



KAKATIYA UNIVERSITY, WARANGAL-506009

DEPARTMENT OF COMPUTER SCIENCE

Scheme with effect from the academic year 2025-2026

B.SC. (COMPUTER SCIENCE)

YEAR/ Semester	Type of Course	Paper Title	Hrs Per Week T: Theory P: Practical	No. of Credits	Max. Marks		
					Intr. Marks	External marks	Total marks
I/I	DSC-1	Programming in C	4 (T)	4	20	80	125
	DSC-1	Programming in C Lab	2 (P)	1	--	25	
I/II	DSC-2	Data Structures using C	4 (T)	4	20	80	125
	DSC-2	Data Structures using C Lab	2 (P)	1	--	25	
II/I	DSC-3	Data Base Management System	4 (T)	4	20	80	125
	DSC-3	Data Base Management System Lab	2 (P)	1	--	25	
II/II	DSC-4	Programming in Java	4 (T)	4	20	80	125
	DSC-4	Programming in Java Lab	2 (P)	1	--	25	

B. Sc. Compute Science

SEMESTER – I

Course type	Paper Title	Hours per week	Marks	
DSC-1	Programming in C	Theory: 04	Internal	external
		Credit: 04	20	80

Unit- I

Overview of programming languages: Generation and classification. Processes involved in program execution: Compilation, interpretation, loading, and linking. Flow charts and Algorithms. **Basics of C Programming:** Introduction to C programming language. Structure of a C program, C tokens, data types, variables, constants, operators, expression evaluation (precedence, associativity), type conversions in C.

Unit- II

Input and Output: Non-formatted and formatted input/output functions. Escape sequences and their usage in I/O. Control Statements: Selection statements if, if-else, nested if, switch, conditional operators. Iterative statements while, for, do-while. Special control statements goto, break, continue, return, exit. **Arrays and Strings:** One-dimensional arrays, Multidimensional arrays character arrays, ctype functions and string functions.

Unit -III

Functions: Function definition, declaration, and calling mechanisms, types of functions, Call-by-value, call-by-reference. Passing arrays to functions, recursion, inline functions. Scope and lifetime of variables, storage classes. **Pointers:** Introduction to pointers, address-of operator (&), pointer types. Pointers and arrays, pointers and strings, pointer to pointer, array of pointers. Dynamic memory allocation malloc, calloc, free.

Unit -IV

User-Defined Data Types: Structures and unions: Definition, initialization, accessing members. Arrays of structures, structures vs. unions. Enumeration types (enum). **File handling:** Introduction to file operations in C. File functions open, close, read, and writes. Working with text and binary files.

Textbook:

1. Pradip Dey, Manas Ghosh, Computer Fundamentals and Programming in C (2e)
2. Reema Thareja, "Programming in C", Oxford University Press, Second Edition, 2016.
3. Kernighan, B.W and Ritchie, D.M, "The C Programming language", Second Edition, Pearson Education, 2015.

References:

1. Ivor Horton, Beginning C
2. Ashok Kamthane, Programming in C
3. Herbert Schildt, The Complete Reference C
4. Paul Deitel, Harvey Deitel, C How to Program
5. Byron S. Gottfried, Theory and Problems of Programming with C
6. Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language
7. B. A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C

Course type	Paper Title	Hours per week	Marks	
DSC-1	Programming in C Lab	Practical: 02	Internal	external
		Credits: 01	--	25

Lab Exercises:

The practical sessions involve implementing and testing programs to reinforce the concepts learned in the theory. Recommended Software for Practical: GCC on Linux, DevC++ or Code Blocks on Windows 10. External Examination Requirements: Students will need to demonstrate the execution of two programs in the external lab exam.

List of Programs:

1. Write a C program to input three numbers and find the largest of two or three numbers using if statements and the conditional (ternary) operator (? :). Display the largest number.
2. Write a C program that takes an integer input and outputs the reversed number.
3. Write a C program to print all prime numbers between 2 and a given number n.
4. Write a C program to find the roots of a quadratic equation $ax^2+bx+c=0$ $ax^2 + bx + c = 0$.
5. Write a C program to print a **triangle** pattern of stars (*), where the number of lines is input by the user.
6. Write a C program to find the largest and smallest elements in an array of n numbers.
7. Write a C program to multiply two matrices.
8. Write a C program to find the **Greatest Common Divisor (GCD)** of two numbers using both **iteration** and **recursion**.
9. Write a C program to demonstrate the use of different **storage classes** (auto, register, static, extern).
10. Write a C program to demonstrate the concepts of **call-by-value** and **call-by-reference**.
11. Write a C program that takes a string from the command-line arguments and counts the occurrence of each alphabet letter in the string.
12. Write a C program to demonstrate the usage of the enum data type.
13. Write a C program that demonstrates various string functions from the <string.h> library.
14. Write a C program that opens a file and counts the total number of characters in it.
15. Write a C program that defines a structure Student and stores information of students in a file.
16. Write a C program that copies content from an existing text file to a new file.

