BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING / COMPUTER SCIENCE AND INFORMATION TECHNOLOGY / INFORMATION TECHNOLOGY / COMPUTER SCIENCE AND TECHNOLOGY

SECOND YEAR (FOURTH SEMESTER)

W.E.F. ADMISSION BATCH 2023-24

Sl. No.	Category	Course Code	Course	Contact Hrs. L-T-P	Credit	University Marks	Internal Evaluation
Subject (Theory)							
1	PC	CSPC2004	Database Engineering	3-0-0	3	100	50
2	PC	CSPC2005	Computer Organization and Architecture	3-0-0	3	100	50
3	PC	CSPC2006	Design and Analysis of Algorithms	3-0-0	3	100	50
4	PC	CSPC2007	Computer Networks	3-0-0	3	100	50
5	PC(ACC)	PCAC2009	Big Data Integration and Management	3-0-0	2	100	50
		PCAC2010	Application Development - Tools & Technologies				
		PCAC2011	Cloud Infrastructure & Applications				
		PCAC2012	Internet of Things and Cloud				
		PCAC2013	Robotics : Mobility & Design				
		PCAC2014	IT Fundamentals for Cybersecurity - II				
6	HS	HSHS2002	Organizational Behaviour	3-0-0	2	100	50
		HSHS2001	Engineering Economics				
Subject (Sessional / Practical)							
7	PC	CSPC2204	Database Engineering Lab.	0-0-3	1.5	-	100
8	PC	CSPC2205	Computer Organization and Architecture Lab.	0-0-3	1.5	-	100
9	PC	CSPC2206	Design and Analysis of Algorithms Lab.	0-0-3	1.5	-	100
10	PC	CSPC2207	Computer Networks Lab.	0-0-3	1.5	-	100
			Total	**18-0-12**	**22**	**600**	**700**

Note : Minimum four (04) weeks of Summer Course / Training / Internship / Skill Course / etc. after 4th Semester.

Click here to view/download the syllabus of the subjects.

CSPC2004 DATABASE ENGINEERING (3-0-0)

Course Learning Objectives

CLO 1. To provide a strong foundation in database concepts, technology, and practice.
CLO 2. To practice SQL programming through a variety of database problems.
CLO 3. To Understand the relational database design principles.
CLO 4. To demonstrate the use of concurrency and transactions in database
CLO 5. To design and build database applications for real world problems.
CLO 6. To become familiar with database storage structures and access techniques.

Module-1 (8 hours)

Introduction to Databases and overview of Database Languages and architecture:
Overview, Database System vs File System, Overall Database Structure, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications, Database System Concept, Types of users, Database design lifecycle, Data Model Schema and Instances, Types of data model, Three schema architecture, Data Independence and Database Language and Interfaces, Data Definitions Language, DML, the Database System environment,

Data Modeling Using the Entity Relationship Model:
ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Specialization, Generalization, Aggregation, Reduction of an ER Diagrams to Tables, Extended ER Model, Relationship of Higher Degree.

Module-2 (8 hours)

Introduction to Relational Data Model:
Relational Model Concepts, Codd's rules, Integrity Constraints, Entity Integrity, Referential Integrity, Key Constraints, Domain Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.

Relational Algebra and Calculus:
Selection and projection, set operations, renaming, Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relational comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs algebra, computational capabilities, Relational Database Design using ER-to- Relational mapping.

Structured Query Language:
Introduction to SQL- Characteristics, advantages of SQL, Types of SQL commands, SQL data definition and data types, specifying constraints in SQL, basic retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views and Indexes, Queries and Sub Queries. Aggregate Functions. Insert, Update and Delete Operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL.

Module-3 (9 hours)

Normalization:
Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd

Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form. Examples on normal forms.

Normalization Algorithms:

Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, Nulls, Dangling tuples, and alternate Relational Designs, Further discussion of Multivalued dependencies and 4NF, Other dependencies and Normal Forms.

