From
Dr. A. Ravi Prasad,
Lecturer & Chairman-BOS (UG), KCDC,
Govt, Degree College for Men,
Srikalahasthi.

To
The Principal,
Krishna Chaitanya Degree College (A),
Nellore -524003.

Sir,

Sub. : KCDC, Nellore-BOS Computer Science Submission of revised syllabus for various UG Major Programs -Reg.

_ _ _ * * * _ _ _

Under the subject Cited the B.O.S of Computer Science (U.G) had met on 12-11-2025 at 03:15 PM in Online to revise the syllabus for various UG Major Programs for the Academic Year 2025-2026. I am here with submitting the syllabus for the following UG programs after the discussion of the BOS by online, confirming the Rules and Regulations laid down by the APSCHE and VSU.

i. B.Sc. (Computer Science)
ii. B.Sc. (Artificial Intelligence)
iii. BCA (Computer Applications)
iv. BCA (Artificial Intelligence)
v. BCA (Artificial Intelligence & Data Science)
vi. B.Com. (Computer Applications)
vii. Skill Enhancement Course (Common to all Programs)

The Revised Syllabus along with Model Papers and signatures of B.O.S members, are Enclosed.

Thanking you,

Dr. A. Ravi Prasad
(**Chairman – BoS**)

Members:

1 **Prof.K.Vijaya Lakshmi**,
Professor, Dept. of Computer Science,
Sri Venkateswara University, Tirupati.

2 **Dr.M.Sreedevi**,
Associate Professor, Dept. of Computer Science,
Sri Venkateswara University, Tirupati.

3 **Prof.P.Penchalaiah**,
Professor & Dean(Academics), Dept. of CSE,
Narayana Engineering College, Nellore.

4 **Sri.C.V.Pavan Kumar,**
Dept. of Computer Science,
Krishna Chaitanya Degree College, Nellore

5 **Sri.Y.Penchala Narasimhulu,**
Dept. of Computer Science,
Krishna Chaitanya Degree College, Nellore

6 **Sri.T.Venkatapathi,**
Dept. of Computer Science,
Krishna Chaitanya Degree College, Nellore.

7 **Sri.Ch.Dinesh Kumar**,
Founder,
Billion Bright Software Solutions, Nellore

8 **Smt.M.Sarika Jain**,
Assistant Professor,
Dept. of Cyber Forensics,
Madras University, Chennai.

Krishna Chaitanya Degree College (Autonomous)
(Affiliated to V.S.University, Nellore)
Nellore

Syllabus for 4-Year UG Honors in B.Sc. (Computer Science) as Major in consonance with Curriculum framework w.e.f. AY 2025-26

COURSE STRUCTURE (for Semesters I & II)

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits	Max. Marks Internal Assessment	Max. Marks University Exam	Total Marks
I	I	1	Computer Fundamentals and Office Automation	3	3	30	70	100
			Computer Fundamentals and Office Automation-Practical	2	1	0	50	50
		2	Problem Solving Using C	3	3	30	70	100
			Problem Solving Using C-Practical	2	1	0	50	50
	II	3	Data Structures using C	3	3	30	70	100
			Data Structures using C-Practical	2	1	0	50	50
		4	Database Management System	3	3	30	70	100
			Database Management System-Practical	2	1	0	50	50

SEMESTER-I

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. **Understand foundational computing concepts**, including the evolution of computers, block diagrams, and generational progress.
2. **Develop knowledge of computer architecture**, focusing on system components.
3. **Acquire practical skills in document creation**, formatting, and digital presentations using word processing tools.
4. **Gain proficiency in spreadsheet operations**, such as data entry, formulas, functions, and charting techniques.
5. **Introduce data visualization and basic modeling principles**, fostering analytical thinking in structuring and interpreting data sets.

Course Outcomes

1. At the End of the Course, The Students will be able to explain the historical evolution of computers, and identify key components in a block diagram.
2. Learners will demonstrate **basic blocks of a computer and Hardware Components.**
3. Learners will create professional-level documents and **design visually appealing presentations** using word processing software.
4. Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations.**
5. Learners will apply data modeling techniques to **analyze, organize, and represent data effectively** in various scenarios.

UNIT-I: Number Systems, Introduction, Functional Components

Number Systems: Binary, Decimal, Octal, Hexadecimal; conversions between number systems (Decimal to Binary, Binary to Decimal)

Introduction to Computers: Characteristics and Limitations of Computer, Types of Computers, Block Diagram of Computer, Computer Generations

Functional Components: Input devices and output devices.

UNIT –II: Memory, Software, N/w fundamentals

Memories: Primary memory, Secondary Memory, and cache memory.

Software : Types of Software, Application Software, System Software.

Networking Fundamentals: Definition, need for networks, types (LAN, WAN, MAN), topology (Star, Ring, Bus).

UNIT –III: MS-Office & MS-Word

Introduction to MS Office & MS Word: Features of MS-Word, MS-Word Window components, creating, saving and opening documents, Printing documents

Formatting documents: Selecting text, Formatting characters, changing cases, Paragraph formatting, Bullets & numbering

Editing Text: Copying & moving data, Finding & replacing text, Reversing actions (undo), Header & footer

Working with Graphics: Inserting pictures from Computer - Insert Shapes

UNIT IV: MS-Excel Fundamentals

Introduction to MS Excel & Its features: Excel Features, MS-Excel window components, Spreadsheets, workbooks, creating, saving & editing a workbook, Renaming sheet, cell entries (numbers, labels, and formulas), find and replace, Adding and deleting rows and columns, Formatting worksheet

Formatting options: Adjusting row height and column width - Formatting cell values, conditional formatting

UNIT-V: Excel Functions, Sorting, Filtering, Charts

Formulas & Functions: Definition, operators, Cell referencing (Relative, Absolute, Mixed) in a formula, Inserting a function in Excel, Types of functions in Excel: Mathematical, Statistical, Logical, Text

Working with Data ranges: Sorting: Sorting on single column, sorting on multiple columns - Filtering**:** Filtering data using AutoFilter

Working with Charts: Different types of charts, Creating a chart, Parts of chart, Changing chart type, changing chart options

Textbooks:

1. **Fundamentals of Computers,** Reema Thareja, Oxford University Press, Second Edition
2. **Fundamentals of Computers**, V. Rajaraman – PHI Learning
3. **Introduction to Computers** by Peter Norton – McGraw Hill
4. **Microsoft Office 2007 Fundamentals**, 1st Edition By Laura Story, Dawna Walls

References:

1. **Microsoft Office 365 In Practice** by Randy Nordell – McGraw Hill Education
2. **Excel 2021 Bible** by Michael Alexander, Richard Kusleika – Wiley
3. **Microsoft Official Docs and Training**: https://learn.microsoft.com
4. **Networking All-in-One For Dummies** by Doug Lowe – Wiley
5. **Google Workspace Learning Center**: https://support.google.com/a/users/

Activities:

Outcome: At the End of the Course, The Students will be able to identify key components in a block diagram.

