CA-C2T: PROBLEM SOLVING TECHNIQUES

Total Teaching Hours: 48	No. Of Lecture Hours/Week: 3
Max Marks:60	Credit : 3
Summative Assessment Marks: 60	Formative (internal) Assessment Marks: 40

Course Objective:

To study about algorithms, flowcharts and programs. To solve problems through logical thinking. To clearly understand the logic of the problem. To analyze the given problem and write the algorithm, flowchart. To write structured C programs using control statements. Demonstrate concepts like arrays, strings and structures.

Course Outcome:

Upon successful completion of the course the student will be able to:

- To clearly understand the logic of the problem.
- To analyze the given problem and write the algorithm and flowchart
- To write structured C programs, this is the foundation of any programming language.

UNIT 1: INTRODUCTION	Teaching Hours :12

Introduction: The Role of Algorithms in Computing, Algorithms as a technology, Analyzing algorithms, Designing algorithms, Growth of Functions, Asymptotic notation, Standard notations and common functions. Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of the Fibonacci sequence, Reversing the digits of an integer, Character to number conversion.

UNIT 2: Introduction of C Programming	Teaching Hours :12

C Programming: Getting Started, Variables and Arithmetic expressions. Input and Output: Standard input and output, formatted output- printf, variable length argument list, formatted input-scanf. Control Flow: Statements and Blocks, If-else, else-if, switch, loops: while loop, for loop, do while, break and continue, goto and labels. Pointers and Arrays: pointers and address, pointers and function arguments, multidimensional array, initialization of pointer arrays, command line arguments

UNIT 3: Factoring Methods	Teaching Hours 12

Factoring Methods: Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, computing the prime factors of an integer, generation of pseudo random numbers, raising a number to a large power. Array Techniques: Array order Reversal, Array counting or Histogramming, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, Finding the kth smallest element, multiplication of two matrices.

UNIT 4: Merging, Sorting, Searching	Teaching Hours :12
Merging: the two-way merge. Sorting: Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning. Searching: binary search, hash search. Text processing and Pattern searching: text line length adjustment, keyword searching in text, text line editing, linear pattern Search.	
Text Book:	
1. *R.G.Dromey, "How to Solve it by Computer", Pearson Education India, 2008.* 2. *Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, The MIT Press Cambridge, Massachusetts London, England, 2008* 3. *Brain M. Kernighan, and Dennis M. Ritchie, "The C Programming Language", 2nd edition, Princeton Hall Software Series, 2012.*	
Reference Text Books:	
1. *Steven S. Skiena, "The Algorithm Design Module", 2nd Edition, Springer-Verlag London Limited, 2008.* 2. *Donald E. Knuth, The Art of Computer Programming", Volume 1: Fundamental Algorithms, 3rd Edition, Addison Wesley Longman, 1997.* 3. *Donald E. Knuth, The Art of Computer Programming", Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.* 4. *Greg Perry and Dean Miller, "C programming Absolute Beginner's Guide", 3rd edition, Pearson Education, Inc, 2014.*	
Web Resources:	
1. *http://algorithmsforinterviews.com "Algorithms for Interviews"* 2. *NPTEL :: Computer Science and Engineering - NOC:Introduction to programming in C*	

CA-C3T: DATA STRUCTURES

Total Teaching Hours: 48	No. Of Lecture Hours/Week: 3
Max Marks:60	Credit : 3
Summative Assessment Marks: 60	Formative (internal) Assessment Marks: 40

Course Objective:

Analyze performance of algorithms and choose the appropriate data structure and algorithm design method for a specified application. Illustrate linear representation of data structures: Stack, Queues, Lists. Demonstrate sorting and searching algorithms. To familiarize with basic techniques of algorithm analysis, master implementation of linked lists, binary trees and graph algorithms. Find suitable data structure during application development/Problem Solving

Course Outcome:

Upon successful completion of the course the student will be able to:

- Use different types of data structures, operations and algorithms
- Use of stack, Queue, Lists, Trees in problem solving.
- Use of Trees, Graph in problem solving.
- Apply searching and sorting operations.

UNIT 1: INTRODUCTION	Teaching Hours :12

Introduction and Overview: Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms. Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, Multi-dimensional arrays, Matrices and Sparse matrices.

UNIT 2: Linear Data Structures	Teaching Hours :12

Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list.

Stacks: Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Conversion of infix expression to postfix expression, Evaluation of Post fix expression, Application of Stacks, Recursion, Towers of Hanoi, Implementation of recursive procedures by stack.

