PyCharm, Jupyter), Keywords, Identifiers, Variables, Data types, Type conversions, Operators and Expressions, Conditional Statements (if, if-else, nested if), Looping (for, while, nested loops), Break, continue, pass statements.	**15%**	**5**
2	**Functions, Strings, Lists, Tuples, Sets, and Dictionaries** Defining Functions, Function Arguments, Recursion, Anonymous Functions (Lambda), String operations and built-in string methods, List operations and list comprehensions, Tuple, Set, Dictionary operations.	**15%**	**5**
3	**Object-Oriented Programming in Python** Object-Oriented Programming in Python Introduction to classes and objects, Constructors and destructors, Data hiding, encapsulation, Inheritance and polymorphism, Operator overloading	**20%**	**6**
4	**File Handling and Exception Handling** Reading from and writing to text files, working with file methods, Pickling objects, Error handling: try-except, else-finally, raising exceptions, User-defined exceptions.	**25%**	**7**
5	**Advanced Python and Libraries** Modules and Packages, Working with NumPy arrays and operations, Introduction to Pandas: Data Frame creation and manipulation, Basic Data Visualization using Matplotlib, Handling JSON data.	**25%**	**7**

i. Text Book and Reference Book:

1. Python Programming (TextBook)
   By by Reema Thareja | Oxford University Press
2. Introducing Python (TextBook)
   By Lubanovic Bill, O' ReILLY
3. PYTHON: The Complete Reference
   By Martin C. Brown | McGraw-Hill, 2001
4. Python Programming: An Introduction to Computer Science
   By John Zelle | Course Technology Cengage Learning Publications | 2nd, Pub. Year 2013

(7)

a. **Course Name:** Python Programming Laboratory
b. **Course Code:** 03010803PC04
c. **Prerequisite:** Basic knowledge of Computer and programming.
d. **Rationale:** To introduce students to Python programming concepts from basic to advanced. To develop problem solving skills using Python programming constructs. To implement object-oriented programming, file handling, and exception handling. To familiarize with data processing using libraries like NumPy, Pandas, and Matplotlib.

e. **Course Learning Objective:**

CLOBJ 1	To explain Python programming fundamentals, control structures, operators, functions, and data types for problem solving.
CLOBJ 2	To apply Python data structures such as strings, lists, tuples, sets, and dictionaries in program development.
CLOBJ 3	To develop object-oriented Python programs using classes, inheritance, polymorphism, and operator overloading concepts.
CLOBJ 4	To implement file handling, exception handling, and Python libraries such as NumPy, Pandas, and Matplotlib for data processing and visualization.

f. **Course Learning Outcomes:**

CLO 1	Explain the basic concepts of Python programming, including data types, operators, control structures, and syntax rules.
CLO 2	Develop modular and efficient programs using functions, recursion, and built-in data structures like lists, tuples, sets, and dictionaries.
CLO 3	Apply object-oriented programming concepts such as classes, objects, inheritance, and polymorphism to design structured applications.
CLO 4	Implement file handling mechanisms and apply exception handling techniques to build robust and error-free programs.
CLO 5	Analyze and solve data processing problems using Python libraries such as NumPy, Pandas, and Matplotlib for scientific computing and data visualization.

g. **Text Book and Reference Book:**

5. Python Programming (TextBook)
   By by Reema Thareja | Oxford University Press
6. Introducing Python (TextBook)
   By Lubanovic Bill, O' ReILLY
7. PYTHON: The Complete Reference
   By Martin C. Brown | McGraw-Hill, 2001
8. Python Programming: An Introduction to Computer Science
   By John Zelle | Course Technology Cengage Learning Publications | 2nd, Pub. Year 2013

h. List of Practical:

Exp. No.	Name of the Experiment
1	Install Python and set up different IDEs. Write a program for basic arithmetic operations.
2	Write programs using conditional and loop constructs.
3	Develop programs using functions, including recursion and lambda functions.
4	Develop programs using String manipulations and regular expressions.
5	Write Programs on lists, tuples, sets, and dictionaries.
6	Create classes and implement OOP concepts (inheritance, polymorphism).
7	Read from and write to text files, perform file operations.
8	Implement exception handling in a program.
9	Perform array manipulations using NumPy.
10	Create simple graphs using Matplotlib and handle CSV files using Pandas.