Module-4 (8 hours)

File Structures, Hashing and Indexing:

Introduction, placing file records on disk, hashing techniques, Parallelizing Disk Access Using RAID Technology, indexing structures for files- Types of Single-Level Ordered Indexes, Dynamic Multilevel Indexes Using B-Trees and B+-Trees, Indexes on Multiple Keys

Query Processing, optimization and Database tuning:

Translating SQL Queries into Relational Algebra, Algorithms for External Sorting, Algorithms for SELECT and JOIN Operations, Algorithms for PROJECT and Set Operations, Implementing Aggregate Operations and OUTER JOINs, Combining Operations Using Pipelining, Using Heuristics in Query Optimization, Using Selectivity and Cost Estimates in Query Optimization, Physical Database Design in Relational Databases, an Overview of Database Tuning in Relational Systems

Module-5 (7 hours)

Transaction Processing:

Introduction to Transaction Processing, Transaction and System concepts, ACID properties, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Checkpoints, Deadlock Handling

Concurrency Control Techniques:

Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes, Recovery with Concurrent Transaction.

Course Outcomes

At the end of the course the student will be able to:

CO1 Identify, analyze and define database objects, enforce integrity constraints on a database using RDBMS.

CO2 Use Structured Query Language (SQL) for database manipulation and also demonstrate the basic of query evaluation.

CO3 Design and build simple database systems and relate the concept of transaction, concurrency control and recovery in database

CO4 Develop application to interact with databases, relational algebra expression. CO 5. Develop applications using tuple and domain relation expression from queries

Textbooks

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

3. C. J. Date, A. Kannan and S. Swamynathan, “An Introduction to Database Systems”, 8th ed, Pearson Education, 2006

Reference Books:

Abraham Silberschatz, Henry F. Korth and S. Sudarshan’s Database System Concepts 6th Edition Tata Mcgraw Hill Education Private Limited.

Weblinks and Video Lectures (e-Resources):

https://www.youtube.com/watch?v=ywTn9qHyI9I
https://www.youtube.com/watch?v=_qbKMdqQS6E
https://www.youtube.com/watch?v=PqPkYmRSQ_w

CSPC2005 COMPUTER ORGANIZATION AND ARCHITECTURE (3-0-0)

Objectives of the course:
To expose the students to the following:
1. How Computer Systems work & the basic principles
2. Instruction Level Architecture and Instruction Execution
3. The current state of art in memory system design
4. How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism
6. To impart the knowledge on micro programming
7. Concepts of advanced pipelining techniques.

Module-I: (8 Hrs.)
Functional blocks of a computer: CPU, memory, input-output subsystems, control Unit, Overview of Computer Architecture and Organization: Fundamentals of computer architecture, Organization of Von Neumann machine, Basic operation concepts, Performance and Historical perspective, Instruction set architecture of a CPU–registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.

Module-II: (08 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.

Module-III: (12 Hrs.)
CPU control unit design: hardwired and micro-programmed design approaches, Memory system design: semiconductor memory technologies, Memory Organization- Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.
Peripheral devices and their characteristics: I/O subsystems, I/O device interface, I/O transfers–program controlled, Asynchronous data transfer, Modes of Transfer, 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

Module-IV: (07 Hrs.)
Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.
Pipeline and Vector Processing: Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards. Vector Processing, Array Processor.
Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor arbitration, Inter-processor communication and synchronization, Cores, and Hyper-Threading ,Cache Coherence.

Module-V: (08 Hrs.)
Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Books:

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

Course outcomes

1. Draw the functional block diagram of a single bus architecture of a computer and describe the function of the instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.
2. Write assembly language program for specified microprocessor for computing 16-bit multiplication, division and I/O device interface (ADC, Control circuit, serial port communication).
3. Write a flowchart for Concurrent access to memory and cache coherency in Parallel Processors and describe the process.
4. Given a CPU organization and instruction, design a memory module and analyze its operation by interfacing with the CPU.
5. Given a CPU organization, assess its performance, and apply design techniques to enhance performance using pipelining, parallelism and RISC methodology

CSPC2006 DESIGN AND ANALYSIS OF ALGORITHMS (3-0-0)

Course Learning Objectives:

CLO1. Explain the methods of analyzing the algorithms and to analyze performance of algorithms.

CLO 2. State algorithm's efficiencies using asymptotic notations.

CLO3. Solve problems using algorithm design methods such as the brute force method, greedy method, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking and branch and bound.

CLO4. Choose the appropriate data structure and algorithm design method for a specified application.

CLO 5. Introduce P and NP classes.