Activity: Create a digital poster or info graphic illustrating the timeline of computer generations with key innovations.

Evaluation Method: Rubric-based assessment of the poster presentation on a 10-point scale focusing on:

- Correct identification of block diagram components
- Visual organization and creativity

Outcome: Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.

Activity: Prepare a formal report (e.g., project proposal) in a word processor and present it using a slide deck with transitions, embedded media, and design elements.

Evaluation Method: Performance-based evaluation using a 10-point scoring scale:

- Formatting and structure of the document
- Presentation aesthetics and clarity
- Communication skills during presentation

Outcome: Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations.**

Activity: Analyze a dataset (e.g., student scores or sales data) using spreadsheet software. Apply formulas and create relevant charts.

Evaluation Method: Practical test with a rubric:

- Correct use of formulas
- Accuracy of data summaries

Outcome: Learners will apply data modeling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Activity: Prepare an interactive dashboard for a given data set using EXCEL.

Evaluation Method: Evaluation of the dashboard on a 10-point scoring scale:

- Presentation aesthetics and clarity
- Instructiveness
- Communication skills during presentation

SEMESTER-I

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Demonstration of Assembling and Dissembling of Computer Systems.
2. Identify and prepare notes on the type of Network topology of your institution.
3. Prepare your resume in Word.
4. Using Word, write a letter to your higher official seeking 10-days leave.
5. Design a visiting card for Managing Director of a company as per the following specification.
    a. Sizeofvisitingcardis3½×2
    b. Name of the company with big font
    c. Phone number, Fax number and E-mail address with appropriate symbols.
    d. Office and Residence address separated by a line
6. Using a spreadsheet, prepare your class Time Table.
7. Using Spreadsheet, calculate the Gross and Net salary of employees (Min 5) considering all the allowances

SNO	Employee Number	Employee Name	Basic Pay	DA	HRA	GPF	Gross Pay	Income Tax	Net Pay
1									
2									

DA:-56% of the basic pay if Basic pay is greater than 20000 or else 44%.
HRA:-15% of the Basic pay subject to maximum of Rs.4000.
GPF: -10% of the basic pay.
INCOME TAX:-10% of basic if Basic pay is greater than 20000. Find who is getting highest salary & who is get lowest salary?

8. Using a Spreadsheet, calculate the Gross and Net salary of employees (Min 5) considering all the allowances.
9. Create an electronic spread sheet in which you implement conversion of numbers.
    a. Convert Decimal Numbers into Binary:35,68,95,78,165,225,355,375,465
    b. Convert Binary Numbers into Decimal:101,1101,11101,11111,10001,11101111
10. Generate the class-wise and subject-wise results for a class of 20 students. Also generate the highest and lowest marks in each subject.
11. Using IF, AND, OR, and IFERROR to Automate Grade Evaluation.
    a. Create a table of student scores in different subjects.
    b. Use IF to assign grades (A/B/C/Fail).
    c. Use IFERROR to handle missing scores or invalid data

12. The ABC Company shows the sales of different product For5years.Create BAR Graph, 3D and Pie chart for the following.

S.No.	Year	Pro1	Pro2	Pro3	Pro4
1	1989	1000	800	900	1000
2	1990	800	80	500	900
3	1991	1200	190	400	800
4	1992	400	200	300	1000
5	1993	1800	400	400	1200

13. Enter the following data into the sheet.

Name	**Department**	**Salary**
Anusha	Accounts	12000
Rani	Engineering	24000
Lakshmi	Accounts	9000
Purnima	Marketing	20000
Bindu	Accounts	4500
Tejaswi	Accounts	11000
Swetha	Engineering	15000
Saroja	Marketing	45000
Sunitha	Accounts	5600
Sandhya	Engineering	24000
Harika	Marketing	8000

- Extract records for department t in Accounts and Salary > 10000
- Sort the data by salary with the department using “sort commands”.
- Sort department wise salaries in descending order
- Calculate total salary for each department

14. Designing a Data Entry Form with Drop-downs and Input Rules
    a. Create a student registration form.
    b. Add drop-down lists for course selection using Data Validation.
    c. Add input messages to guide users.
    d. Add error alerts for wrong entries.

15. Monthly Budget Planning using Goal Seek and Scenario Manager
    a. Create a simple personal budget (income, expenses, savings).
    b. Use Goal Seek to determine income needed to save a desired amount.
    c. Use Scenario Manager to compare different budgeting scenarios (best/ worst/ realistic case).
    d. Create a one-variable Data Table to analyze how different expenses affect savings.

SEMESTER – I

COURSE 1: COMPUTER FUNDAMENTALS & OFFICE AUTOMATION

Model Question Paper

Time: 3 hours Max. Marks: 70

SECTION - A

Answer any FIVE of the following questions.

(Marks: 5x4 marks =20 marks)

1. Write about characteristics of computers.
2. Define Number System. Explain binary number system
3. What is cache memory?
4. Briefly explain LAN and WAN.
5. Explain about the features of MS-Word.
6. Write about inserting pictures from computer into the document.
7. How to create a new workbook in MS-Excel?
8. Write about of conditional formatting in MS-Excel.
9. Define formula. Explain about the formulas in MS-Excel.
10. Explain about operators in MS-Excel.