Queues: Definition, Array representation of queue, Linked list representation of queues. Types of queue: Simple queue, Circular queue, Double-ended queue, Priority queue, Operations on Queues, Applications of queues.

UNIT 3: Non Linear Data Structures	Teaching Hours 12
Binary Trees: Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, Lexicographic Search Trees: Tries, External Searching: B-Trees, Applications of Trees. Graphs: Mathematical Back ground, Computer Representation, Graph Traversal, Topological Sorting	
UNIT 4: Searching and Sorting	Teaching Hours :12
Searching: Introduction and Notation, Sequential Search, Binary Search, Comparison of Methods. Sorting: Introduction and Notation, Insertion Sort, Selection Sort, Shell Sort, Divide And Conquer, Merge sort for Linked List, Quick sort for Contiguous List. Hashing: Sparse Tables, Choosing a Hash function, Collision Resolution with Open Addressing, Collision Resolution by Chaining.	
Text Book:	
1. *Seymour Lipschutz, "Data Structures with C", Schaum's outLines, Tata Mc Graw Hill, 2011.* 2. *Robert Kruse, C.L.Tondo, Bruce Leung,Shashi Mogalla, "Data Structures and Program Design using C", Pearson Education, 2009.*	
Reference Text Books:	
1. *Mark Allen Weiss, " Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2013.* 2. *Forouzan, "A Structured Programming Approach using C",2nd Edition, Cengage LearningIndia,2008.*	
Web Resources:	
1. *NPTEL :: Computer Science and Engineering - Data Structures And Algorithms* 2. *NPTEL :: Computer Science and Engineering - NOC:Programming, Data Structures and Algorithms*	

CA-C4P: PROBLEM SOLVING LAB USING C

Total Teaching Hours:	No. Of Lecture Hours/Week : 3
Exam Marks:50	Credit : 2
Summative Assessment Marks: 25	Formative (internal) Assessment Marks: 25

Course Objective:

To impart adequate knowledge on the need of programming languages and problem solving techniques. To develop an in-depth understanding of functional and logical concepts of C Programming. To provide exposure to problem-solving through C programming. To familiarize the basic syntax and semantics of C Language.

Course Outcome:

Upon successful completion of the course the student will be able to:

- Construct mathematical models using C programs.
- Understand the fundamentals of arrays in C programming.
- Understand the concept and implementations of functions and strings
- Implement different Operations on pointers, structures and unions.

NOTE: For all the programs write the output, flowchart and number of basic operations performed.

1. To read radius of a circle and to find area and circumference
2. To read three numbers and find the biggest of three
3. To check whether the number is prime or not
4. To read a number, find the sum of the digits, reverse the number and check it for palindrome
5. To read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
6. To read percentage of marks and to display appropriate message (Demonstration of else-if ladder
7. To find the roots of quadratic equation
8. To read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
9. To remove Duplicate Element in a single dimensional Array
10. To perform addition and subtraction of Matrices
11. To find factorial of a number
12. To generate fibonacci series
13. To remove Duplicate Element in a single dimensional Array
14. To find the length of a string without using built in function
15. To demonstrate string functions
16. To read, display and add two m x n matrices using functions
17. To read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
18. To Swap Two Numbers using Pointers
19. To demonstrate student structure to read & display records of n students
20. To demonstrate the difference between structure & union.

CA-C5P: DATA STRUCTURES LAB PROGRAMS

Total Teaching Hours:	No. Of Lecture Hours/Week: 3
Max Marks:50	Credit : 2
Summative Assessment Marks: 25	Formative (internal) Assessment Marks: 25

Course Objective:
To implement recursive functions. To arrange data using different sorting techniques. To implement stack, queue, linked list. To implement tree structures

Course Outcome:
Upon successful completion of the course the student will be able to:

- Implement different sorting and searching algorithms
- Implement the stack, Queue and their applications
- Implement various types of linked lists and their applications
- Perform basic operations on trees

NOTE: For all the programs write the output, flowchart and number of basic operations performed.