Semester 4

(1)

a. **Course Name:** Probability Statistics and Numerical Methods
b. **Course Code:** 03019104BS01
c. **Prerequisite:** Basic concepts of Statistics and Probability.
d. **Rationale:** The course provides systematic knowledge of probability, numerical and statistical methods.

e. **Course Learning Outcomes:**

CLO 1	Analyse correlation and regression between two variables and fit a curve to the given set of values.
CLO 2	Calculate probabilities and analyse random variables to determine expectation and variance.
CLO 3	Evaluate hypotheses by conducting significance tests for proportions, means, standard deviations, and variances using large sample tests, chi-square tests, and other appropriate statistical methods.
CLO 4	Apply numerical methods such as Gauss-Jacobi, Gauss Seidel, bisection method, Newton-Raphson method, and Regula-Falsi method to solve systems of linear equations and algebraic/transcendental equations
CLO 5	Interpolate data using finite differences and various interpolation techniques including Newton's forward/backward difference formulae, and Lagrange's formulae for unequal intervals.
CLO 6	Utilize numerical integration techniques such as the trapezoidal rule, Simpson's rules, and Gaussian quadrature formulae, as well as numerical methods including Taylor's series, Euler's method, Modified

f. **Teaching & Examination Scheme:**

Teaching Scheme				Evaluation Scheme					
				Internal Evaluation			ESE		Total
L	T	P	C	MSE	CE	P	Theory	P	
4	-	-	4	20	20	-	60	-	100

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

g. **Course Content:**

Sr. No.	Content	Weightage	Teaching Hours
1	**UNIT 1Correlation, Regression and Curve fitting:** Correlation and Regression – Rank correlation Curve Fitting by The Method of Least Squares- Fitting of Straight Lines, Second Degree Parabolas and More General Curves.	**18%**	**11**
2	**UNIT 2Probability and Probability Distributions:** Probability Spaces, Conditional Probability, Bayes' Rule, Discrete and continuous random variables, Independent Random Variables, Expectation and Variance of Discrete and Continuous Random Variables, Distribution and Their Properties: Binomial	**23%**	**13**

	Distribution, Poisson Distribution, Normal Distribution.		
3	**UNIT 3Testing of Hypothesis:** Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations. Test for single mean, difference of means, Test for ratio of variances, Chi-square test for goodness of fit and independence of attributes.	**26%**	**15**
4	**UNIT 4Solution of a System of Linear Equations, Roots of Algebraic and Transcendental Equations:** Gauss-Jacobi and Gauss Seidel Methods, Solution of Polynomial and Transcendental Equations – Bisection Method, Newton-Raphson Method and Regula-Falsi Method.	**11%**	**7**
5	**UNIT 5Finite Differences and Interpolation:** Finite Differences, Relation between Operators, Interpolation using Newton's Forward and Backward Difference Formulae. Newton's Divided and Lagrange's Formulae for Unequal Intervals.	**11%**	**7**
6	**UNIT 6Numerical Integration:** Trapezoidal rule, Simpson's 1/3rd and 3/8th Rules, Gaussian Quadrature Formulae. **Numerical solution of Ordinary Differential Equations:** Taylor's Series, Euler and Modified Euler's Methods. Runge- Kutta Method of Fourth Order for Solving First and Second Order Equations.	**11%**	**7**

h. Text Book and Reference Book:

1. Introductory Methods of Numerical Analysis
   By Sastry S. S | Prentice Hall of India
2. Introduction to Numerical Analysis
   By C.E. Froberg | Addison Wesley Publishing Company
3. Numerical Methods in Engineering & Science with Programs in C and C++ (TextBook)
   By Dr. B. S. Grewal | Khanna Publishers
4. Introduction to Probability (TextBook)
   By P. G. Hoel, S. C. Port and C. J. Stone, | UBS Publishers,
5. Fundamentals of Mathematical Statistics (TextBook)
   By S.C. Gupta and V. K. Kapoor | Sultan Chand & Sons

(2)

a. **Course Name:** Digital Electronics
b. **Course Code:** 3010703ES01
c. **Prerequisite:** Basic Electronics
d. **Rationale:** This course is design to provide basic ideas of computer architecture. This course also makes help to understand organization and architecture of computer. It will help to develop their logical abilities.
e. **Course Learning Outcomes:**

CLO 1	Identify and Explain the digital number system and also able to justify the practical application of number system.
CLO 2	Understand and Explain different logic gates and codes and also how to use them in real word application.
CLO 3	Realize the minimization techniques of digital Circuits.
CLO 4	Design different Adders, Subtracters, Multiplexers, decoders and many more circuits
CLO 5	Apply the theoretical knowledge to design flip-flops, counters and many more sequential circuits.
CLO 6	Identify and illustrate specifications of different logic families and memories and analyze them in critical way.

f. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
2	**-**	**-**	**2**	**20**	**20**	**-**	**60**	**-**	**100**

g. **Course Content:**

Sr. No.	Content	Weightage	Teaching Hours
1	**Fundamentals of Digital Systems and logic families:** Digital signals, digital circuits, Number Systems: binary, signed binary, octal, hexadecimal number, binary arithmetic, • ones and twos complements arithmetic, codes, BCD arithmetic ,error detecting and correcting codes, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, examples of IC gates, characteristics of digital lCs, Digital Logic families:TTL and CMOS logic, interfacing CMOS and TTL.	**15%**	**7**
2	**Minimization Techniques:**Boolean Algebra, Boolean postulates and laws, De-Morgans Theorem, Principle of Duality, Boolean expression, • Minterm, Maxterm, Sum of Products (SOP), Product of Sums (POS), K-map representation, simplification and minimization of logic functions using K-map. Don't care conditions and Quine-McCluskey Method of minimization. Variable Entered Maps, Realizing Logic Function with Gates.	**20%**	**8**
	Combinational Digital Circuits: Binary Adders and	**20%**	**9**