SECTION - B

Answer FIVE questions. Choosing ONE question from each unit.

(Marks: 5x10 marks =50 marks)

UNIT-I

11. Explain briefly about output Devices.

(Or)

12. Explain about block diagram of a computer

UNIT-II

13. What is software? Explain various types of software used in computers.

(Or)

14. Explain about the different network topologies.

UNIT-III

15. How to create, save and open a document in MS-word?

(Or)

16. Explain editing operations in MS-Word.

UNIT-IV

17. Write about MS-Excel window components..

(Or)

18. Explain different formatting options in Excel.

UNIT-V

19. What is filter? How to use filters in MS-Excel.

(Or)

20. Define Chart. Explain the types of charts in MS-Excel.

Instruction to Paper Setter:

Two questions must be given from each unit in Section-A.

SEMESTER-I

COURSE 2: PROBLEM SOLVING USING C

Theory **Credits: 3** **3 hrs/week**

Course Objectives:

1. Understand the fundamentals of computer programming, Apply structured problem-solving approaches using algorithms, flowcharts, and C programming constructs.
2. Develop efficient logic using decision-making, loop, and jump control statements.
3. Utilize derived data types like arrays and strings for modular program design.
4. Design and implement modular solutions using functions, recursive logic, pointer operations, and dynamic memory management.
5. Handle complex data structures including structures, unions, and text file operations.

Course Outcomes:

At the end of the course, students will be able to:

1. Understand basic computing concepts, programming paradigms and write structured C programs.
2. Apply control flow statements to solve logical and repetitive tasks in C.
3. Implement arrays and string operations to manage and manipulate data efficiently.
4. Design modular code using functions, recursion, and appropriate parameter passing.
5. Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

UNIT-I: Introduction to computer programming

Introduction, Programming Languages - Generations of Programming Languages, Algorithms and Flow Charts,

C Fundamentals: Features of C, C Tokens: Keywords - Identifiers - Constants - Operators in C. Basic Data Types in C, Variables, Structure of C Program - I/O Statements (scanf, printf)

UNIT-II: Control statements:

Decision making Statements: simple if, if..else, else if ladder, nested if, switch statements – Programming Examples

Loop Control Statements: while loop, for loop and do-while loop

Jump Control statements: break and continue Statement - goto Statement

UNIT – III: Derived data types in C

Arrays: Introduction - Declaration of Arrays - Accessing elements of the Array – Storing Values in Array - one dimensional array for inter-function communication – Two dimensional Arrays - two dimensional arrays for inter-function communication

Strings: Introduction - String operations - String functions

UNIT – IV: Functions, Structures and Unions

Functions: Introduction - using functions - Function declaration/ prototype – Function definition - function call - return statement - Passing parameters - Scope of variables -Storage Classes - Recursive functions.

Structure and Unions: Introduction - Nested Structures - Arrays of Structures – Structures and Functions - Unions - Arrays of Unions Variables

UNIT- V: Dynamic Memory Management

Pointers: Introduction to Pointers - declaring Pointer Variables - Passing Arguments to Functions using Pointer - Pointer and Arrays - Dynamic Memory Allocation

File Handling: Introduction to Files, File modes, File operations, Reading Data from Files, Writing Data from Files, Detecting the End-of-file

Text Books:

1. Programming in ANSI C, E. Balagurusamy, Tata McGraw Hill, 6 th Edn,
2. Computer fundamentals and programming in C, Reema Theraja, Oxford University Press

Reference Books:

1. Let us C, Y Kanetkar, BPB publications
2. Head First C: A Brain-Friendly Guide, David Griffiths, Dawn Griffiths

Activities:

Outcome: Understand basic computing concepts, programming paradigms and write structured C programs.

Activity: Create a concept map of computing fundamentals and programming paradigms (procedural, structured, object-oriented). Then, they write a structured C program (e.g., a calculator or student grade system) using proper syntax, indentation, and modular design.

Evaluation Method: Rubric-based Code Review & Viva to check the

- The correctness of the concept map
- Correct use of structure (main + functions)
- Identification of paradigm used
- Code readability and documentation

Outcome: Apply control flow statements to solve logical and repetitive tasks in C.

Activity: Implement a program that solves a logic puzzle (e.g., number guessing game, pattern generation, or prime number finder) using if, switch, for, while, and do-while.

Evaluation Method: Automated Test Cases + Peer Review to check the

- Correct use of control statements
- Logical correctness of output
- Efficiency and edge case handling
- Peer feedback on clarity and logic

Outcome: Implement arrays and string operations to manage and manipulate data efficiently.

Activity: Build a program that stores and arranges student marks in ascending and descending order using arrays and performs string operations like concatenation, comparing, and formatting names.

Evaluation Method: Functional Demonstration + Code Walkthrough to check the

- Correct array and string usage
- Memory efficiency
- Handling of invalid inputs
- Explanation of sorting/searching logic

Activity:

- **Recursive Problem Solver**
  Students write a modular program to solve a recursive problem (e.g., factorial, Fibonacci, or Tower of Hanoi) using functions with parameters and return values.

Evaluation Method:

- **Code Trace + Written Quiz**
  - Correct function decomposition
  - Proper parameter passing (by value/reference)
  - Recursion depth and base case handling

- Quiz on tracing recursive calls

Outcome: Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

Activity: Create a program that dynamically stores user input (e.g., survey responses) using pointers and writes/reads the data to/from a text file.