1. Given {4,7,3,2,1,7,9,0} find the location of 7 using Linear and Binary search and also display its first occurrence.
2. Given {5,3,1,6,0,2,4} order the numbers in ascending order using Bubble Sort Algorithm
3. Perform the Insertion and Selection Sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements {61,16,8,27} into singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to insert the elements {61,16,8,27} into linear queue and delete three elements from the list. Display your list after each insertion and deletion.
6. Write a program to insert the elements {61,16,8,27} into circular queue and delete 4 elements from the list. Display your list after each insertion and deletion.
7. Write a program to insert the elements {61,16,8,27} into ordered singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
8. Write a program to add $6x^3+10x^2+0x+5$ and $4x^2+2x+1$ using linked list.
9. Write a program to push 5,9,34,17,32 into stack and pop 3 times from the stack, also display the popped numbers.
10. Write a recursive program to find GCD of 4,6,8.
11. Write a program to inert the elements {5,7,0,6,3,9} into circular queue and delete 6,9&5 from it(using linked list implementation)..
12. Write a program to convert an infix expression x^y/(5*z)+2 to its postfix expression
13. Write a program to evaluate a postfix expression 5 3+8 2 - *.
14. Write a program to create a binary tree with the elements

{18,15,40,50,30,17,41} after creation insert 45 and 19 into tree and delete 15,17 and 41 from tree. Display the tree on each insertion and deletion operation

15. Write a program to create binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
16. Write a program to Sort the following elements using heap sort {9.16,32,8,4,1,5,8,0}
17. Given S1={"Flowers"} ; S2={"are beautiful"} I. Find the length of S1 II. Concatenate S1 and S2 III. Extract the substring "low" from S1 IV. Find "are" in S2 and replace it with "is"

CA-C6T: COMPUTER ARCHITECTURE

Total Teaching Hours: 48	No. Of Lecture Hours/Week: 3
Max Marks:60	Credit : 3
Summative Assessment Marks: 60	Formative (internal) Assessment Marks: 40

Course Objective:

To conceptualize the basics of organizational and architectural design of a digital computer. Be familiar with the history and development of modern computers. Be familiar with Number system and Boolean Algebra. Be familiar with Combinational and logic circuits. Be familiar with organization and design of modern computers and its architecture.

Course Outcome:

Upon successful completion of the course the student will be able to:

- To use math and Boolean algebra in performing computations in various number systems
- Understand the basic components of a computer, including CPU, memories, and input/output, and their organization.
- Ability to design efficient combinational and sequential logic circuit implementations from functional description of digital systems.

UNIT 1: INTRODUCTION	Teaching Hours :12

Number Systems: Binary, Octal, Hexa decimal numbers, base conversion, addition, subtraction of binary numbers, one's and two's complements, positive and negative numbers, character codes ASCII, EBCDIC.

Computer Arithmetic: Addition and Subtraction, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations.

Structure of Computers: Computer types, Functional units, Basic operational concepts, Von-Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer, Digital Logic Circuits: Logic gates, Boolean algebra, Map Simplification.

Combinational Circuits: Half Adder, Full Adder, Flip Flops.

Sequential circuits: Shift registers, Counters, Integrated Circuits, Mux, Demux, Encoder, Decoder. Data representation: Fixed and Floating point.

UNIT 2: Basic Computer Organization & Design	Teaching Hours :12

Basic Computer Organization and Design: Instruction codes, Computer Registers, Computer Instructions and Instruction cycle. Timing and Control, Memory-Reference Instructions, Input-Output and interrupt.

Central processing unit: Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC), CISC vs RISC

UNIT 3: Register Transfer & Micro-operations	Teaching Hours 12
Register Transfer and Micro-operations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logic shift unit. Micro-programmed Control: Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit. Input Output: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. Instruction level parallelism: Instruction level parallelism (ILP)-over coming data hazards, limitations of ILP	
UNIT 4: Memory System	Teaching Hours :12
Memory System: Memory Hierarchy, Semiconductor Memories, RAM(Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, Performance considerations, Virtual memory, Paging, Secondary Storage, RAID. Multiprocessors And Thread level Parallelism: Characteristics of multiprocessors, Multi-Threaded Architecture, Distributed Memory MIMD Architectures, Interconnection structures,	
Text Book:	
1. *Mano M Morris, "Computer System Architecture", 3rd edition Pearson India(2019).* 2. *William Stallings, "Computer Organization and Architecture designing for performance", 10th edition, Pearson(2016)*	
Reference Text Books:	
1. *Subrata Ghoshal, "Computer Architecture And Organization", Pearson India(2011).* 2. *Andrew S. Tanenbaum " Structured Computer Organization", 5th edition, Pearson Education Inc(2006).* 3. *Carl Hamacher, Zvonks Vranesic,SafeaZaky, "Computer Architecture And Organization", 5th edition McGraw Hill New Delhi,India(2002).* 4. *Kai Hwang, "Advanced Computer Architecture - Parallelism, Scalability, Programmability", Tata Mcgraw-Hill (2008).*	
Web Resources:	
1. *NPTEL :: Computer Science and Engineering - NOC:Computer architecture and organization* 2. *NPTEL :: Computer Science and Engineering - NOC:Computer Architecture(Course sponsored by Aricent)*	