3	Subtractors, Parallel binary adder & subtractor, Serial adder, BCD adder, Carry look ahead adder, Multiplexer/De Multiplexer, Encoder/Decoders, Popular MSI chips, Magnitude comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices.		
4	**SEQUENTIAL CIRCUITS:**A 1-bit memory, the circuit properties of Bi-stable latch, the clocked SR flip flop, J- K-T and Dtypesflip flops, applications of flipflops, shift registers, Applications of shift registers, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters,special counter IĊs, asynchronous sequential counters, applications ofcounters.	**20%**	**9**
5	**A/D and D/A Converters:**Digital to analog converters: weighted resistor/converter, R-2R Ladder, examples of D to A converters IĊs, Analog to • Digital converters: successive approximation, A/D converter, dual slope A/D Converter, Example of A/DConverterICs.	**10%**	**5**
6	**Semiconductor Memories And Programmable Logic Devices:**Classification and characteristics of memories, Content addressable memory (CAM), commonly used memory chips, Introduction of PLD,ROM as a PLD, Programmable logic array, Programmable array logic, Complex Programmable logic devices (CPLDS), Field Programmable Gate Array (FPGA)	**15%**	**7**

i. Text Book and Reference Book:

1. Modern Digital Electronics (TextBook)
   By R. P. Jain | Tata McGraw-Hill Education
2. Digital Logic and Computer Design
   By Morris Mano | PHI
3. Fundamentals of Digital Circuits
   By Anand Kumar | Prentice-Hall of India Private Limited, New Delhi (2006)

(3)

a. **Course Name:** Digital Electronics Laboratory
b. **Course Code:** 03010703ES02
c. **Prerequisite:** Basic Electronics
d. **Rationale:** This course is design to provide basic ideas of computer architecture. This course also makes help to understand organization and architecture of computer. It will help to develop their logical abilities.
e. **Course Learning Outcomes:**

CLO 1	Identify and Explain the digital number system and also able to justify the practical application of number system.
CLO 2	Understand and Explain different logic gates and codes and also how to use them in real word application.
CLO 3	Realize the minimization techniques of digital Circuits.
CLO 4	Design different Adders, Subtracters, Multiplexers, decoders and many more circuits
CLO 5	Apply the theoretical knowledge to design flip-flops, counters and many more sequential circuits.
CLO 6	Identify and illustrate specifications of different logic families and memories and analyze them in critical way.

f. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
0	0	2	1	0	0	20	0	30	50

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

g. **Text Book and Reference Book:**

4. Modern Digital Electronics (TextBook)
   By R. P. Jain | Tata McGraw-Hill Education
5. Digital Logic and Computer Design
   By Morris Mano | PHI
6. Fundamentals of Digital Circuits
   By Anand Kumar | Prentice-Hall of India Private Limited, New Delhi (2006)

h. **List of Practical:**

Exp. No.	**Name of the Experiment**
1	**To Study and Testing of various Logic Gates ICs.** To Study and Testing of various Logic Gates ICs.
2	**Configuring NAND and NOR logic gates as universal gates.** Configuring NAND and NOR logic gates as universal gates.
3	**Design Logic Gates using TTL Logic Gamily.** Design Logic Gates using TTL Logic Gamily.
4	**Study and Implementation of Boolean Logic Functions and combinational circuits like**

	Adder/ Subtractor, Code Converters, using Logic Gates. Study and Implementation of Boolean Logic Functions and combinational circuits like Adder/ Subtractor, Code Converters, using Logic Gates.
5	**Study and Implementation of Boolean Logic Functions and combinational circuits like Multiplexers/De-Multiplexres using Logic Gates.** Study and Implementation of Boolean Logic Functions and combinational circuits like Multiplexers/De-Multiplexres using Logic Gates.
6	**Study and Implementation of Boolean Logic Functions and combinational circuits like Encoders/ Decoders, using Logic Gates.** Study and Implementation of Boolean Logic Functions and combinational circuits like Encoders/ Decoders, using Logic Gates.
7	**Study and configure of flip-flop using digital ICs. Design digital system using these circuits.** Study and configure of flip-flop using digital ICs. Design digital system using these circuits.
8	**Study and configure of registers and counters using digital ICs. Design digital system using these circuits.** Study and configure of registers and counters using digital ICs. Design digital system using these circuits.
9	**Study and Design A to D / D to A converters.** Study and Design A to D / D to A converters.
10	**Introduction to FPGA / CPLD. Implementation of digital circuits studied in previous sessions using PLD/ CPLD / FPGA** Introduction to FPGA / CPLD. Implementation of digital circuits studied in previous sessions using PLD/ CPLD / FPGA