Evaluation Method: Memory Debugging + File I/O Assessment to check the

- Proper use of malloc, calloc, realloc, and free
- Pointer arithmetic and dereferencing
- File creation, reading, writing, and error handling

SEMESTER-I

COURSE 2: PROBLEM SOLVING USING C

Practical **Credits: 1** **2 hrs/week**

List of Experiments:

1. Write a program to check whether the given number is Armstrong or not.
2. Write a program to find the sum of individual digits of a positive integer.
3. Write a program to generate the first n terms of the Fibonacci sequence.
4. Write a program to find both the largest and smallest number in a list of integer values
5. Write a program to demonstrate change in parameter values while swapping two integer variables using Call by Value & Call by Address
6. Write a program to perform various string operations.
7. Write a program to search an element in a given list of values.
8. Write a program that uses functions to add two matrices.
9. Write a program to calculate factorial of given integer value using recursive functions
10. Write a program for multiplication of two N X N matrices.
11. Write a program to sort a given list of integers in ascending order.
12. Write a program to calculate the salaries of all employees using Employee (ID, Name, Designation, Basic Pay, DA, HRA, Gross Salary, Deduction, Net Salary) structure.
    a. DA is 30 % of Basic Pay
    b. HRA is 15% of Basic Pay
    c. Deduction is 10% of (Basic Pay + DA)
    d. Gross Salary = Basic Pay + DA+ HRA
    e. Net Salary = Gross Salary - Deduction
13. Write a program to read / write the data from / to a file.
14. Write a program to reverse the contents of a file and store in another file.
15. Write a program to create Book (ISBN,Title, Author, Price, Pages, Publisher)structure and store book details in a file and perform the following operations
    a. Add book details
    b. Search a book details for a given ISBN and display book details, if available
    c. Update a book details using ISBN
    d. Delete book details for a given ISBN and display list of remaining Books

SEMESTER – I
COURSE 2: PROBLEM SOLVING USING C
Model Question Paper

Time: 3 hours Max. Marks: 70

SECTION - A

Answer any FIVE of the following questions.
(Marks: 5x4 marks =20 marks)

1. Write the Generations of programming language.
2. Write about variables used in C.
3. Explain switch statement with example.
4. Explain Break and Continue statements.
5. Define an array. How do you store values in an array?
6. What is string? Write about any three string operations.
7. Write about any two storage classes
8. Write about nested structures.
9. How do you declare pointer variables?
10. Write file modes in C

SECTION - B

Answer FIVE questions. Choosing ONE question from each unit.
(Marks: 5x10 Marks = 50 marks)

UNIT-I

11. What is Data type? Write about various data types used in C language.

(Or)

12. Explain Flow Chart and Algorithm with Example.

UNIT-II

13. Explain decision Branching statements

(Or)

14. Explain Looping Control statements with an example program

UNIT-III

15. Define Array. Explain one dimensional array with an example

(Or)

16. Define String. Explain various string functions used in C language.

UNIT-IV

17. What is Function? Write about recursive function with an example.

(Or)

18. What is structure? Briefly explain structures used in C language.

UNIT-V

19. Write about Dynamic Memory Allocation.

(Or)

20. What is file? Explain about reading and writing data to files

Instruction to Paper Setter:

Two questions must be given from each unit in Section-A.

SEMESTER-II

COURSE 3: DATA STRUCTURES USING C

Theory **Credits: 3** **3 hrs/week**

Course Objectives:

1. Understand fundamental concepts of algorithms and data structures with focus on complexity analysis and abstract data types.
2. Explore various types of linked lists and their dynamic memory representations and operations.
3. Analyze and implement linear data structures, such as stacks and queues, and examine their real-world applications.
4. Apply sorting and searching algorithms, understanding their performance implications and optimization strategies.
5. Design and manipulate hierarchical and graph-based structures, applying traversal algorithms and understanding their practical uses in computing.

Course Outcomes:

Learners will be able to:

1. Explain algorithm characteristics, time and space complexity, and asymptotic notations with clarity.
2. Implement and analyze different types of linked lists, including insertion, deletion, and traversal operations.
3. Develop stack and queue data structures using arrays and linked lists, and apply them in expression evaluation.
4. Apply efficient searching and sorting algorithms to solve computational problems and evaluate performance trade-offs.
5. Construct and traverse tree and graph structures, using them to solve problems like shortest path and spanning trees.

UNIT-I: Basic Concepts:

Algorithm: Definition and characteristics, Complexity analysis: Space Complexity, Time Complexity, Asymptotic Notations.

Introduction to Data structures: Definition, Types of Data structures, Abstract Data Types (ADT), Introduction to Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array.

UNIT-II: Linked Lists:

Linked Lists: Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list; Implementation of Single Linked List ADT: Creating a List, Traversing a linked list, Searching in linked list, Insertion and deletion into linked list (At first Node, Specified Position, Last node).

UNIT-III: Stacks and Queues:

Stacks: Introduction to stack ADT, Implementation of stacks using array and Linked List, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation.

Queues: Introduction to Queue ADT, Implementation of Queues using array and Linked List, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue.

UNIT-IV: Searching and Sorting:

Searching: Linear or Sequential Search, Binary Search,

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

UNIT-V: Trees and Graphs:

Trees: Tree Terminology, Binary Tree Representation, Traversal techniques, Binary Search Tree-Definition, Operations on a Binary Search Tree: Creation, Search, Insertion & deletion.

Graphs: Introduction to Graphs, Terminology, Representation (Adjacency Matrix, Adjacency List), Traversal of Graphs (DFS, BFS), Applications of Graphs, Concept of Shortest Path Problems, Concept of Minimum Cost Spanning Tree

Textbooks:

1. Data Structures Using C, Balagurusamy E. Tata MCGraw Hill
2. Data Structures using C, Reema Thareja, Third Edition, Oxford University Press

Reference Books:

1. Data Structures, Lipschutz, Schaum's Outline Series, Tata Mcgraw-hill
2. Data Structures Using C, Ch. Vijay Kumar, Pen Press International

Activities:

Outcome: Explain algorithm characteristics, time and space complexity, and asymptotic notations with clarity

Activity: Create a comparative chart of algorithms with different notations related to time and space complexities.

Evaluation Method: Rubric-based assessment of the chart for correctness, clarity, and depth of explanation on a 10-point scale.

Outcome: Implement and analyze different types of linked lists, including insertion, deletion, and traversal operations

Activity: Code a menu-driven program in C to implement single linked lists with all basic operations.

Evaluation Method: Practical lab assessment with test cases and Viva-style questioning to explain pointer manipulation.

Outcome: Develop stack and queue data structures using arrays and linked lists, and apply them in expression evaluation

Activity: Build a program to convert infix expressions to postfix and evaluate them using stacks; Implement queues using both arrays and linked lists with enqueue/dequeue operations.