CA-C8T: DATABASE MANAGEMENT SYSTEMS

Total Teaching Hours: 48	No. Of Lecture Hours/Week: 3
Max Marks:60	Credit : 3
Summative Assessment Marks: 60	Formative (internal) Assessment Marks: 40

Course Objective:
Provide a strong foundation in database concepts, technology, and practice. Practice SQL programming through a variety of database problems. Demonstrate the use of concurrency and transactions in database. Design and build database applications for real world problems

Course Outcome:
Upon successful completion of the course the student will be able to:

- Identify, analyze and define database objects, enforce integrity constraints on a database using RDBMS.
- Use Structured Query Language (SQL) for database manipulation.
- Design and build simple database systems.
- Develop application to interact with databases.

UNIT 1: Introduction	Teaching Hours :12

Databases and Database Users: Introduction, An example, Characteristics of the Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of Using DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS. Database System Concepts and Architecture: Data Models, Schemas, and Instances, Three-schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client-Server Architectures, Classification of Database Management Systems.

UNIT 2: Data Model & ER Diagram	Teaching Hours :12

Data Modeling Using Entity-Relationship Model: Using High-Level Conceptual Data Models for Database Design, An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design Company Database Diagrams, Naming Conventions and Design. Issues, File organization and storage, secondary storage devices, type of single level ordered index, multi-level indexes, indexes on multiple keys, other types of indexes.

UNIT 3: Relational Model & Relational Algebra	Teaching Hours 12

Relational Model and Relational Algebra: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraint Violations, Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from SET Theory, Binary Relational Operations: JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra. Relational Database Design: Anomalies in a database, functional dependency, normal forms, lossless join and dependency, BCNF, normalization through synthesis, higher order normal forms.

SQL- SQL Data Definition and Data Types, Specifying Constraints in SQL, Schema Change Statements in SQL, Basic Queries in SQL, More Complex SQL Queries, Insert, Delete and Update Statements in SQL, Specifying Constraints as Assertion and Trigger, Views(Virtual Tables) in SQL, Embedded SQL, Dynamic SQL.	
UNIT 4: SQL	Teaching Hours :12
Introduction to transaction processing, transaction and system concepts, desirable properties of transactions, transaction support in SQL. Concurrency control techniques: two-phase locking techniques, concurrency control based on timestamp ordering. Recovery techniques: recovery concepts, recovery in multi-database systems, database backup and recovery from catastrophic failures.	
Text Book:	
1. *Elmasri and Navathe: Fundamentals of Database Systems, 7th Edition, Addison - Wesley, 2016.* 2. *Silberschatz, Korth and Sudharshan Data base System Concepts, 7th Edition, Tata McGraw Hill, 2019.*	
Reference Text Books:	
1. *C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson education, 2009* 2. *Database Management Systems :Raghu Ramakrishnan and Johannes Gehrke: , 3rd Edition, McGraw-Hill, 2003*	
Web Resources:	
1. *NPTEL :: Computer Science and Engineering - NOC:Data Base Management System* 2. *Data Base Management System - Course (nptel.ac.in)* 3. *NPTEL :: Computer Science and Engineering - Database Design*	

CA-C9P: JAVA PROGRAMMING LAB

Total Teaching Hours:	No. Of Lecture Hours/Week: 3
Max Marks:50	Credit : 2
Summative Assessment Marks: 25	Formative (internal) Assessment Marks: 25

Course Objective:

Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc. Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc. and exception handling mechanisms. Understand the principles of inheritance, packages, interfaces and multithreading. Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and Event Handling.

Course Outcome:

Upon successful completion of the course the student will be able to:

- Write Java application programs using OOP principles and proper program structuring.
- Write Java programs to implement error handling techniques using exception handling.
- Apply event handling on AWT and Swing components

NOTE: For all the programs write the output, number of basic operations performed.

1. Write a simple java application, to print the message, "Welcome to java"
2. Write a program to display the month of a year. Months of the year should be held in an array.
3. Write a program to demonstrate a division by zero exception
4. Write a program to create a user defined exception say Pay Out of Bounds. .
5. Write a java program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.
6. Write a program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.
7. Write a program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.
8. Write a java program to create a student class with following attributes: Enrollment_id: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declaredas zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.

