B.Tech (Information Technology)

SEMESTER- IV

Sl No.	Course Code	Course Title	Hours Per Week			Total Credits	ESE	IA
			Lecture	Tutorial	Practical			
1.	106401	Computer Organization and Architecture	3	0	0	3	70	30
2.	106402	Formal Language and Automata Theory	3	1	0	4	70	30
3.	106403	Design and Analysis of Algorithms	3	0	0	3	70	30
4.	106404	Database Management System	3	0	0	3	70	30
5.	106405	Effective Technical Communication	3	0	0	3	70	30
6.	106406	Computer Networks	3	0	0	3	70	30
7.	106401P	Computer Organization and Architecture Lab	0	0	2	1	30	20
8.	106403P	Design and Analysis of Algorithms Lab	0	0	2	1	30	20
9.	106404P	Database Management System Lab	0	0	2	1	30	20
10.	106406P	Computer Networks Lab	0	0	2	1	30	20
11.	106407	NPTEL-I (Open Course)	12 Weeks			3	75	25
TOTAL						26	900	

Semester-IV

Course Code- 106401 Computer Organization & Architecture 3 0 0 3

Unit-1.0 **7 hrs.**

Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU–registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction sets of some common CPUs.

Unit-2.0 **7 hrs.**

Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and-add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

Unit-3.0 **7 hrs.**

Introduction to x86 architecture. CPU control unit design: hardwired and micro- programmed design approaches, Case study – design of a simple hypothetical CPU. Memory system design: semiconductor memory technologies, memory organization.

Unit-4.0 **7 hrs.**

Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers–program controlled, interrupt driven and DMA, privileged and non- privileged instructions, software interrupts and exceptions. Programs and processes–role of interrupts in process state transitions, I/O device interfaces – SCII, USB.

Unit-5.0 **7 hrs.**

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.
Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

Unit-6.0 **7 hrs.**

Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. Block size, mapping functions, replacement algorithms, write policies.

Text/Reference:

1. "Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
2. "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.
3. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill
4. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
5. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

Course Code- 106402 Formal Language & Automata Theory 3 0 0 3

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 EducationAsia.
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- 106403 Design and Analysis of Algorithms 3 0 0 3

Unit-1.0 **7 hrs**

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

Unit-2.0 **7 hrs**

Introduction to Divide and Conquer paradigm: Binary Search, Quick and Merge sorting techniques, linear time selection algorithm, Strassen's Matrix Multiplication, Karatsuba Algorithm for fast multiplication etc. Introduction to Heap: Min and Max Heap, Build Heap, Heap Sort

Unit-3.0 **7 hrs**

Overview of Brute-Force, GreedyProgramming, Dynamic Programming, Branch-and-Bound and Backtrackingmethodologies. Greedy paradigm examples of exact optimization solution: Minimum Cost Spanning Tree, Knapsack problem, Job Sequencing Problem, Huffman Coding, Single source shortest path problem.

Unit-4.0 **7 hrs**

Dynamic Programming, difference between dynamic programming and divide and conquer, Applications: Fibonacci Series, Matrix Chain Multiplication, 0-1 Knapsack Problem, Longest Common Subsequence, Travelling Salesman Problem, Rod Cutting, Bin Packing.
Heuristics – characteristics and their application domains.

Unit-5.0 **7 hrs**

Graph and Tree Algorithms: Representational issues in graphs, Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms: Bellman- Ford algorithm,Dijkstra's algorithm & Analysis of Dijkstra's algorithm using heaps, Floyd-Warshall's all pairs shortest path algorithm.Transitive closure, Topological sorting, Network Flow Algorithm, Connected Component

Unit-6.0 **7 hrs**

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.Approximation algorithms, Randomized algorithms

Text/Reference:

1. Introduction to Algorithms, 4th Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MITPress/McGraw-Hill.
2. Horowitz & Sahani, "Fundamental of Computer Algorithm", Galgotia.
3. Basse, "Computer Algorithms: Introduction to Design & Analysis", Addision Wesley.
4. Algorithm Design, 1st Edition, Jon Kleinberg and Éva Tardos, Pearson.
5. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
6. Algorithms—A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Course Code- 106404 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.

Course Code- 106405 Effective Technical Communication 3 0 0 3

Unit 1.0- Information Design and Development 8 hrs

Different kinds of technical documents,Information development life cycle, Organization structures, factors affecting information and document design, Strategies for organization, Information design and writing for print and for online media.