Evaluation Method: Code review and execution of programs for sample cases and evaluation based on correctness and efficiency.

Outcome: Apply efficient searching and sorting algorithms to solve computational problems and evaluate performance trade-offs

Activity: Implement and compare sorting algorithms (e.g., selection sort and bubble sort) and searching algorithms (e.g., Linear vs. Binary Search) on datasets of varying sizes. Record number of swaps and iterations for preparing a chart to assimilate the results.

Evaluation Method: Performance report with graphs and analysis. Oral presentation or peer review discussing trade-offs and algorithm selection rationale.

Outcome: Construct and traverse tree and graph structures, using them to solve problems like shortest path and spanning trees

Activity: Implement binary trees and graphs using adjacency lists/matrices.

Evaluation Method: Lab demo with sample inputs and visual output (e.g., tree traversal order, graph paths).

SEMESTER-II

COURSE 3: DATA STRUCTURES USING C

Practical **Credits: 1** **2 hrs/week**

List of Experiments

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array
   a. Add an element at the beginning of an array
   b. Insert an element at given index of array
   c. Update an element using a values and index
   d. Delete an existing element
2. Write a program to implement Single Linked List with insertion, deletion and traversal operations
3. Write a program to implement Doubly Linked List with insertion, deletion and traversal operations
4. Write a program to implement the Stack operations using Arrays and Linked Lists.
5. Write a program to convert a given infix expression to a postfix expression using stacks.
6. Write a program to implement the Queue operations using Arrays and Linked Lists.
7. Write a program to implement the Circular Queue operations using Arrays.
8. Write a program for Binary Search Tree Traversals
9. Write a program to search an item in a given list using the following Searching Algorithms
   a. Linear Search
   b. Binary Search.
10. Write a program for implementation of the following Sorting Algorithms
    a. Bubble Sort
    b. Insertion Sort
    c. Quick Sort
    d. Merge Sort

SEMESTER – II
COURSE 3: DATA STRUCTURES USING C
Model Question Paper

Time: 3 hours Max. Marks: 70

SECTION - A

Answer any FIVE of the following questions.
(Marks: 5x4 marks =20 marks)

1. Define space complexity.
2. Write the types of data structures.
3. What is Linked List?
4. Write about circular single linked list.
5. What is stack? Write the applications of stack.
6. Write about priority queue.
7. What is searching? Discuss about linear searching.
8. Write about selection sort.
9. Write about tree traversal techniques.
10. What is Graph?

SECTION-B

Answer FIVE questions. Choosing ONE question from each unit.
(Marks: 5x10 marks =50 marks)

UNIT-I

11. Define algorithms. Write the characteristics of algorithm.

(Or)

12. What is abstract data type? Write the Comparison between Linked List and Array.

UNIT-II

13. Define Doubly Linked List. Explain in detail.

(Or)

14. Write the implementation of single linked list.

UNIT-III

15. Write about Implementation of stacks using array.

(Or)

16. What is queue? Write about the different types of queues.

UNIT-IV

17. Write about binary search with example.

(Or)

18. Sort the following data using bubble sort 79 15 47 32 22 13 09 56.

UNIT-V

19. What is binary search tree? Write the operations of binary search tree.

(Or)

20. Explain in detail about DFS with examples.

Instruction to Paper Setter:
Two questions must be given from each unit in Section-A.

SEMESTER-II

COURSE 4: DATABASE MANAGEMENT SYSTEM

Theory **Credits: 3** **3 hrs/week**

Course Objectives:

1. To understand the fundamentals of data, information, and the evolution from file-based systems to modern database management systems.
2. To develop the ability to design conceptual data models using Entity-Relationship (ER)
3. To explore relational model principles, such as keys, integrity constraints, relational algebra and calculus, and normalization.
4. To perform data definition and manipulation using SQL commands including queries, joins, sub queries, views, and set operations.
5. To apply procedural logic using PL/SQL, incorporating control structures, functions, procedures, and database triggers.

Course Outcomes:

At the end of the course, students will be able to:

1. **Describe** the fundamentals of data, database systems, and the differences between file-based and database approaches. **Compare and classify** various DBMS architectures, data models, and their components.
2. **Design** conceptual data models using Entity-Relationship diagrams.
3. **Apply** relational model concepts, including CODD rules and normalization techniques.
4. **Construct and execute** SQL queries for data definition, manipulation, aggregation, joining, and sub queries, including views and set operations.
5. **Develop** PL/SQL programs incorporating control structures, procedures, and functions to manage database behavior effectively.

UNIT I

Introduction to Database Management System: Introduction to data, information, database, database management systems, Classification of Databases, advantages and disadvantages of database approach, Components of Database Management System

The Relational Database Model: Various Data Models, Relational Database model, Keys used in Relational model, Codd's relational database rules.

UNIT II

Entity–Relationship Model: Introduction, The components of an Entity–Relationship model, entities, attributes relationships, Classification of Entity Sets, Attribute Classification, Relationship Degree, and Relationship Classification.

Normalization: Purpose of Normalization, concept of functional dependency, normal forms based on functional dependency (lNF, 2NF and 3NF), Boyce-Codd normal form (BCNF)

UNIT III

Structured Query Language: Introduction, SQL literals, Data types in SQL, SQL operators, Commands in SQL, Data Definition Language (DDL) commands, Creating tables, Table Modification, Table Truncation, Creating Tables with constraints, Dropping tables, Data Manipulation Language (DML) commands: Inserting data into tables, updating data in tables and deleting data in tables. Transaction Control commands in SQL, Queries.

UNIT IV

Aggregate Functions, Sub queries and correlated queries, Relational Set Operators, Joining Database Tables, Views: Types of views, creating views, dropping views, Data Control Language (DCL) commands: Grant and Revoke commands.