9. In a college first year class are having the following attributesName of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class
10. Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class
11. Write a Java program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.
12. Create a package‘ student.Fulltime.BCA‘ in your current working directory a. Create a default class student in the above package with the following attributes: Name, age, sex. b. Have methods for storing as well as displaying
13. Write a small program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
14. Write a program to handle Null Pointer Exception and use the “finally” method to display a message to the user.
15. Write a program which create and displays a message on the window
16. Write a program to draw several shapes in the created window
17. Write a program to create an applet and draw grid lines
18. Write a program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
19. Create a frame which displays your personal details with respect to a button click
20. Create a simple applet which reveals the personal information of yours.
21. Write a program to move different shapes according to the arrow key pressed.
22. Write a java Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night
23. Demonstrate the various mouse handling events using suitable example.
24. Write a program to create menu bar and pull-down menus.

CA-C10P: DATABASE MANAGEMENT SYSTEM LAB

Total Teaching Hours:	No. Of Lecture Hours/Week: 3
Max Marks:50	Credit : 2
Summative Assessment Marks: 25	Formative (internal) Assessment Marks: 25

Course Objective:
To implement recursive functions. To arrange data using different sorting techniques. To implement stack, queue, linked list. To implement tree structures

Course Outcome:
Upon successful completion of the course the student will be able to:

- Create tables with various constraints with DDL, DML.
- Work on various aggregate functions.
- Work on various tables.
- Perform TCL commands.

NOTE: For all the programs write the output.

1. Draw E-R diagram and convert entities and relationships to relation table for a given scenario. Two assignments shall be carried out i.e. consider two different scenarios (eg. bank, college)
   Consider the Company database with following Schema
   EMPLOYEE (FNAME, MINIT, LNAME, SSN, DATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)
   DEPARTMENT (DNAME, DNUMBER,MGRSSN,MSRSTARTDATE)
   DEPT_LOCATIONS (DNUMBER, DLOCATION)
   PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)
   WORKS_ON (ESSN, PNO<HOURS)
   DEPENDENT (ESSN, DEPENDENT_NAME, SEX, BDATE, RELATIONSHIP)

2. Perform the following:
   a. Viewing all databases, Creating a Database, Viewing all Tables in a Database,
   Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)

3. Perform the following:
   a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database.

4. For a given set of relation schemes, create tables and perform the following Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause).

5. Execute the fallowing queries
   a. How the resulting salaries if every employee working on the 'Research' Departments is given a 10% raise.
   b. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department

6. Execute the fallowing queries
   a. Retrieve the name of each employee Controlled by Department number 5 (use EXISTS operator).
   b. Retrieve the name of each dept and number of employees working in each Department which has at least 2 employees

7. Execute the fallowing queries
   a. For each project, retrieve the project number, the project name, and the number of employee who work on that project.(use GROUP BY)
   b. Retrieve the name of employees who born in the year 1990's

8. For each Department that has more than five employees, retrieve the department number and number of employees who are making salary more than 40000.

9. For each project on which more than two employees work, retrieve the project number, project name and the number of employees who work on that project.

10. For a given set of relation tables perform the following: Creating Views (with and without check option), Dropping views, Selecting from a view

PART B

Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries.
BRANCH (Branchid, Branchname, HOD)
STUDENT (USN, Name, Address, Branchid, sem)
BOOK (Bookid, Bookname, Authorid, Publisher, Branchid)
AUTHOR (Authorid, Authorname, Country, age)
BORROW (USN, Bookid, Borrowed_Date)

1. Perform the following:
   a. Viewing all databases, Creating a Database, Viewing all Tables in a Database,
   Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)
   Execute the following Queries:

2. a. List the details of Students who are all studying in 2nd sem BCA.
   b. List the students who are not borrowed any books.

3. a. Display the USN, Student name, Branch_name, Book_name, Author_name, Books_Borrowed_ Date of 2nd sem BCA Students who borrowed books.
   b. Display the number of books written by each Author.

4. a. Display the student details who borrowed more than two books.
   b.Display the student details who borrowed books of more than one Author.

5. a. Display the Book names in descending order of their names.
   b. List the details of students who borrowed the books which are all published by the same publisher.

   Consider the following schema:
   STUDENT (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA)

6. Perform the following:
   a. Creating Tables (With and Without Constraints),
   Inserting/Updating/Deleting
   Records in a Table, Saving (Commit) and Undoing (rollback)

7. Execute the following queries:
   a. Find the GPA score of all the students.
   b. Find the students who born on a particular year of birth from the date_of_birth column.

8. a. List the students who are studying in a particular branch of study.
   b. Find the maximum GPA score of the student branch-wise.