B. Sc. Compute Science
SEMESTER – II

Course type	Paper Title	Hours per week	Marks	
DSC-2	Data Structures Using C	Theory: 04	Internal	external
		Credits: 04	20	80

UNIT – I

Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis of an algorithm- Time and Space Complexity, Asymptotic Notations. **Arrays:** Arrays – ADT, Polynomials, Sparse matrices, Strings-ADT, Pattern Matching. **Stacks:** Stack ADT, Stacks using Arrays, Stacks using dynamic arrays, Evaluation of Expressions – Evaluating Postfix Expression, Infix to Postfix expression, Checking well formed Parenthesis, Reversing a string.

UNIT – II

Queues: Queues ADT, operations, Circular Queues, **Applications Linked Lists:** Singly Linked Lists and Chains, Linked Stacks and Queues, Polynomials, Operations for circularly linked lists, Equivalence Classes, Sparse matrices, Doubly Linked Lists. **Hashing:** Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques.

UNIT – III

Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps, **Binary Search trees (BST):** Definition, Searching an element, Insertion into a BST, and Deletion from a BST. Efficient Binary Search Trees: **AVL Trees:** Definition, Searching an element, Insertion into a AVL, **Graphs:** Graph Abstract Data Type, Elementary Graph operations (DFS and BFS), Minimum Cost Spanning Trees (Prim's and Kruskal's Algorithms).

UNIT – IV

Sorting and Searching: Sequential search ,Binary Search, Insertion sort, Quick sort, Best computing time for Sorting, Merge sort, Heap sort, shell sort, Sorting on Several Keys, List and Table Sorts, Summary of Internal Sorting, Linear and Binary Search algorithms.

Suggested Books:

1. Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals of Data structures in C, 2nd Edition (2008), Universities Press.

Reference Books:

1. Mark A Weiss, Data Structures and Algorithm Analysis In C, Second Edition (2002), Pearson.
2. Kushwaha D. S and Misra A.K, Data structures A Programming Approach with C, Second Edition (2014), PHI.
3. Gilberg R. F and Forouzan B. A, Data structures: A Pseudocode Approach with C, Second Edition (2007), Cengage Learning
4. Tanenbaum A. M , Langsam Y. Augenstein M. J, Data Structures using C, Second Edition (2008), Pearson.

Course type	Paper Title	Hours per week	Marks	
DSC-2	Data Structures Using C Lab	Practical: 02	Internal	external
		Credits: 01	--	25

List of Experiments (Using C programming Language):

1. Implementation of Stacks and Queues using Arrays.
2. Implementation of Circular Queue.
3. Implementation of Infix to Postfix Conversion, Postfix Expression Evaluation.
4. Implementation of Singly Linked List
5. Implementation of Doubly Linked List.
6. Implementation of Circular Linked List.
7. Implementation of Stacks, Queues using Linked Lists.
8. Implementation of Binary Search and Hashing
9. Implementation of Operations on Binary Tree (Delete Entire Tree, Copy Entire Tree, Mirror Image, Level Order, Search for a Node etc.)
10. Implementation of Tree Traversals on Binary Trees.
11. Implementation of Binary Search Tree. (Insertion, Deletion and Search operations)
12. Implementation of operations on AVL Trees.
13. Implementation of Traversal on Graphs.
14. Implementation of Prim's and Kruskals Algorithm.
15. Implementation of Selection, Merge, Quick, Heap, and Insertion Sort.

B. Sc. Compute Science
SEMESTER – III

Course type	Paper Title	Hours per week	Marks	
DSC-3	Data Base Management System	Theory: 04	Internal	external
		Credits: 04	20	80

Unit-I

Introduction to Database Management System: Data, Information, Metadata, Database, Database Management System, File Based System, Drawbacks of File-Based System, The Database approach, Components of Database Environment, Advantage of DBMS, Three-Tier Architecture, Database Languages, Data Models, Database Users, Database Administrator, Types of Databases.