(4)

a. **Course Name:** Data Communication and Networking
b. **Course Code:** 03010804PC01
c. **Prerequisite:** knowledge of Computer and Hardware
d. **Rationale:** This course is designed to provide the basic knowledge of networking. It also provides a firm foundation for understanding how data communication occurs in different Transmission Medium. It will help to develop logical and practical abilities to set up the network.

e. **Course Learning Objective:**

CLOBJ 1	To explain the fundamentals of data communication, network architectures, transmission media, and networking devices in modern communication systems.
CLOBJ 2	To analyze application layer and transport layer protocols, network services, security mechanisms, and quality of service techniques.
CLOBJ 3	To apply IP addressing, subnetting, routing protocols, and network layer concepts for efficient data delivery across networks.
CLOBJ 4	To evaluate data link and physical layer protocols, error control mechanisms, multiple access techniques, and emerging networking technologies such as SDN and cloud networking.

f. **Course Learning Outcomes:**

CLO 1	Explain data communication principles, network components, and topologies to understand data flow and transmission media.
CLO 2	Apply data link layer protocols, error control mechanisms, and MAC techniques for efficient medium access.
CLO 3	Compare network layer addressing, routing techniques, and switching methods for effective packet delivery.
CLO 4	Examine transport layer protocols, congestion control strategies, and QoS metrics to optimize network performance.
CLO5	Apply application layer protocols and security mechanisms to enhance network communication and data protection.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
3	0	0	3	20	20	0	60	0	100

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination.

h. Course Content:

Sr. No.	Content	Weightage	Teaching Hours
1	**Data Communication and Network Components:** Representation of data and its flow - Data Transmission Modes, Network types and connection topologies, Protocols, Standards, and OSI Model, TCP/IP model, Example Networks- ARPANET, NSFNET, Architecture of Internet, Transmission Media, Wired and Wireless, Local Area Networks (LANs), Wireless LANs (WLANs), Virtual LANs (VLANs), Connecting devices- Repeaters, Hub, Bridges, Switch, Router, Gateways, Introduction to Wireless Sensor Networks (WSNs), Electronic Mail, World Wide Web	**15%**	**8**
2	**Application Layer Protocols and Security Mechanisms:** Domain Name System (DNS), Dynamic DNS (DDNS), TELNET, Email Protocols (SMTP, POP3, IMAP), File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP, HTTPS), Simple Network Management Protocol (SNMP), Introduction to Cloud Networking and Edge Computing, Network Security Concepts- Firewalls, Cryptography Basics	**20%**	**8**
3	**Transport Layer Protocols and Quality of Service:** Process-to-Process Communication **Transport Layer Protocols**: UDP, TCP, SCTP, Congestion Control Mechanisms (Leaky Bucket, Token Bucket) **Quality of Service**: QoS in Networking, Network Performance Metrics (Latency, Throughput, Jitter)	**20%**	**8**
4	**Network Layer Protocols and IP Addressing:** **Logical Addressing**: IPv4 Addressing Classes, Subnetting, CIDR (Classless Inter-Domain Routing), IPv6 Addressing: Address types (Unicast, Multicast, Anycast), Format, Notation, IPv6 Addressing and Transition Mechanisms, Comparison: IPv4 vs. IPv6, Address Mapping: ARP, RARP, BOOTP, DHCP, **Routing Concepts**: Delivery, Forwarding, Static & Dynamic Routing, Unicast, Multicast, and Broadcast Routing Protocols (RIP, OSPF, BGP)	**20%**	**10**
5	**Data Link Layer and Physical Layer Protocols:** Error Detection and Correction (Block Coding, Hamming Distance, CRC), Flow Control and Error Control Protocols: Stop & Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Transmission media (Twisted pair, Coaxial cable, Fiber optic cable),Multiple Access Techniques: ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA, MAC Addressing and Ethernet Standards (802.3, 802.11),Guided Transmission Media, Magnetic Media Bandwidth Utilization Techniques: Multiplexing (FDM, TDM, WDM), Switching Techniques (Circuit Switching, Packet Switching) Spread Spectrum Concepts (FHSS, DSSS), Introduction to Software-Defined Networking (SDN)	**25%**	**11**

i. Text Book and Reference Book:

1. “Computer Networks – A Systems Approach”
   By L. Peterson and B. Davie | Elsevier Morgan Kaufmann Publisher | 5th Edition
2. “Computer Networks and Internet/TCP-IP”
   By D. Comer | Prentice Hall
3. Computer Networks
   By Andrew S. Tanenbaum and David J. Wetherall | PEARSON Edition
4. Computer networks (TextBook)
   By Andrew S. Tanenbaum | Prentice Hall Publication | FIFTH EDITION