UNIT V

PL/SQL: Introduction, Structure of PL/SQL program, PL/SQL Data Types, operators used in PL/SQL, variables, declaring variables in PL/SQL, Creating and running a PL/SQL Program, Control Structures, Conditional control statements, Iterative Control statements

Textbooks:

1. Database System Concepts, Avi Silberschatz, Henry F. Korth,S. Sudarshan, Seventh Edition, McGraw-Hill
2. Database Management Systems by Raghu Ramakrishnan, McGrawhill

Reference Books:

1. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
2. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson

Activities:

Outcome: Describe the fundamentals of data, database systems, and the differences between file-based and database approaches. Compare and classify various DBMS architectures, data models, and their components, including the three-schema architecture.

Activity: Create a comparative presentation or info graphic illustrating:

- File-based vs. DBMS approaches
- Types of DBMS architectures (1-tier, 2-tier, 3-tier)
- Data models and the three-schema architecture

Evaluation Method: Rubric-based assessment of the presentation covering clarity, accuracy, and depth of comparison. Include a short quiz to test conceptual understanding.

Outcome: Design conceptual data models using Entity-Relationship diagrams

Activity: Model a university or hospital database using ER diagrams that shows:

- Entity sets, relationships
- Participation and cardinality constraints

Evaluation Method: Diagram submission with peer review and instructor feedback. Use a checklist to assess completeness, correctness, and notation usage.

Outcome: Apply relational model concepts, including CODD rules and normalization techniques.

Activity: Normalize a given unstructured dataset up to 3NF. Then, write sample queries.

Evaluation Method: Written assignment graded on:

- Correctness of normalization steps
- Accuracy of relational algebra expressions
- Short-answer questions on CODD rules

Outcome: Construct and execute SQL queries for data definition, manipulation, aggregation, joining, and sub queries, including views and set operations.

Activity: Implement a mini-project (e.g., Library or Inventory DB) using SQL. Include:

- Table creation (DDL)
- Data manipulation (DML)
- Aggregation, joins, subqueries, views, and set operations

Evaluation Method: Lab-based practical test with query execution and output validation. Include a viva to explain logic and optimization.

Outcome: Develop PL/SQL programs incorporating control structures, procedures and functions to manage database behavior effectively.

Activity: Build a PL/SQL-based payroll or student grading system using:

- Procedures and functions
- Control structures (IF, LOOP)
- Triggers for automated updates

Evaluation Method: Code review and demonstration. Evaluate based on:

- Syntax correctness
- Logical flow

SEMESTER-II

COURSE 4: DATABASE MANAGEMENT SYSTEM

Practical **Credits: 1** **2 hrs/week**

Experiment 1 : Database: Inventory Management

Table 1: Products

Structure:

Column Name	Data Type	Constraints
product_id	NUMBER(10)	PRIMARY KEY
product_name	VARCHAR2(50)	NOT NULL
price	NUMBER(10,2)	CHECK(price > 0)
stock_qty	NUMBER(4)	CHECK(stock_qty >= 0)

Sample Data:

product_id	product_name	price	stock_qty
1	Pen	10.00	100
2	Notebook	50.00	200
3	Stapler	120.00	50
4	Marker	25.00	80
5	File Folder	60.00	150

Table 2: Suppliers

Structure:

Column Name	Data Type	Constraints
supplier_id	NUMBER(10)	PRIMARY KEY
supplier_name	VARCHAR2(50)	NOT NULL
contact_no	VARCHAR2(20)	UNIQUE
product_id	NUMBER(10)	FOREIGN KEY REFERENCES Products(product_id)

Sample Data:

supplier_id	supplier_name	contact_no	product_id
101	StationeryMart	9876543210	1
102	PaperWorld	9876500000	2
103	OfficeSupplies	9876512345	3
104	MarkerHub	9876522222	4
105	FileDepot	9876533333	5

Section A: DDL (Data Definition Language)

1. Create a database called Inventory DB.
2. Create a table Products and table Suppliers with the specified columns and constraints:

Section B: DML (Data Manipulation Language)

4. Insert at least 5 rows into the Products table.
5. Insert at least 5 rows into the Suppliers table.
6. Update the stock quantity of product 'Pen' to 120.
7. Delete a supplier with a specific supplier_id.
8. Write a query to rename 'Notebook' to 'NoteBook A4'

Section C: DQL (SELECT Queries)

9. Display all records from the Products table.
10. Display only product_name and price of all products.
11. List all products that have a stock quantity less than 100.
12. Show all products between 20 and 100 price range.
13. Find all suppliers whose contact number starts with '98765'.
14. Find the average price of products.
15. Display the total number of products in the inventory.
16. Show the maximum and minimum stock quantities.
17. Count how many suppliers supply each product.
18. Show all products where price > 50 AND stock_qty > 100.
19. Show all products where price < 20 OR stock_qty < 80.
20. Display suppliers whose supplier_name contains the word 'Mart'
21. List all suppliers along with the product they supply (use INNER JOIN).
22. Display suppliers whose name starts with 'S'.
23. Find products whose name has exactly 5 characters
24. Find suppliers who supply products costing more than 100.

Experiment 2 : ONLINE BOOKSTORE DB

An online book store wants to implement a BOOKSTORE DB for managing their online transactions by using the following tables.

Authors Table

Column Name	Data Type	Constraints
author_id	NUMBER	PRIMARY KEY
first_name	VARCHAR2(50)	NOT NULL
last_name	VARCHAR2(50)	NOT NULL
nationality	VARCHAR2(50)	NULL allowed

Books Table

Column Name	Data Type	Constraints
book_id	NUMBER	PRIMARY KEY
Title	VARCHAR2(50)	NOT NULL
author_id	NUMBER	FOREIGN KEY REFERENCES Authors
publication_year	NUMBER	
Price	NUMBER(10,2)	

Customers Table

Column Name	Data Type	Constraints
customer_id	NUMBER(10)	PRIMARY KEY
first_name	VARCHAR2(50)	NOT NULL
last_name	VARCHAR2(50)	NOT NULL
Email	VARCHAR2(50)	UNIQUE, NOT NULL
Address	VARCHAR2(50)	NOT NULL

Orders Table

Column Name	Data Type	Constraints
order_id	NUMBER(10)	PRIMARY KEY
customer_id	NUMBER(10)	FOREIGN KEY REFERENCES Customers
book_id	NUMBER(10)	FOREIGN KEY REFERENCES Books
order_date	DATE	NOT NULL
quantity	NUMBER(10)	NOT NULL