Unit 2.0- Technical Writing, Grammar and Editing 10 hrs

Technical writing process, forms of discourse, Writing drafts and revising, Collaborative writing, creating indexes, technical writing style and language. Basics of grammar, study of advanced grammar, editing strategies to achieve appropriate technical style. Introduction to advanced technical communication, Usability, Human factors, Managing technical communication projects, time estimation, Single sourcing, Localization.

Unit 3.0 – Self-development and Assessment 4 hrs

Self assessment, Awareness, Perception and Attitudes, Values and belief, Personal goal setting, career planning, Self-esteem. Managing Time; Personal memory, Rapid reading, taking notes; Complex problem solving; Creativity.

Unit 4.0- Communication 8 hrs

Public speaking, Group discussion, Oral; presentation, Interviews, Graphic presentation, Presentation aids, Personality Development.

Unit 5.0 – Technical Writing 6 hrs

Writing reports, project proposals, brochures, newsletters, technical articles, manuals, official notes, business letters, memos, progress reports, minutes of meetings, event report.

Unit 6.0- Ethics 6 hrs

Business ethics, Etiquettes in social and office settings, Email etiquettes, Telephone Etiquettes, Engineering ethics, managing time, Role and responsibility of engineer, Work culture in jobs, Personal memo

Text/ Reference:-

1. David F. Beer and David McMurrey, Guide to writing as an Engineer, John Willey. New York, 2004.
2. Kulbhushan Kumar, Effective Communication Skills, Khanna Publishing House, 2019.
3. Diane Hacker, Pocket Style Manual, Bedford Publication, New York, 2003. (ISBN 0312406843).
4. Shiv Khera, You Can Win, Macmillan Books, New York, 2003.
5. Raman Sharma, Technical Communications, Oxford Publication, London, 2004.
6. Dale Jungk, Applied Writing for Technicians, McGraw Hill, New York, 2004. (ISBN: 07828357-4).
7. Sharma, R. and Mohan, K. Business Correspondence and Report Writing, TMH New Delhi 2002.
8. Xebec, Presentation Book, TMH New Delhi, 2000 (ISBN 0402213)

Course Code- 106406 Computer Networks **3 0 0 3**

Unit-1.0 **4 hrs.**

Overview of Data Communication and Networking: OSI Reference Model, TCP/IP Protocol Suite; Network Architecture and Physical Topology.

Unit-2.0 **10 hrs.**

Physical Layer: Analog and Digital Signals, Transmission Impairment, Data Rate Limits, Performance Analysis of a Network; Representation and Synchronization of Bits, Analog and Digital Transmission; Multiplexing and Spreading Techniques; Guided Transmission Media; Circuit, Packet and Virtual Circuit Switching.

Unit-3.0 **10 hrs.**

Data Link Layer: Framing, Flow and Error Control (Noiseless and Noisy Channels Protocols), PointToPoint Protocol; Random Access protocols (Pure/slotted ALOHA, CSMA/CD, CSMA/CA), Controlled Access Protocol (Bit-Map, Polling and Token Passing), Channelization (TDMA, FDMA, CDMA); Physical Addressing and Ethernet; Connecting LANs and Virtual LANs.

Unit-4.0 **7 hrs.**

Network Layer: Internet Protocol version 4 and 6; Address Mapping (ARP, RARP, BOOTP and DHCP), ICMP and IGMP, Routing Algorithms.

Unit-5.0 **7 hrs.**

Transport Layer: UDP, TCP; Congestion Control and QoS; Client-Server Model and Socket Interface.

Unit-6.0 **4 hrs.**

Application Layer: DNS, Remote Logging, Electronic Mail (SMTP, POP), FTP, Introduction to WWW and HTTP.

Text/ Reference:-

1. B. Forouzan, “Data Communication and Network”, McGraw-Hill Publications. 4th ed.
2. A. S. Tanenbaum., “Computer Networks”, Pearson Education Asia. 5th ed.
3. W. Stalling, “Data and Computer Communication”, PHI (EEE). 8th ed.
4. A. L. Garcia and I. Widjaja, “Communication Networks: Fundamental Concepts and Key Architectures”, Tata McGraw-Hill. 2nd ed.
5. S. Sharma, “A course in Computer Networks”, Kataria. 3rd ed.

B. Tech (Information Technology) **(SESSION- 2024 onwards)**

Course Code- 106403P Design and Analysis of Algorithms Lab 0 0 2 1

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 Dijikstra'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 = {s1, s2,....., sN} 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.

B. Tech (Information Technology) **(SESSION- 2024 onwards)**

Course Code- 106404P Database Management System Lab 0 0 2 1

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.

Course Code- 106406P Computer Network Lab 0 0 2 1

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

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 (Information Technology) **(SESSION- 2024 onwards)**

Course Code-106407 **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.