(5)

a. **Course Name:** Data Communication and Networking Lab
b. **Course Code:** 03010804PC02
c. **Prerequisite:** Knowledge of Computer and Hardware.
d. **Rationale: This course provides fundamental knowledge of computer networks, emphasizing data communication in transmission media. It builds a strong foundation in network flow and enhances practical skills through Cisco Packet Tracer for simulation and Wireshark for packet analysis, offering hands-on experience in network design, configuration, and troubleshooting.**

e. **Course Learning Objective:**

CLOBJ 1	To identify and configure network devices and network topologies using Cisco Packet Tracer.
CLOBJ 2	To design and implement VLANs, WLANs, VPNs, and routing protocols in simulated networking environments.
CLOBJ 3	To apply subnetting techniques and transport layer concepts for efficient network communication and management.
CLOBJ 4	To analyze network traffic and protocol behavior using packet capture and monitoring tools such as Wireshark.
CLOBJ 5	To identify and configure network devices and network topologies using Cisco Packet Tracer.

i. **Course Learning Outcomes:**

CLO 1	Analyze the behavior and functionality of network devices using Cisco Packet Tracer.
CLO 2	Design various network topologies for efficient communication.
CLO 3	Design VLANs, WLANs, and VPNs for secure network segmentation.
CLO 4	Configure routing techniques, including subnetting and inter-networking with routers.
CLO 5	Analyze network traffic using Wireshark for security threats and protocol behaviors.

j. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
0	0	2	1	0	0	20	0	30	50

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

k. Text Book and Reference Book:

1. Fluid Mechanics & Hydraulics Dr.R. K. Bansal; Laxmi Publications
2. Fluid Mechanics, By Yunus A. Cengel, John M. Cimbala | Tata McGraw Hill.
3. Introduction to Fluid Mechanics and Fluid Machines, By S. K. Som and G. Biswas | Tata McGraw Hill.
4. Fluid Mechanics and Fluid Power Engineering D.S. Kumar; S.K. Kataria & Sons
5. Engineering Fluid Mechanics, By K.L. Kumar | Eurasia Publication House

l. Experiment List:

Exp. No.	Name of the Experiment
1	**Study of network devices such as switches, routers, hubs, access points, and firewalls using cisco packet tracer.**
2	**Study and design of various network topologies using cisco packet tracer.**
3	**Study of HUB and switch behavior using cisco Packet Tracer.**
4	**Study and Configuration of Virtual LAN (VLAN) using Cisco Packet Tracer.**
5	**Study and Design of Wireless LAN (WLAN) using Cisco Packet Tracer.**
6	**Perform subnetting on** **A) Any of the IP Addresses of Class C** **B) Any of the IP Addresses of Class A**
7	**Study and implement routing protocols at the network layer in cisco packet tracer.**
8	**Study and configure Virtual Private Network (VPN) in cisco packet tracer.**
9	**Study experiment on transport layer using cisco packet tracer.**
10	**Packet Capture and Analysis Using Wireshark A: Network Traffic Monitoring - Inspect packet headers, and analyze data flow within a network. B: Capturing and Examining Protocol Headers- Analyze TCP/IP packet structures to identify traffic patterns and detect potential security threats.**

(6)

a. **Course Name:** Design and Analysis of Algorithm
b. **Course Code:** 03010504PC03
c. **Prerequisite:** Data structures, Fundamental of programming
d. **Rationale:** Analyze the asymptotic performance of algorithms. Write rigorous correctness proofs for algorithms. Demonstrate a familiarity with major algorithms and data structures. Apply important algorithmic design paradigms and methods of analysis. Synthesize efficient algorithms in common engineering design situations.

e. **Course Learning Objective:**

CLOBJ 1	To explain algorithm design techniques, asymptotic analysis, recurrence relations, and complexity measures for evaluating computational efficiency.
CLOBJ 2	To apply divide-and-conquer, greedy, and dynamic programming strategies for solving optimization and computational problems.
CLOBJ 3	To analyze graph traversal, backtracking, and branch-and-bound techniques for solving complex algorithmic challenges.
CLOBJ 4	To evaluate string matching algorithms and NP-complete problem concepts for advanced problem-solving and computational analysis.

f. **Course Learning Outcomes:**

CLO 1	Develop the ability to analyze the running time of any given algorithm using asymptotic analysis and prove the correctness of basic algorithms.
CLO 2	Design efficient algorithms for computational problems, using various algorithm design techniques taught in the course.
CLO 3	Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
CLO 4	Analyze String matching algorithms.
CLO 5	Explain the complexity classes P, NP, and NP-Complete, and demonstrate the NP-Completeness of a specific problems.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
3	-	0	3	20	20	-	60	-	100

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

h. Course Content:

Sr. No.	Content	Weightage	Teaching Hours
1	**Introduction and Analysis of Algorithms:** Algorithm: Definition, Properties, Types of Algorithms, Writing an Algorithm Analysis: Parameters, Design Techniques of Algorithms. Asymptotic Analysis: Big Oh, Big Omega & Big Theta Notations, Lower Bound, Upper Bound and Tight Bound, Best Case, Worst Case, Average Case Analyzing control statement, Loop invariant and the correctness of the algorithm, Recurrences- substitution method, recursion tree method, master method. Sorting Techniques with analysis: Bubble Sort, Selection Sort, Insertion sort.	**20%**	**10**
2	**Divide & Conquer Algorithms:** Structure of divide-and-conquer algorithms, examples: Binary search, quick sort, Merge sort**,** Strassen Multiplication; Max-Min problem	**20%**	**6**
3	**Greedy Algorithms:** Introduction, Elements of Greedy Strategy - Minimum Spanning Tree: Kruskal's & Prim's Algorithm, Dijkstra's Algorithm, Knapsack Problem, Activity Selection Problem, Huffman Codes	**20%**	**8**
4	**Dynamic Programming:** Principal of Optimality, 0/1 Knapsack Problem, Making Change problem, Chain matrix multiplication, Longest Common Subsequence, All pair shortest paths: Warshall's and Floyd's algorithms.	**20%**	**8**
5	**Exploring Graphs:** An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort.	**5%**	**3**
6	**Backtracking and Branch & Bound:** Introduction to Backtracking, Introduction to Branch & Bound, 0/1 Knapsack Problem, N-Queens Problem, Travelling Salesman Problem.	**5%**	**4**
7	**String Matching & NP Completeness:** **String Matching:** - Introduction to String Matching, Naive String Matching, Rabin-Karp Algorithm, Kruth-Morris-Pratt Algorithm, String Matching using Finite Automata **NP Completeness:** Introduction to NP Completeness, P class Problems, NP Class Problems, Hamiltonian Cycle	**10%**	**6**

i. Text Book and Reference Book:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill. (TextBook)
2. Fundamentals of Algorithms – E. Horowitz et al. (TextBook)
3. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson
4. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
5. Algorithms—A Creative Approach,3RD Edition, UdiManber, Addison-Wesley, Reading, MA

(7)

a. **Course Name:** Design and Analysis of Algorithms Lab
b. **Course Code:** 03010504PC04
c. **Prerequisite:** Design and Analysis of Algorithms (DAA) is crucial for efficient problem-solving and algorithm development. It provides tools to measure algorithm performance and make informed decisions on choosing the best algorithms for specific tasks. DAA helps optimize time and space complexities, leading to improved computational efficiency.

d. Course Learning Objective:

CLOBJ 1	To implement and compare algorithms based on empirical time complexity analysis and computational efficiency.
CLOBJ 2	To apply searching, sorting, greedy, and binary search techniques for solving optimization and array-based problems.
CLOBJ 3	To develop graph and grid-based solutions using traversal algorithms for connectivity, cycle detection, and shortest path problems.
CLOBJ 4	To solve dynamic programming and string manipulation problems using efficient algorithmic approaches.

e. Course Learning Outcomes:

CLO 1	Develop the ability to design and implement efficient algorithms for fundamental problems.
CLO 2	Cultivate critical thinking skills to analyze problem requirements and constraints, allowing for the selection and modification of appropriate algorithms to solve specific computational problems.
CLO 3	Master the use of essential data structures such as arrays, matrices, graphs, and trees to efficiently store, manage, and manipulate data within algorithm implementations.
CLO 4	Learn techniques for optimizing algorithms to improve their efficiency and scalability, focusing on aspects such as time complexity, and space complexity,

f. Teaching & Examination Scheme:

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
0	**-**	**2**	**1**	**-**	**-**	**20**	**-**	**30**	**50**

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

g. List of Practical:

Exp. No.	**Name of the Experiment**
1	Empirical Analysis of Time Complexity (O(n), O(n log n), O(n^2)) Write programs to compare execution time of the following algorithms: i) Linear search(O(n)) ii) Bubble sort(O(n2)) iii) Merge Sort(O(nlogn))