Module-I: (8 Hours)

Notion of Algorithm: Growth of functions, Recurrences: The Master method, The Substitution method, The Iteration method, Asymptotic Notations and Basic Efficiency Classes (Use of Big O, θ, etc.) in analysis of algorithms, Mathematical Analysis of few Non-Recursive and Recursive Algorithms.

Module-II: (8 Hours)

Sorting and Searching Techniques: Selection Sort, Bubble Sort, Insertion Sort, Sequential Search, Binary Search, Depth First Search and Breadth First Search, Balanced Search Trees, AVL Trees, Red-Black Trees, Heaps and Heap Sort, Disjoint Set and their Implementation, Divide and Conquer Paradigm of problem solving, Complexity analysis and understanding of Merge Sort, Quick Sort, Binary Search Trees.

Module-III: (8 Hours)

Greedy Techniques: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's and Bellman Ford Algorithm, Huffman Trees, Knapsack problem.

Dynamic Programming Paradigm: Floyd-Warshall Algorithm, Optimal Binary Search trees, Matrix Chain Multiplication Problem, Longest Common Subsequence Problem, 0/1 Knapsack Problem, Maximum
Network Flow Problem.

Module-IV: (8 Hours)

String Matching Algorithms: Naive string-matching algorithm, The Rabin-Karp Algorithm, string matching with Finite Automata, Knuth Morris Pratt string matching algorithm.

Backtracking: n-Queen's problem, Hamiltonian Circuit problem, Subset-Sum problem, State Space Search Tree for these problems

Module-V: (8 Hours)

Branch and Bound: Travelling Salesman Problem and its State Space Search Tree.

Introduction to Computability: Polynomial-time verification, NP-Completeness and Reducibility, NP- Complete problems. Approximation Algorithms: Vertex Cover Problem.

Course outcome:

At the end of the course the student will be able to:

CO1 Analyze the performance of the algorithms, state the efficiency using asymptotic notations and analyze mathematically the complexity of the algorithm.

CO2 Apply divide and conquer approaches and decrease and conquer approaches in solving the problems analyze the same

CO3 Apply the appropriate algorithmic design technique like greedy method, transform and conquer approaches and compare the efficiency of algorithms to solve the given problem.

CO4 Apply and analyze dynamic programming approaches to solve some problems, and improve an algorithm time efficiency by sacrificing space.

CO5 Apply and analyze backtracking, branch and bound methods and to describe P, NP and Complete problems.

Textbooks:

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms", PHI Publication.
2. A.V. Aho, J. E. Hopcroft and J.D. Ullman, "The Design and Analysis of Computer Algorithms", Pearson Education.
3. R. S. Salaria, Khanna, "Data Structure & Algorithms", Khanna Book Publishing Co. (P) Ltd.

Reference Books:

1. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009. Pearson.

Web links and Video Lectures (e-Resources):

1. https://nptel.ac.in/courses/106/101/106101060/
2. http://cse01-iiith.vlabs.ac.in/

CSPC2007 COMPUTER NETWORKS (3-0-0)

Course Learning Objectives:

CLO1 Fundamentals of data communication networks.
CLO2 Software and hardware interfaces
CLO3 Application of various physical components and protocols
CLO4 Communication challenges and remedies in the networks.

Module-I (8Hours)

Introduction to Networks: Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Data and signals: analog and digital, periodic analog signals, digital signals, transmission impairments, data rate limit, Guided transmission media twisted pairs, coaxial cable, fiber optics, Wireless transmission, unguided transmission media.

Module–II (8 hours)

Data Link Layer: Design issues, framing, Error detection and correction, CRC codes Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

Module–III (8 Hours)

Connecting devices: Learning bridges, spanning tree bridges, repeaters, hubs, bridges, switches, routers and gateways, definition of multiplexing and types.

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, link state protocols, path vector routing, Congestion Control Algorithms, Quality of Service

Module–IV (10 Hours)

Internetworking: logical addressing, internet protocols, IP address, CIDR, IPv4 addressing, IPv6 Protocol addressing, addresses mapping, ICMP, IGMP, ARP, RARP, DHCP.

Transport Protocols: process to process delivery, UDP, TCP, TCP Service Model, TCP Sliding Window, TCP Congestion Control, congestion control and quality of service.