SAMPLE DATA SET for BOOKSTORE DB

Authors Table

author_id	first_name	last_name	nationality
1	Jane	Austen	British
2	George	Orwell	British
3	Gabriel	Garcia Marquez	Colombian
4	Toni	Morrison	American
5	Mark	Twain	American
6	Harper	Lee	American
7	Fyodor	Dostoevsky	Russian

Books Table

book_id	Title	author_id	publication_year	price
101	Pride and Prejudice	1	1813	12.99
102	1984	2	1949	9.50
103	One Hundred Years of Solitude	3	1967	15.00
104	Beloved	4	1987	11.25
105	Animal Farm	2	1945	8.75
106	Adventures of Huckleberry Finn	5	1884	10.50
107	To Kill a Mockingbird	6	1960	14.00

Customers Table

customer_id	first_name	last_name	Email	address
201	Alice	Smith	alice.s@example.com	12 Oak St, London
202	Bob	Johnson	bob.j@example.com	45 Pine Ave, Oxford
203	Charlie	Brown	charlie.b@example.com	78 Maple Rd, Bristol
204	Diana	Prince	diana.p@example.com	34 Queen St, York
205	Edward	Norton	edward.n@example.com	22 River Ln, Leeds
206	Fiona	Hall	fiona.h@example.com	56 Lake Dr, Bath
207	Greg	Miller	greg.m@example.com	89 Park Ave, Glasgow

Orders Table

order_id	customer_id	book_id	order_date	Quantity
301	201	101	20-07-2025	1
302	202	102	21-07-2025	2
303	201	105	22-07-2025	1
304	203	103	23-07-2025	1
305	204	106	24-07-2025	1
306	205	107	25-07-2025	3
307	206	104	26-07-2025	2

Section A: DDL (Schema Design & Constraints)

1. Write SQL statements to create all 4 tables (Authors, Books, Customers, Orders) with:
   - Primary Keys
   - Foreign Keys
   - Appropriate data types
   - NOT NULL constraints where necessary.
2. Alter the Books table to add a constraint that price must be greater than 0.
3. Add a new column phone_number to the Customers table VARCHAR2 (15)) and ensure it is unique.
4. Drop the phone_number column from the Customers table.

Section B: DML (Data Manipulation)

5. Insert at least 7 records for each table (use sample dataset above).
6. Update the price of the book titled *Animal Farm* by increasing it by 10%.
7. Delete all orders made before 2025-07-21.
8. Change the nationality of Gabriel Garcia Marquez to “Latino-American”.

Section C: SELECT Queries (Data Querying)

9. List all books published between 1900 and 2000.
10. Find all customers whose email contains “example.com”.
11. Retrieve books whose price is between 10 and 15 and published before 1950.
12. Show authors who are either ‘British’ or ‘American’.
13. Find books that have a price less than 10 or are published after 1980.
14. Show all books sorted by price in descending order.
15. List authors in alphabetical order by last_name.
16. Display orders sorted by order_date (latest first).

Use of Date Functions

17. Show all orders placed in July 2025.
18. Show all orders with an estimated delivery date (5 days after order date).
19. Show customers who placed an order on a weekend.
20. Calculate how many days have passed since the last order was placed.

Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)

21. Count the total number of books in the database.
22. Find the average price of all books.
23. Show the highest-priced book.
24. Count how many orders each customer has placed.
25. Calculate the total sales (price × quantity) for each customer.

GROUP BY and HAVING

26. Count how many books are written by each author.
27. Group orders by customer_id and display total quantity ordered.
28. Show customers who have ordered more than 2 books in total (use HAVING).
29. Find the total number of books sold per author (GROUP BY author).

Experiment 3: EMPLOYEE DB

An enterprise wants to automate its employee management process by implementing an Employee Database.The goal is to replace manual record-keeping with a centralized system that stores employee, department, and project details. Use the following table structures and data set to implement Employee DB.

EmployeeDB – Table Structures

1. Departments Table

Column	Type	Constraints
dept_id	NUMBER(5)	PRIMARY KEY
dept_name	VARCHAR2(50)	UNIQUE, NOT NULL
location	VARCHAR2(50)	NOT NULL

2. Employees Table

Column	Type	Constraints
emp_id	NUMBER(5)	PRIMARY KEY
first_name	VARCHAR2(50)	NOT NULL
last_name	VARCHAR2(50)	NOT NULL
email	VARCHAR2(50)	UNIQUE, NOT NULL
phone	VARCHAR2(50)	CHECK (phone LIKE '--____')
hire_date	DATE	NOT NULL
job_title	VARCHAR2(50)	NOT NULL
salary	NUMBER(10,2)	CHECK (salary > 0)
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)

3. Projects Table

Column	Type	Constraints
project_id	NUMBER(10)	PRIMARY KEY
project_name	VARCHAR2(50)	NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NULL
dept_id	NUMBER(5)	FOREIGN KEY REFERENCES Departments(dept_id)

4. Employee_Project Table (Many-to-Many)

Column	Type	Constraints
emp_id	INT	FOREIGN KEY REFERENCES Employees(emp_id), PRIMARY KEY(emp_id, project_id)
project_id	INT	FOREIGN KEY REFERENCES Projects(project_id)
hours_allocated	INT	CHECK (hours_allocated > 0)

Sample Data Set

Departments Table

dept_id	dept_name	Location
1	HR	New York
2	IT	San Francisco
3	Finance	Chicago
4	Marketing	Boston
5	Operations	Seattle
6	Legal	Washington D.C.
7	Sales	Dallas
8	R&D	Austin
9	Procurement	Denver
10	Customer Care	Miami

