List of Experiments (to be conducted)

2nd Year Lab (Odd Sem)

1. DSA Lab (BCS 351)

Exp. No.	Name of Experiment	AKTU/ Value added
1	Write a menu-driven program to sort a user defined array in C using following methods: Bubble sort, Insertion sort, Selection sort	AKTU
2	WAP to implement Radix Sort and Quick sort in C.	AKTU
3	WAP to implement Linear search and Binary Search on Array	AKTU
4	WAP in C for Digit Extraction	AKTU
5	WAP to implement stack through Array and Linked List also perform various operations on stack	AKTU
6	WAP in C to balance Parenthesis.	AKTU
7	WAP to convert Infix to Postfix notation	AKTU
8	WAP to implement Queue through Array and Linked List also perform various operations on the Same.	AKTU
9	WAP to implement below mentioned Data Structure using Linked list: Circular linked List,Double linked list Reverse linked list,Polynomial Addition	AKTU
10	WAP to implement Tree and perform operations on that Tree	AKTU
11	WAP to Create a Min. heapand Max heap out of the tree.	AKTU
12	WAP to Create Graph using Adjacency Matrix, and perform BFS on that Graph	AKTU
13	WAP to implement Prim's and Kruskal Algorithm	AKTU
14	WAP to find shortest path Using Dijkstra Algorithm.	AKTU

2. <u>COA Lab (BCS 352)</u>

List of Experiments:

1. Implementing HALF ADDER, FULL ADDER using basic logic gates

2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.

3. Implementing 3-8 line DECODER.

4. Implementing 4x1 and 8x1 MULTIPLEXERS.

5. Verify the excitation tables of various FLIP-FLOPS.

6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.

7. Design of an 8-bit ARITHMETIC LOGIC UNIT.

8. Design the data path of a computer from its register transfer language description.

9. Design the control unit of a computer using either hard-wiring or micro-programming

based on its register transfer language description.

10. Write an algorithm and program to perform Matrix Multiplication of two n×n matrices on the 2-D mesh SIMD model

Course Beyond Syllabus (Value added programs)

11. To design a BCD adder

12. To design the circuit of full subtractor

3rd Year Lab (Odd Sem)

1. AI Lab (BCAI 501)

Exp. No.	Name of Experiment	AKTU/ Value added
1	Write a python program to implement Breadth First Search Traversal	AKTU
2	Write a python program to implement Water Jug Problem	AKTU
3	Write a python program to remove punctuations from the given string	AKTU
4	Write a python program to sort the sentence in alphabetical order	AKTU
5	Write a program to implement Hangman game using python.	AKTU
6	Write a program to implement Tic-Tac-Toe game using python.	AKTU
7	Write a python program to remove stop words for a given passage from a text file using NLTK	AKTU
8	Write a python program to implement stemming for a given sentence using NLTK	AKTU
9	Write a python program to POS (Parts of Speech) tagging for the give sentence using NLTK	AKTU
10	Write a python program to implement Lemmatization using NLTK	AKTU
11	Write a python program to for Text Classification for the give sentence using NLTK	AKTU
12	Write a program for Spam Detection using Machine Learning (Naïve Bayes)	Value Added
13	Write a program for Iris Flower Classification using K-Nearest Neighbors (KNN)	Value Added

2. DAA Lab (BCS 553)

Exp. No.	Name of Experiment	AKTU/ Value added
1	Program for Recursive Binary & Linear Search.	AKTU
2	Program for Heap Sort.	AKTU
3	Program for Merge Sort.	AKTU
4	Program for Selection Sort.	AKTU
5	Program for Insertion Sort.	AKTU
6	Program for Quick Sort.	AKTU
7	Knapsack Problem using Greedy Solution	AKTU
8	Perform Travelling Salesman Problem	AKTU
9	Find Minimum Spanning Tree using kruskal's Algorithm	AKTU
10	Implement N Queen Problem using Backtracking	AKTU
11	Implement, the 0/1 Knapsack problem using(a)Dynamic Programming method (b)Greedy method.	AKTU
12	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.	AKTU
13	Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.	AKTU
14	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.	AKTU
15	Write programs to (a) Implement All- Pairs Shortest Paths problem using Floyd's algorithm. (b)Implement Travelling Sales Person problem using Dynamic programming.	AKTU
16	Design and implement 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.	AKTU
17	Design and implement to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.	AKTU
18	Program to print all permutation of a given string	Value Added
19	Implement N Queen Problem using Backtracking for real world applications with example	Value Added

20	Implement Travelling Sales Person problem using Dynamic programming for real world problem with example	Value Added

3. DBMS Lab (BCS 551)

Exp. No.	Title/ objective of the Experiment to be conducted
1	a. Introduction to DBMS, RDBMS, DBMS Vs RDBMS b. Introduction to SQL & Types of SQL Statements (**DDL, DML, TCL, DCL**) c. Introduction to various RDBMS software d. Oracle Installation, Oracle Live
2	Creating Entity-Relationship Diagram using case tools.
3	**DDL Commands**: CREATE, ALTER, DROP, TRUNCATE, CREATING A TABLE From a Table Using SELECT Command
4	**DML Commands**: Managing data using INSERT, DELETE and UPDATE command
5	Data Constraints at Column Level and Table level: **PRIMARY Key, FOREIGN Key, UNIQUE, NOT NULL, CHECK, DEFAULT** value concept, Defining and Dropping Constraints using ALTER Table Command. Assigning User defined names to constraints Data Constraints at Column Level and Table level.
6	**DML Commands**: Retrieving database using **SELECT command, Order by clause**
7	**SQL Functions**: Date functions, Conversion functions
8	**SQL Functions**: **Aggregate functions**, Single row functions
9	Retrieving information from database using **Group by and Having clause**
10	Retrieving data using **Sub queries** Single row, Multiple rows, ANY operator, ALL operator, EXIST, Scalar, Multiple columns, Nested, Correlated
11	Retrieving data from multiple tables **using JOIN** EQUI, INNER, SELF, OUTER
12	Retrieving data using **SET operators** UNION, UNION ALL, INTERSECT, MINUS, retrieving data from multiple tables using JOIN Practice Questions
13	**PL/SQL** Function & Procedure
14	**PL/SQL** Cursors & Trigger