Module-V (6 Hours)

Application Layer- Introduction, providing services, Client server model, Standard client-server application-HTTP, FTP, electronic mail, TELNET, DNS.

Course Outcomes:

At the end of the course the student will be able to:

CO1 Learn the basic needs of communication system.
CO2 Interpret the communication challenges and its solution.
CO3 Identify and organize the communication system network components CO 4. Design communication networks for user requirements.

TEXT BOOKS:

1. A. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India.2
2. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.

REFERENCE BOOKS:

1. James F. Kurose, K. W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet”, Pearson Education.
2. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education

Weblinks and Video Lectures (e-Resources):

https://nptel.ac.in/courses/106105183
https://nptel.ac.in/courses/106105081

CSPC2204 DATABASE ENGINEERING LAB (0-0-3)

1. Consider the following schema for a Library Database:
   BOOK (Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS (Book_id, Author_Name) PUBLISHER (Name, Address, Phone)
   BOOK_COPIES (Book_id, Programme_id, No-of_Copies)
   BOOK_LENDING (Book_id, Programme_id, Card_No, Date_Out, Due_Date) LIBRARY_PROGRAMME (Programme_id, Programme_Name, Address)
   Write SQL queries to:
   - Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc.
   - Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2025 to Jun 2025.
   - Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
   - Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
   - Create a view of all books and its number of copies that are currently available in the library.

2. Consider the schema for College Database:
   STUDENT (REGNO, SName, Address, Phone, Gender) SEMSEC (SSID, Sem, Sec)
   CLASS (REGNO, SSID)
   COURSE (Subcode, Title, Sem, Credits)
   IAMARKS (REGNO, Subcode, SSID, Test1, Test2, Test3, FinalIA)
   Write SQL queries to:
   - List all the student details studying in fourth semester ‘C’ section.
   - Compute the total number of male and female students in each semester and in each section.
   - Create a view of Test1 marks of student REGNO ‘1BP22CS101’ in all Courses.
   - Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.

   Categorize students based on the following criterion:
   If FinalIA = 17 to 20 then CAT = ‘Outstanding’
   If FinalIA = 12 to 16 then CAT = ‘Average’ If FinalIA< 12 then CAT = ‘Weak’
   Give these details only for 8th semester A, B, and C section students.

3. Consider the following schema for Order Database:
   SALESMAN (Salesman_id, Name, City, Commission)
   CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id)
   ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)
   Write SQL queries to :
   - Count the customers with grades above Bangalore’s average.
   - Find the name and numbers of all salesman who had more than one customer.
   - List all the salesman and indicate those who have and don’t have customers in their cities (Use UNION operation.)
   - Create a view that finds the salesman who has the customer with the highest order of a day.

- Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

4. Consider the following tables namely “DEPARTMENTS” and “EMPLOYEES” Their schemas are as follows,
   Departments (dept _no, dept_ name, dept_location); Employees (emp_id, emp_name, emp_salary, dept_no);
   - Develop a query to grant all privileges of employees table into departments table
   - Develop a query to grant some privileges of employees table into departments table
   - Develop a query to revoke all privileges of employees table from departments table
   - Develop a query to revoke some privileges of employees table from departments table
   - Write a query to implement the save point.

5. Using the tables “DEPARTMENTS” and “EMPLOYEES” perform the following queries
   - Display the employee details, departments that the departments are same in both the emp and dept.
   - Display the employee’s name and Department name by implementing a left outer join.
   - Display the employee’s name and Department name by implementing a right outer join.
   - Display the details of those who draw the salary greater than the average salary.

6. Consider the schema for Election Commission:
   CONSTITUENCY (cons_id, csname, csstate, no_of_voters) PARTY (pid, pname, psymbol);
   CANDIDATES (cand_id, name, age, state, phone_no); VOTER (vid, vname, vage, vaddr)
   Queries:
   - List the details of the candidates who are contesting from more than one constituency which belongs to different states.
   - Display the state name having maximum number of constituencies.
   - Create a stored procedure to insert the tuple into the voter table by checking the voter’s age. If voter’s age is at least 18 years old, then insert the tuple into the voter else display the message “Not an eligible voter”.
   - Create a stored procedure to display the number_of_voters in the specified constituency, where the constituency name is passed as an argument to the stored procedure.
   - Create a TRIGGER to UPDATE the count of “Number_of_voters” of the respective constituency in “CONSTITUENCY” table, AFTER inserting a tuple into the “VOTER” table.