Unit -II

Entity-Relationship Model: Introduction, Components of ER Diagrams, Entities, Attributes, Keys, Strong and Weak entity sets, Relationships, Types of Entities, Types of Attributes, Types of Relationships, Degree of a relationship, Cardinality. Generalization and Specialization, Aggregation and Composition. **Normalization:** Data Redundancy, Functional Dependencies, Basic Normal Forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), Boyce Codd Normal Form (BCNF), De-normalisation.

Unit-III

Structured Query Language: Basic form of SQL query, Data Types, Integrity Constraints, Null Values, Commit, rollback, Logical operators, Special Operators, Order by, Group by and Having clause, Aggregate Functions, Set Operators, **Alias**, Views, Joins, Nested Sub queries, Correlated Sub queries, Views: Create a view, Delete a View. **PL/SQL:** Introduction, Structure of PL/SQL, Elements, Data Types, Control Structures, Steps to Create a PL/SQL Program, Iterative Control Cursors, Steps to Create a Cursor, Procedures, Functions, Packages, Exception Handling, Database Triggers, Types of triggers.

Unit -IV

Transaction Management: What is transaction, Properties of Transactions, Database, Concurrency control, serializability, recoverability, Concurrency control with locking methods, concurrency control with time stamping methods, Concurrency control with optimistic methods, Deadlock. Database Recovery: The Need for Recovery, Transactions and Recovery, Recovery Facilities, Recovery Techniques. Database Security: Threats, Computer Based Controls: Authorization, Access Controls, Views, Backup and Recovery, Integrity, Encryption, RAID.

Text books:

1. Abraham Silberschatz, H. Korth and S. Sudarshan, Database System Concepts, 6th Ed., Tata McGraw Hill, 2011
2. Thomas M. Connolly, Carolyn E. Begg, Database Systems—A Practical Approach to Design, Implementation, and Management (6e)
3. Ivan Bayross, **SQL**, PL/SQL: The Programming Language of Oracle, 4th Revised Edition with CD-ROM by Ivan Bayross, BPB Publications.

Course type	Paper Title	Hours per week	Marks	
DSC-3	Data Base Management System Lab	Theory: 02	Internal	external
		Credits: 01	--	25

1. Create a database having three tables to store the details of students of Computer Department in your college: **Student** (Roll number, Name of student, Date of birth, Address, Marks, Phone number) **Paper** (Paper code, Name of the Paper) **Attendance** (College roll number, Paper Code, Attendance, Marks).

- Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
- Design a query that will return the records from the table name **Paper** along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper2.
- List all students who live in “Warangal” and have marks greater than 60 in paper1.
- Find the total attendance and total marks obtained by each student.
- List the names of all the students who secured highest marks in paper2.

2. Create the following tables, enter at least 5 records in each table and answer the queries given below:

Employee (Person_Name, Street, City)

Works (Person_Name, Company_Name, Salary)

Company (Company_Name, City)

Manages (Person_Name, Manager_Name)

- Identify primary and foreign keys.
 - Alter table employee, add a column “email” of type varchar(20).
 - Find the name of all managers who work for both Samba Bank and NCB Bank.
 - Find the names, street address and cities of residence and salary of all employees who work for “Samba Bank” and earn more than \$10,000.
 - Find the names of all employees who live in the same city as the company for which they work.
 - Find the highest salary, lowest salary and average salary paid by each company.
 - Find the sum of salary and number of employees in each company.
 - Find the name of the company that pays highest salary.
3. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Suppliers (SNo, Sname, Status, SCity)

Parts (PNo, Pname, Colour, Weight, City)

Project (JNo, Jname, Jcity)

Shipment (Sno, Pno, Jno, Qunatity)

- Identify primary and foreign keys.
- Get supplier numbers for suppliers in Paris with status>20.
- Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.
- Get suppliers names for suppliers who do not supply part P2.
- For each shipment get full shipment details, including total shipment weights.
- Get all the shipments where the quantity is in the range 300 to 750 inclusive.
- Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.
- Get the names of cities that store more than five red parts.
- Get full details of parts supplied by a supplier in Hyderabad.