2. Employees Table

emp_id	first_name	last_name	Email	phone	hire_da te	job_title	salary	dept_id	manager_i d
101	Alice	Johnson	alice.j@ corp.com	123-456-7890	15-03-2020	HR Manager	75000	1	NULL
102	Bob	Smith	bob.s@ corp.com	234-567-8901	20-05-2019	IT Analyst	65000	2	104
103	Charlie	Brown	charlie.b@ corp.com	345-678-9012	01-10-2021	Finance Executive	58000	3	106
104	Diana	Prince	diana.p@ corp.com	456-789-1234	07-12-2018	IT Manager	90000	2	NULL
105	Ethan	Hunt	ethan.h@ corp.com	567-890-1234	25-02-2022	Marketing Lead	62000	4	NULL
106	Fiona	Hall	fiona.h@ corp.com	678-901-2345	11-01-2017	Finance Manager	85000	3	NULL
107	Greg	Miles	greg.m@ corp.com	789-012-3456	15-04-2023	IT Manager	45000	2	104
108	Hannah	White	hannah.w@ corp.com	890-123-4567	09-05-2021	HR Executive	50000	1	101
109	Ian	Scott	ian.s@ corp.com	901-234-5678	20-11-2020	Operation s Analyst	56000	5	NULL
110	Julia	Adams	julia.a@corp.com	012-345-6789	18-12-2019	Legal Advisor	70000	6	NULL

3. Projects Table

project_id	project_name	start_date	end_date	dept_id
201	Payroll System	01-01-2023	NULL	3
202	Website Upgrade	10-02-2023	NULL	2
203	Recruitment Drive	05-03-2023	NULL	1
204	Ad Campaign	20-05-2023	NULL	4
205	New CRM Tool	15-04-2023	NULL	7
206	Compliance Portal	10-06-2023	NULL	6
207	Inventory System	01-07-2023	NULL	5
208	AI Research	05-08-2023	NULL	8
209	Customer Feedback	10-09-2023	NULL	10
210	Procurement System	01-10-2023	NULL	9

4. Employee_Project Table

emp_id	project_id	hours_allocated
102	202	120
104	202	80
103	201	100
106	201	150
101	203	50
105	204	70
107	202	60
109	207	90
110	206	110
108	203	40

Section A: DDL (Schema Creation & Modification)

1. Write SQL statements to create the above tables with the specified constraints
2. Alter the Employees table to add a column bonus NUMBER(10,2)(8,2) with default value0.
3. Drop the column bonus from Employees.

Section B: DML (Insert, Update, Delete)

4. Insert at least 10 rows into Departments, Employees, Projects, and Employee_Project.(use the above data set)
5. Try inserting an employee with a negative salary (should fail due to CHECK constraint).
6. Update the salary of the employee with emp_id = 103 by 15%.
7. Delete an employee record who has resigned (choose any emp_id).
8. Increase all employees' salaries in the IT department by 5%.
9. Change the department of an employee to "Research".(should fail due to FK constraint)

Section C: DQL (Select Queries)

10. List all employees and their details.
11. Show all employees in the "HR" department.
12. Find employees with salaries between 50,000 and 80,000.
13. Retrieve employees hired after 2020.
14. Show employees who are in either the IT or Finance department.
15. Find employees whose email ends with "@corp.com".
16. List all employees with salary > 60,000 AND located in "New York".
17. Display employees in descending order of salary.
18. Count the number of employees in each department.
19. Show the average salary of employees department-wise.
20. Display departments where the average salary is greater than 70,000.
21. Find the number of employees in each project.
22. Display departments with more than 3 employees.
23. Show the sum of all salaries department-wise.
24. List all distinct department IDs from the Employees table.
25. Show employee names with the year they were hired.
26. Show employees grouped by the year of hire.
27. List employees hired in the last 90 days.
28. List the no of years of experience of all the employees

Section D: Joins

29. List all employees with their department names (INNER JOIN).
30. Display all departments along with employees, including those departments without employees (LEFT JOIN).
31. Show employees and the projects they are working on (JOIN 3 tables: Employees, Employee_Project, Projects).
32. List projects along with total hours allocated by employees.
33. Write a query to find employees who are working on more than one project.
34. Show all projects handled by the 'Finance' department.

Section E: PL/SQL Programming

1. Write a PL/SQL program to find factorial of a number.
2. Write a PL/SQL program to find sum of digits of an integer
3. Write a PL/SQL Program to demonstrate a for loop.
4. Write a PL/SQL Program to demonstrate procedures.
5. Write a PL/SQL Program to demonstrate Aggregate functions.
6. Write a procedure GetEmpInfo that takes emp_id as input and displays name, salary, and department.
7. Write a PL/SQL block that checks if an employee's salary is above 50,000. If yes, print "High Salary" ;Otherwise print "Standard Salary".
8. Write a PL/SQL program to display the top 10 rows in the Emp table based on their job and salary
9. Write a stored procedure GiveBonus that takes department ID and a designation as input, along with a bonus amount, and updates the salary of all employees in that department who have the specified designation by adding the bonus amount to their current salary.

SEMESTER – II
COURSE 4: DATABASE MANAGEMENT SYSTEM
Model Question Paper

Time: 3 hours Max. Marks: 70

SECTION - A

Answer any FIVE of the following questions.
(Marks: 5x4 marks =20 marks)

1. Explain data and information.
2. Write about keys used in relational model..
3. Write about Relational Degree.
4. What is functional dependency?
5. Write about SQL Literals.
6. Discus Table modification command.
7. What are aggregate functions? Explain any three functions with an example
8. What are the uses of Grant and Revoke commands?
9. Write the structure of PL/SQL program.
10. Define variable in PL/SQL. Explain declaration of variables.

SECTION - B

Answer FIVE questions. Choosing ONE question from each unit.
(Marks: 5x10 marks =50 marks)

UNIT-I

11. Write the advantages of Database Management System.

(Or)

12. Write the Codd's relational database rules.

UNIT-II

13. Write the components of an Entity–Relationship model.

(Or)

14. What is normalization? Discuss about Normal forms.

UNIT-III

15. Write about DDL commands with syntax and example.

(Or)

16. Discuss TCL commands with an example.

UNIT-IV

17. Write about Sub query and correlated query.

(Or)

18. What is an operator? Write about Relational Set operators.

UNIT-V

19. Write about data types in PL/SQL.

(Or)

20. Write about conditional control statements.

Instruction to Paper Setter:

Two questions must be given from each unit in Section-A.