7. Consider the following relations containing airline flight information: FLIGHTS (flno, distance, departs, arrives)
   AIRCRAFT (aid, aname, cruisingrange) CERTIFIED (eid, aid)
   EMPLOYEES (eid, ename, salary)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in relational algebra, tuple relational calculus, and domain relational calculus.

- Find the eids of pilots certified for some Boeing aircraft.
- Find the names of pilots certified for some Boeing aircraft.
- Find the aids of all aircraft that can be used on non-stop flights from DELHI TO MUMBAI.
- Identify the flights that can be piloted by every pilot whose salary is more than $100,000. (Hint: The pilot must be certified for at least one plane with a sufficiently large cruising range.)
- Find the names of pilots who can operate some plane with a range greater than 3,000 miles but are not certified on any Boeing aircraft.
- Find the eids of employees who make the highest salary.
- Find the eids of employees who make the second highest salary.
- Find the eids of pilots who are certified for the largest number of aircraft.
- Find the eids of employees who are certified for exactly three aircraft.
- Find the total amount paid to employees as salaries.

8. Create cursor for Employee table & extract the values from the table. Declare the variables, Open the cursor & extract the values from the cursor. Close the cursor.
Employee (E_id, E_name, Age, Salary)

9. Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class.
Write a PL/SQL block for using procedure created with above requirement.
Stud_Marks (name, total_marks)
Result (Roll, Name, Class)
Frame the separate problem statement for writing PL/SQL Stored Procedure and function, inline with above statement. The problem statement should clearly state the requirements.

10. Write a PL/SQL block of code for the following requirements: Schema:
Borrower (Rollin, Name, Date_of_Issue, Name_of_Book, Status)
Fine (Roll_no, Date, Amt)
    - Accept roll_no & name of book from user.
    - Check the number of days (from date of issue), if days are between 15 to 30 then fine amount will be Rs 5per day.
    - If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5per day.
    - After submitting the book, status will change from I to R.
    - If condition of fine is true, then details will be stored into fine table.

    Frame the problem statement for writing PL/SQL block inline with above statement.
    NOTE: Use Control structure and Exception handling.

CSPC2205 COMPUTER ORGANIZATION AND ARCHITECTURE LAB. (0-0-3)

1. Simulation and design of Fast Multiplication and Division Programs.
2. Some experiments using hardware training kits for floppy drive, dot matrix printer etc.
3. Dismantling and Assembling a PC along with study of connections, ports, chipsets, SMPS etc. Draw a block diagram of motherboard and other board.
4. A Study Project on some hardware technologies (memory, serial bus, parallel bus, microprocessor, i/o devices, motherboard etc.)

CSPC2206 DESIGN AND ANALYSIS OF ALGORITHMS LAB. (0-0-3)

1. Sort a given set of n integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the brute force method works along with its time complexity analysis: worst case, average case and best case.

2. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

3. Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.

4. Write a program to solve Knapsack problem using Greedy method.

5. Write a program to find shortest paths to other vertices from a given vertex in a weighted connected graph, using Dijkstra's algorithm.

6. Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.

7. Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.

8. Write a program to solve All-Pairs Shortest Paths problem using Floyd's algorithm.
   Write a program to solve Travelling Sales Person problem using Dynamic programming.
   Write a program to solve 0/1 Knapsack problem using Dynamic Programming method.

9. Design and implement C++/Java Program to find a subset of a given set S = {Sl, 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}. Display a suitable message, if the given problem instance doesn't have a solution.

10. Design and implement a program to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

CSPC2207 COMPUTER NETWORKS LAB. (0-0-3)

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing

2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP

3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism

4. Take an example subnet of hosts and obtain a broadcast tree for the subnet.

5. Implement distance vector routing algorithm for obtaining routing tables at each node.

6. Implement data encryption and data decryption.

7. Write a program for congestion control using Leaky bucket algorithm

8. Write a program for frame sorting technique used in buffers.

9. Wireshark
   - Packet Capture Using Wire shark
   - Starting Wire shark
   - Viewing Captured Traffic
   - Analysis and Statistics & Filters.

