

B. Tech in Computer Science and Engineering

SEMESTER- IV

Sl No.	Course Code	Course Title	Hours Per Week			Total Credits	ESE	IA
			Lecture	Tutorial	Practical			
1.	105401	Computer Organization and Architecture	3	0	0	3	70	30
2.	105402	Formal Language and Automata Theory	3	1	0	4	70	30
3.	105403	Design and Analysis of Algorithms	3	0	0	3	70	30
4.	105404	Database Management System	3	0	0	3	70	30
5.	105405	Effective Technical Communication	3	0	0	3	70	30
6.	105406	Computer Networks	3	0	0	3	70	30
7.	105401P	Computer Organization and Architecture Lab	0	0	2	1	30	20
8.	105403P	Design and Analysis of Algorithms Lab	0	0	2	1	30	20
9.	105404P	Database Management System Lab	0	0	2	1	30	20
10.	105406P	Computer Networks Lab	0	0	2	1	30	20
11.	105407	NPTEL-I (Open Course)	12 Weeks			3	75	25
TOTAL						26	900	

Course Code- 105402 Formal Language & Automata Theory 3 1 0 4

Unit- 1.0: Introduction: **5 hrs**

Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Unit- 2.0: Regular languages and finite automata **9 hrs**

Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata

Unit- 3.0: Context-free languages and pushdown automata: **8 hrs**

Context-free grammars (CFG) and Context-free languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Unit- 4.0: Context-sensitive languages: **5 hrs**

Context-sensitive grammars (CSG) and Context-sensitive languages, linear bounded automata and equivalence with CSG.

Unit- 5.0: Turing machines: **8 hrs**

The basic model for Turing machines (TM), Turing recognizable (Recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

Unit-6.0: Undecidability: **7 hrs**

Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

Text/ Reference:-

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
5. John Martin, Introduction to Languages and the Theory of Computation, Tata McGraw Hill.

Course Code- 105404 Database Management System 3 0 0 3

Unit 1.0- Database system architecture 8 hrs

Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

Unit 2.0- Relational query languages: 10 hrs

Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Unit 3.0 – Storage strategies: 4 hrs

Indices, B-trees, hashing.

Unit 4.0- Transaction processing: 8 hrs

Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

Unit 5.0 – Database Security: 6 hrs

Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

Unit 6.0- Advanced topics: 6 hrs

Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

Text/ Reference:-

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
3. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
4. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison -Wesley.

Perform any ten Experiments**List of Experiments:**

1. Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted. The elements can be read from a file or can be generated using the random number generator.
2. Implement a Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted .The elements can be read from a file or can be generated using the random number generator.
3. Compute the transitive closure of a given directed graph using Warshall's algorithm.
4. Implement 0/1 Knapsack problem using Dynamic Programming.
5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
7. Print all the nodes reachable from a given starting node in a digraph using the BFS method.
8. Check whether a given graph is connected or not using the DFS method.
9. Find a subset of a given set $S = \{s_1, s_2, \dots, s_N\}$ of n positive integers whose sum is equal to a given positive integer d. For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1,2,6\}$ and $\{1,8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.
10. Implement any scheme to find the optimal solution for the Travelling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
11. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
12. Implement N Queen's problem using Backtracking.

Perform all Experiments**List of Experiments:**

1. Student should decide on a case study and formulate the problem statement.
2. Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.
3. Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys) Note: Student is required to submit a document showing the database tables created from ER Model.
4. Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize up to Third Normal Form.
5. Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables.
6. Practicing DML commands- Insert, Select, Update, Delete.
7. Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc.
8. Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).
9. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.
10. Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger.

Perform any ten Experiments**List of Experiments:**

1. Write a report that includes a diagram showing the topology, type of connection devices, and speed of the wired and wireless LAN in your organization. Also find out the MAC and IP addresses and the subnet mask of your computer.
2. Install and run a network diagnosis tool such as TCP dump or Wireshark. Start capturing packets on an active interface, open a browser and type the address of your favourite search engine. Wait till the page loads and stop capture. List out the type and number of each type of packets captured.
3. Write a program to create a server that listens to port 53 using stream sockets. Write a simple client program to connect to the server. Send a simple text message "Hello" from the client to the server and the server to the client and close the connection.
4. Write a program to create a chat server that listens to port 54 using stream sockets. Write a simple client program to connect to the server. Send multiple text messages from the client to the server and vice versa. When either party types "Bye", close the connection.
5. Write a program to create a server that listens to port 55 using stream sockets. Write a simple client program to connect to the server. The client should request for a text file and the server should return the file before terminating the connection.
6. Write a program to create a server that listens to port 56 using stream sockets. Write a simple client program to connect to the server. Run multiple clients that request the server for binary files. The server should service each client one after the other before terminating the connection.
7. Write a program to create a server that listens to port 57 using stream sockets. Write a simple client program to connect to the server. Run multiple clients that request the server for text files. The server should service all clients concurrently.
8. Write a program to create a server that listens to port 59 using datagram sockets. Write a simple client program that requests the server for a binary file. The server should service multiple clients concurrently and send the requested files in response.
9. Creation of sample network topologies using CISCO packet tracer.
10. Creation of a subnet, assignment of addresses and communication within and outside the subnet.
11. Error Detection using CRC (Python/Java): Implement the Cyclic Redundancy Check (CRC) algorithm for error detection in data transmission.
12. Hamming Code for Error Correction (Python/Java): Implement the Hamming Code algorithm to detect and correct single-bit errors in transmitted data.
13. Sliding Window Protocol (Python/Java): Simulate Stop-and-Wait and Go-Back-N ARQ protocols to demonstrate flow control in a network.
14. Selective Repeat ARQ (Python/Java): Implement the Selective Repeat ARQ protocol for reliable data transmission.
15. CSMA/CD and CSMA/CA Simulation (Python/Java): Simulate Carrier Sense Multiple Access with Collision Detection (CSMA/CD) and Collision Avoidance (CSMA/CA).
16. IPv4 and IPv6 Addressing (Python/Java): Implement a program to validate and classify IP addresses as IPv4 or IPv6.
17. ARP and RARP Implementation (Python/Java): Simulate Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP).
18. BOOTP and DHCP Simulation (Python/Java): Implement the BOOTP and DHCP mechanisms for dynamic IP address allocation.
19. Routing Algorithms (Python/Java): Implement the Distance Vector Routing (DVR) and Link State Routing (LSR) algorithms.
20. TCP and UDP Simulation (Python/Java): Implement a client-server model to demonstrate TCP

B. Tech in (CSE)

(SESSION- 2024 onwards)

- and UDP socket programming.
21. Congestion Control using Leaky Bucket Algorithm (Python/Java): Simulate the Leaky Bucket algorithm for congestion control.
 22. Congestion Control using Token Bucket Algorithm (Python/Java): Implement the Token Bucket algorithm to regulate network traffic.
 23. FTP and HTTP Simulation (Python/Java): Implement a program to simulate FTP file transfer and send HTTP GET/POST requests.
 24. DNS Query Resolution (Python/Java): Write a program to perform DNS resolution and retrieve IP addresses for given domain names.
 25. Email Client Simulation (Python/Java): Implement a basic email client that can send and receive messages using SMTP and POP3 protocols.



B. Tech in (CSE)

(SESSION- 2024 onwards)

Course Code-105407

NPTEL (Open Course)

12 Weeks (3 Credits)

NPTEL (Open Course) Guidelines:

NPTEL (Open Course) should be taken from the course other than core subjects of the branch.