2	Given a sorted array and a target value,return the index if the target is found. If not,return the index where it would be if it were inserted in order.
3	There are N children standing in a line with some rating value. You want to distribute a minimum number of candies to these children such that: Each child must have at least one candy. The children with higher ratings will have more candies than their neighbours. You need to write a program to calculate the minimum candies you must give.
4	There is a new barn with N stalls and C cows. The stalls are located on a straight line at positions x1,xN (0 <= xi <= 1,000,000,000). We want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?
5	Given an undirected graph with V vertices and E edges, check whether it contains any cycle or not
6	There are n servers numbered from 0 to n – 1 connected by undirected server-to-server connections forming a network where connections[i] = [ai, bi] represents a connection between servers ai and bi. Any server can reach other servers directly or indirectly through the network.A critical connection is a connection that, if removed, will make some servers unable to reach some other servers. Return all critical connections in the network in any order.
7	Given a grid of size NxM (N is the number of rows and M is the number of columns in the grid) consisting of '0's (Water) and '1's(Land). Find the number of islands.
8	Given a grid of dimension N x M where each cell in the grid can have values 0, 1, or 2 which has e following meaning: 0: Empty cell, 1: Cells have fres, 2. Cells have oranrotten oranges We have to determine what is the minimum time required to rot all oranges. A rotten orange at index [i,j] can rot other fresh oranges at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (up, down, left and right) in unit time'
9	Given two strings str1 and str2 and below operations that can be performed on str1. Find minimum number of edits (operations) required to convert 'str1' into 'str2'. Insert Remove Replace, All of the above operations are of equal cost.
10	Minimum Path Sum" says that given a n x m grid consisting of non-negative integers and we need to find a path from top- left to bottom right, which minimizes the sum of all numbers along the path.
11	Given string num representing a non-negative integer num, and an integer k, return the smallest possible integer after removing k digits from num.
12	There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]).The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time. Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner.

(8)

a. **Course Name:** Programming in Python with Full Stack Development

b. **Course Code:** 03010804PE01

c. **Prerequisite:** Basic knowledge of Programming and web applications

d. **Rationale:** This course bridges the gap between core programming logic and high-performance backend engineering. By mastering Python, Django, and MongoDB, you will learn to architect secure, data-driven RESTful services and deploy scalable full-stack applications to the cloud.

e. **Course Learning Objective:**

CLOBJ 1	To apply Python programming fundamentals, data structures, and problem-solving techniques for software development.
CLOBJ 2	To develop modular and object-oriented Python applications using functions, classes, inheritance, and standard libraries.
CLOBJ 3	To implement database connectivity, file handling, exception handling, and CRUD operations using SQL and NoSQL databases.
CLOBJ 4	To design and deploy RESTful APIs and full-stack web applications using Flask, Django, and cloud deployment platforms.

f. **Course Learning Outcomes:**

CLO 1	Apply core Python syntax, data structures (Lists, Dicts, Comprehensions), and Object-Oriented principles to solve complex algorithmic problems and manage data effectively
CLO 2	Design and implement RESTful APIs using Flask or FastAPI, demonstrating proficiency in HTTP methods, JSON handling, and API versioning.
CLO 3	Integrate Python applications with both SQL (MySQL) and NoSQL (MongoDB) databases, utilizing Djongo ORM for efficient data modeling and CRUD operations.
CLO 4	Construct full-featured web applications using the Django framework, incorporating secure user authentication, URL routing, and dynamic template rendering.
CLO 5	Manage project versions using Git and execute end-to-end deployment of Python applications to cloud platforms like AWS, Heroku, or Render while ensuring security best practices.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
3	-	-	3	20	20	-	60	-	100

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

h. Course Content:

Sr. No.	Content	Weightage	Teaching Hours
1	**Python Fundamentals & Data Structures** Introduction to Python and applications, Python environment setup (venv, pip), Variables, data types, operators and expressions, Control flow (if-elif-else) and loops (for, while), Working with strings, lists, tuples, sets, and dictionaries, Built-in functions, comprehensions, Introduction to Git & GitHub (version control basics), Basic problem-solving and algorithmic thinking	**18%**	**8**
2	**Functions, OOP & Modular Programming** Functions (definition, parameters, return values, *args, **kwargs), Lambda and higher-order functions, Introduction to OOP concepts, Classes and objects, constructors (__init__, self), Encapsulation (private members, @property), Inheritance (single, multiple, super()), Polymorphism, Modules and packages (import, __init__.py), Python standard library overview, Practical OOP case studies	**17%**	**7**
3	**Database Connectivity & Data Management** File handling (text, CSV, JSON) and context managers, Exception handling (try-except-else-finally blocks), Custom exceptions, Introduction to databases (SQL vs NoSQL), MySQL connectivity and CRUD operations (mysql-connector-python), Parameterized queries and SQL injection prevention, MongoDB connectivity and CRUD operations (pymongo), Collections, queries, aggregation basics, Exception handling in database operations	**20%**	**9**
4	**RESTful API Development with Flask** RESTful architecture principles, HTTP methods and status codes, Introduction to Flask framework, Flask routes, request/response handling, Building REST APIs with endpoint design, JSON handling (request.get_json(), jsonify()), Flask-SQLAlchemy ORM integration, CRUD API with database, JWT authentication (Flask-JWT-Extended), API versioning and CORS, API documentation basics (Swagger/OpenAPI), Testing APIs with Postman and pytest basics	**22%**	**10**
5	**Django Full-Stack & Deployment** Django framework introduction (MVT architecture), Django installation and project structure, MongoDB integration with Djongo, Django models, field types, migrations, Django ORM (QuerySets, filtering, CRUD), Django Admin customization, Views (FBV, CBV), URL configuration, Template system (syntax, inheritance), Static and media files handling, Forms and validation, User authentication and authorization, Django REST Framework (DRF) basics (serializers, ViewSets), Deployment to cloud (AWS/Heroku/Render), Security best practices	**23%**	**11**

i. Text Book and Reference Book:

1. **Fluent Python, 2nd Edition by Luciano Ramalho (TextBook)**
2. **Learn Python3 the Hard Way By Zed Shaw**
3. **Django for Beginners: Build websites with Python and Django by William S. Vincent.**
4. **Learning Django Web Development by Samuli Natri.**
5. **Flask Web Development with Python by Miguel Grinberg.**
6. **Mastering Flask by Jack Stouffer.**
7. **Building RESTful Python Web Services by Gastón C. Hillar.**
8. **Building Web APIs with FastAPI by Samuel Colvin.**

(9)

a. **Course Name:** Programming in Python with Full Stack Development Laboratory
b. **Course Code:** 03010804PE02
c. **Prerequisite:** Basic knowledge of Programming and web applications.
d. **Rationale:** This course bridges the gap between core logic and professional backend engineering. By mastering Python, Django, and MongoDB, you will learn to architect secure RESTful services and deploy scalable, data-driven applications to the cloud.
e. **Course Learning Objective:**

CLOBJ1	To develop Python programs using control structures, data structures, functions, and object-oriented programming concepts.
CLOBJ 2	To implement file handling, exception handling, modules, and database connectivity using MySQL and MongoDB.
CLOBJ 3	To design web applications using Django with templates, forms, authentication, and CRUD operations.
CLOBJ 4	To develop, test, secure, and deploy RESTful API-based full-stack applications using Django REST Framework and cloud platforms.

f. **Course Learning Outcomes:**

CLO 1	**Procedural & OOP Proficiency:** Solve complex problems by applying Pythonic syntax, advanced data structures, and Object-Oriented principles such as inheritance and encapsulation.
CLO 2	**Database Integration & Management:** Design and implement robust data models using the Djongo ORM to perform complex CRUD operations on MongoDB databases.
CLO 3	**Secure Web Framework Architecture:** Construct dynamic web applications using the Django MVT (Model-View-Template) pattern, incorporating secure user authentication and authorization protocols.
CLO 4	**RESTful Service Engineering:** Design, build, and document scalable APIs using Django REST Framework, featuring JWT authentication and standard HTTP response handling.
CLO 5	**Full-Stack Deployment & DevOps:** Execute end-to-end project lifecycles by integrating frontend components with Python backends and deploying the unified system to cloud platforms like Heroku or Render.

g. **Teaching & Examination Scheme:**

Teaching Scheme				**Evaluation Scheme**					
				Internal Evaluation			**ESE**		**Total**
L	**T**	**P**	**C**	**MSE**	**CE**	**P**	**Theory**	**P**	
-	**-**	**2**	**1**	**-**	**-**	**20**	**-**	**30**	**50**

L- Lectures; **T**- Tutorial; **P-** Practical; **C**- Credit; **MSE-** Mid-Semester Evaluation, **CE**- Continuous Evaluation, **ESE-** End Semester Examination

h. **Experiment List:**

Sr. NO.	**Experiment List**
1	**Python Basics and Data Structures** o Write programs for arithmetic operations, conditional statements, and loops o Work with strings, lists, tuples, sets, and dictionaries

	Implement list/dictionary comprehensions and basic problem-solving exercises
2	**Functions and OOP Concepts** o Define and use functions with arguments and return values o Implement recursion and lambda functions Create classes and implement OOP concepts: constructors, encapsulation, inheritance, polymorphism
3	**Exception Handling and File Operations** o Handle exceptions using try-except-else-finally o Read from and write to text/CSV files Work with JSON data for file-based applications
4	**Modules, Libraries, and Database Connectivity** o Work with Python modules and packages o Connect Python with MySQL and MongoDB (CRUD operations) Practice basic requests and API data handling
5	**Django Project Setup and Basic App** o Install Django in virtual environment and create a project o Set up Djongo (MongoDB integration) and define models Create migrations, admin console setup, and basic CRUD operations
6	**Django Views, Templates, and Static Files** o Create views, URL mapping, and template rendering (Jinja2) o Handle static and media files Create forms with validation for user input
7	**User Authentication and Authorization** o Implement Django's built-in authentication system o Create registration, login, and logout functionalities Handle user roles and access permissions
8	**RESTful API Development with Django** o Design and implement RESTful APIs with Django REST Framework o Handle GET, POST, PUT, DELETE requests Implement JWT-based authentication and error handling
9	**API Testing and Integration** o Test APIs using Postman or Python requests library o Integrate frontend forms with API endpoints Handle API versioning and response formatting
10	**Mini Project: Full-Stack Django Application and Deployment** o Build a mini project using Django + MongoDB o Implement CRUD operations, authentication, and REST API endpoints o Deploy the application to Heroku, Render, or Railway Apply basic security, performance optimization, and version control with Git