10. Do the following using NS2 Simulator
    - NS2 Simulator-Introduction
    - Simulate to Find the Number of Packets Dropped
    - Simulate to Find the Number of Packets Dropped by TCP/UDP
    - Simulate to Find the Number of Packets Dropped due to Congestion
    - Simulate to Compare Data Rate& Throughput.
    - Simulate to Plot Congestion for Different Source/Destination
    - Simulate to Determine the Performance with respect to Transmission of Packets

HSHS2002 ORGANISATIONAL BEHAVIOUR (3-0-0)

Objectives:
The objective is to develop an understanding of the behavior of individuals and groups inside organizations and to enhance skills in understanding and appreciating individuals, interpersonal, and group process for increased effectiveness both within and outside of organizations. Further, it is to develop theoretical and practical insights and problem-solving capabilities for effectively managing the organizational processes.

Module-I: (06 Hrs.)
Fundamentals of OB: Definition, scope and importance of OB, Relationship between OB and the individual, Evolution of OB, Theoretical framework (cognitive), behavioristic and social cognitive), Limitations of OB.

Module-II: (12 Hrs.)
Attitude: Importance of attitude in an organization, Right Attitude, Components of attitude, Relationship between behavior and attitude, Developing Emotional intelligence at the workplace, Job attitude, Barriers to changing attitudes.
Personality and values: Definition and importance of Personality for performance, The Myers-Briggs Type Indicator and The Big Five personality model, Significant personality traits suitable to the workplace (personality and job — fit theory), Personality Tests and their practical applications.
Perception: Meaning and concept of perception, Factors influencing perception, Selective perception, Attribution theory, Perceptual process, Social perception (stereotyping and halo effect). Motivation: Definition & Concept of Motive & Motivation, The Content Theories of Motivation (Maslow's Need Hierarchy & Herzberg's Two Factor model Theory), The Process Theories (Vroom's expectancy Theory & Porter Lawler model), Contemporary Theories — Equity Theory of Work Motivation.

Module-III: (10 Hrs.)
Foundations of Group Behavior: The Meaning of Group & Group behavior & Group Dynamics, Types of Groups, The Five — Stage Model of Group Development.
Managing Teams: Why Work Teams, Work Teams in Organization, Developing Work Teams, Team Effectiveness & Team Building.
Leadership: Concept of Leadership, Styles of Leadership, Trait Approach Contingency Leadership Approach, Contemporary leadership, Meaning and significance of contemporary leadership, Concept of transformations leadership, Contemporary theories of leadership, Success stories of today's Global and Indian leaders.

Module-IV: (08 Hrs.)
Organizational Culture : Meaning & Definition of Organizational Culture, creating & Sustaining Organizational Culture, Types of Culture (Strong vs. Weak Culture, Soft Vs. Hard Culture & Formal vs. Informal Culture), Creating Positive Organizational Culture, Concept of Workplace Spirituality.

Module-V: (09 Hrs.)
Organizational Change: Meaning, Definition & Nature of Organizational Change, Types of Organizational Change, Forces that acts as stimulants to change.
Implementing Organizational Change : How to overcome theResistance to Change,
Approaches to managing Organizational Change, Kurt Lewin's-Three step model, Seven Stage model of Change & Kotter's Eight-Step plan for Implementing Change, Leading the Change Process, Facilitating Change, Dealing with Individual & Group Resistance, Intervention Strategies for Facilitating Organizational Change, Methods of Implementing Organizational Change, Developing a Learning Organization.

Course Outcomes:
At the end of the course, students will be able to:
1. Understand the basic concepts of OB, change management, organizational culture and their implementation in organizations.
2. Identify and examine team characteristics for improved organizational performance.
3. Apply theories and frameworks to solve problems and take effective decisions for organizational success.

4. Analyze group behavior and leadership styles for effective people management.
5. Evaluate individual personality types and group behaviours for improving organizational processes and practices.
6. Develop leadership competency to manage organizational situations.

Books:

1. Understanding Organizational Behaviour, Parek, Oxford
2. Organizational Behaviour, Robbins, Judge, Sanghi, Pearson.
3. Organizational Behaviour, K. Awathappa,HPH.
4. Organizational Behaviour, VSP Rao, Excel
5. Introduction to Organizational Behaviour, Moorhead, Griffin, Cengage.
6. Organizational Behaviour, Hitt, Miller, Colella, Wiley.