- j) Get part numbers for part supplied by a supplier in Warangal to a project in Chennai.
- k) Get the total number of project supplied by a supplier (say, S1).
- l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).
- 4. Write a PL/SQL Program to demonstrate Procedure.
- 5. Write a PL/SQL Program to demonstrate Function.
- 6. Write a PL/SQL program to Handle Exceptions.
- 7. Write a PL/SQL Program to perform a set of DML Operations.
- 8. Create a View using PL/SQL program.
- 9. Write a PL/SQL Program on Statement Level Trigger.
- 10. Write a PL/SQL Program on Row Level Trigger.

B. Sc. Compute Science
SEMESTER – IV

Course type	Paper Title	Hours per week	Marks	
DSC-4	Programming in Java	Theory: 04	Internal	external
		Credits: 04	20	80

UNIT – I

Introduction to Java: Features of Java, bytecode, Structure of java program, language Fundamentals – tokens (comments, Identifiers, Keywords), data types, variables and types of variables (reference, local, static, final), command line arguments, operators, expressions, type conversion and casting, control statements. **Object-Oriented Programming (OOP) Basics** Classes and Objects - Concepts of classes, objects, constructors, methods (types of methods), access control (public, private, protected and default). this keyword, garbage collection, nested and inner classes, String, StringBuffer and StringBuilder.

UNIT – II

Inheritance: Inheritance concept, types of inheritance, Member access rules, use of super and final. **Abstract** classes and methods, **Polymorphism** - method overriding(**Runtime-time Polymorphism**) and method overloading(**Compile-time Polymorphism**), overloading constructors. **Interfaces:** Defining an interface, implementing interfaces, extending interface. **Packages:** inbuilt packages, user defined packages - Defining, Creating and Accessing a Package, importing packages. **Exception handling:** Benefits of exception handling, Exception hierarchy, exception Vs error, classification, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, re throwing exceptions, built in exceptions, Implementing custom exceptions.

UNIT – III

Multithreading: Java Thread Model – Thread, Thread Life Cycle, The Main Thread, Ways to create a Thread – Implementing Runnable interface, Extending Thread class, creating multiple threads, thread priorities, synchronization. **Collections:** Overview of Java Collection framework, **java.util** - Collection classes – Array List, Linked List, Hash Set, Tree Set, Collection Interfaces – Collection, List, Set, Accessing Collection via iterator, working with Map. Other Utility classes: String Tokenizer, Date, Calendar, Gregorian calendar. Scanner. **Java Input/Output:** exploring java.io, Java I/O classes and interfaces, File, Stream classes, byte stream, character stream, serialization.

UNIT – IV

Swing: JFrame, JLabel, ImageIcon, JTextField, JPasswordField, the Swing buttons, JTabbedPane, JScrollPane, JList, JComboBox, Event Handling: Event Classes, Source of Events, Event Listener Interfaces. Handling mouse and keyboard events. Introduction to **JavaFX** and Setup, Scene Graph (root, children, hierarchy). layout containers –Vbox, HBox, GridPane, StackPane, Buttons, controls, Styling in JavaFX, Animation and Transitions in JavaFX.

Suggested Readings:

1. Herbert Scheldt, “The Complete Reference Java, 11th Edition, Tata McGraw Hill, 2021.
2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002.
3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 6th Edition, McGraw Hill Publishing, 2019.
4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education

Course type	Paper Title	Hours per week	Marks	
DSC-4	Programming in Java Lab	Practical: 02	Internal	external
		Credits: 01	--	25

List of Experiments:

1. Write a Java program to illustrate the concept of class with method overloading.
2. Write a Java Program that reads a line of integers, displays each integer, and find the sum of all the integers.
3. Write a Java program to illustrate the concept of Single level and Multi level Inheritance.
4. Write a Java program to demonstrate the Interfaces & Abstract Classes.
5. Write a Java program to implement the concept of exception handling.
6. Write a Java program to illustrate the concept of threading using Thread Class and runnable Interface.
7. Write a Java program to illustrate the concept of Thread synchronization.
8. Write a Java program to illustrate collection classes like Array List and Linked List.
9. Write a Java program to illustrate the concept of I/O Streams.
10. Write a Java program to implement serialization concept.
11. Write a Java applet program for handling mouse & key events.
12. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
13. create a simple user profile form with fields for name, age, and a profile picture using Swing.
14. Develop a simple JavaFX application that demonstrates the following core concepts
 - a. Utilize a VBox as the root node to establish a basic scene graph hierarchy
 - b. Incorporate HBox, GridPane, and StackPane layout containers to arrange various UI elements.
 - c. Include Button, Label, and TextField controls for user interaction
15. Apply inline CSS styling to customize the appearance of at least one button