PCAC2012 INTERNET OF THINGS AND CLOUD (3-0-0)

OVERALL COURSE OBJECTIVES: To provide learners with an in-depth understanding of the evolution of Internet of Things and related technologies, equip them with the skills to utilize advanced technology platforms like DragonBoard™ and AWS, and allow them to apply these skills in developing innovative IoT-enabled applications and systems.

LEARNING OUTCOMES: On successful completion of the course the students shall be able to:

1. Understand, compare, and explain how telephony and media delivery networks operate.
2. Understand circuit switched networks and packet switched networks and their trade-offs.
3. Comprehend key innovations that have transformed the communication, entertainment, and consumer electronics industry.
4. Describe the DragonBoard™ 410c peripherals, I/O expansion capabilities, computing capabilities, and connectivity capabilities.
5. Use Linux terminal for embedded purposes and configure integrated development environment (IDE) for software development.
6. Understand and utilize various AWS cloud services such as EC2, IoT and more, to build and integrate projects that leverage the cloud.

COURSE CONTENT:

Module 1: Internet of Things: How did we get here? [21 Hours]

This course explores the convergence of multiple disciplines that have led to the advent of present-day smartphones and the Internet of Things. The lessons explore the evolution of telephony networks, broadcast networks, and consumer electronics, along with the impact of the internet, multimedia content, smartphones, and apps. It also covers the emerging, interconnected platform: the Internet of Things. Upon completion, learners will understand how peer-to-peer networks differ from broadcast networks, the tradeoffs between circuit-switched and packet-switched networks, and the workings of several key innovations and digital services. The course provides an important grounding for anyone interested in the technological development of the Internet of Things, and further resources for a more in-depth exploration of the topics.

Sub-Topics
Circuit Switched Networks
Computer Telephony
Features and Apps
Future Outlook
Packet Switched Networks
Wireless Technologies

Formative Assessments:
16 quizzes.

Module 2: Internet of Things V2: DragonBoard™ bring up and community ecosystem [21 Hours]

This course is designed for individuals seeking to develop the skills needed to prototype embedded products using advanced technologies. The course utilizes the DragonBoard™ 410c single board computer (SBC) to provide a hardware and software development environment for Internet of Things specialization courses. Ideal for learners interested in using Linux for embedded purposes, pursuing a career in the design and development of Internet of Things products, or those involved in entrepreneurial, innovative, or DIY communities, this course offers both theoretical knowledge and hands-on development practice. Key learning outcomes include understanding the DragonBoard™ 410c peripherals, navigating a Linux terminal, configuring an integrated development environment (IDE) for software development, utilizing Git and GitHub for version control, and creating projects that interface with sensors and actuators through GPIO and Arduino.

Sub-Topics
Advanced Projects and Code
Changing your Operating System (Supplemental / Optional)
DragonBoard Bringup and Ecosystem
Mezzanines and Sensors (Canned Demos w/ software)
Rescuing your Bricked Board (Supplemental / Optional)

Formative Assessments:

5 quizzes and 1 peer-review assignment.

Module 3: Internet of Things V2: Setting up and Using Cloud Services [10 Hours]

This course provides an introduction to Amazon Web Services (AWS) and its significance, enabling learners to make informed design decisions about which services to use. The course covers interfacing with the AWS cloud, developing software for data sending and receiving, and how to structure projects with diverse services. Upon completion, learners will have a clear understanding of the cloud, be able to install and configure the AWS CLI and SDK on a Linux system, utilize various AWS services such as EC2, IoT, etc., build projects heavily leveraging the cloud, and integrate the cloud into embedded systems.

Sub-Topics
Advanced Projects and Code - Deep dive
Systems Architecture

Cloud 101 for Dragonboard 410c
Real projects using AWS Cloud services

Formative Assessments:

3 quizzes and 1 peer-review assignment.

ASSESSMENT:

For summative assessments, Coursera will provide question banks for which exams can be conducted on the Coursera platform or the faculty will create their own assessments.

Note: If a Course or Specialization becomes unavailable prior to the end of the Term, Coursera may replace such Course or Specialization with a reasonable alternative Course or Specialization.