



VIJAYANAGARA SRI KRISHNADEVARAYA UNIVERSITY
JNANASAGARA CAMPUS, BALLARI-583105

**Department of Studies in
Computer Science**

I & II Semester Syllabus

BACHELOR OF COMPUTER SCIENCE

Programme as per State Education Policy 2024

Under Choice Based Credit System (CBCS)

With effect from 2024-25 and onwards

Title of Subject: PROGRAMMING IN C (Major – 3)		
COURSE CODE: 24MJCOMP1L	CIA Marks: 20	
SEMESTER: I	SEE Marks: 80	
Contact Hours: (L:T:P): 4-0-0	Credit: 04	Duration of Exam: 03
<p>Course Outcomes:After completing this course satisfactorily, a student will be able to:</p> <ol style="list-style-type: none"> 1. Confidently operate Desktop Computers to carry out computational tasks 2. Understand working of Hardware and Software and the importance of operating systems 3. Understand programming languages, number systems, peripheral devices. 4. Read, understand and trace the execution of programs written in C language 5. Write the C code for a given problem 6. Perform input and output operations using programs in C 7. Write programs that perform operations on arrays, functions and structures. 		
UNIT – I:		10 Hours
<p>Fundamentals of Computers:Computer Definition, Characteristics of Computers, Basic Organisation of a Digital Computer; Input devices: - Keyboard, Mouse, Light pen, Joystick, Scanner, Digitizer. Output devices- Various types of printers, Plotters, Types of Computers, Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software– System Software, application software and Utility Software; Computer Languages - Machine Level, Assembly Level & High-Level Languages, Translator Programs –Assembler, Interpreter and Compiler; Planning a Computer Program-Algorithm, Flow chart and Pseudo code with Examples.</p>		
UNIT – II:		12 Hours
<p>Introduction to C Programming:Introduction to C Programming, History and Features of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C.</p> <p>C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants.</p> <p>Input and output with C: Formatted I/O functions - printf and scanf, control stings and escape sequences, output specifications with printf functions; Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and puts functions.</p>		
UNIT – III:		12 Hours
<p>C Operators & Expressions:Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Expression, Types of Expression, Evaluation of arithmetic Expressions; Type conversion, Mathematical functions.</p> <p>Control Structures: Decision Making statements- Simple if, if_else, nested if_else,else_if ladder, Switch-case, goto. Looping Statements - Entry controlled and exit controlled statements, while, do-while, for loops, Nested loops, break & continue statements.</p>		
UNIT – IV:		10 Hours
<p>Arrays:One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation.</p> <p>Strings: Declaring & Initializing string variables; String handling functions strlen, strcmp,strcpy and strcat;table of strings. Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.</p>		

Pointers: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;

UNIT – V:

12 Hours

User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type.

User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition, defining unions, declaring union variables, accessing union members; difference between Structures and Unions.

Text Books:

1. Balaguruswamy: Programming in ANSI C, Tata McGraw-Hill.
2. Brian W. Kernighan and Dennis M. Ritchie: The C Programming Language, PHI

Reference:

1. V. Rajaraman: Fundamentals of Computers, PHI(EEE).
2. Kamthane, Programming with ANSI and Turbo C. Pearson Education, Asia.
3. Herbert Schildt: C. The complete reference, 4th edition.
4. Yeshwant Kanetkar: Let us C, BPB Publications.
5. Rajesh Hongal Computer Concepts and C Programming.

Title of Subject: PROGRAMMING IN C Lab (Major – 3)		
COURSE CODE: 24MJCOMP1P	CIA Marks: 10	
SEMESTER: I	SEE Marks: 40	
Contact Hours: (L:T:P): 0-0-4	Credit: 02	Duration of Exam: 03

Part A

1. Number System Conversion

- Write a program to convert a given decimal number to binary, octal, and hexadecimal.
- Write a program to convert a given binary number to decimal, octal, and hexadecimal.

2. Simple C Program

- Write a program to display "Hello, World!" and understand the structure of a C program.

3. Creating and Executing a C Program

- Write a C program to add two numbers and display the result. Practice compiling and executing the program.

4. Formatted and Unformatted I/O Functions

- Write a program to read and display a string using formatted I/O functions (printf, scanf).
- Write a program to read and display a single character and a string using unformatted I/O functions (getchar, putchar, gets, puts).

5. Arithmetic Operators

- Write a program to demonstrate the use of arithmetic operators (+, -, *, /, %).

6. Relational and Logical Operators

- Write a program to demonstrate the use of relational (==, !=, >, <, >=, <=) and logical operators (&&, ||, !).

7. Control Structures

- Write a program to find the largest of three numbers using if_else statements.

- Write a program to print the grade of a student based on marks using switch-case statements.

8. Looping Statements

- Write a program to print the first 10 natural numbers using while, do-while, and for loops.
- Write a program to calculate the factorial of a number using a for loop.

Part B

1. One Dimensional Arrays

- Write a program to read and display elements of an array.
- Write a program to find the sum and average of elements in an array.

2. Two Dimensional Arrays

- Write a program to read and display elements of a 2D array (matrix).
- Write a program to perform matrix addition.

3. String Handling

- Write a program to demonstrate the use of string handling functions (strlen, strcmp, strcpy, strcat).

4. Pointer Basics

- Write a program to declare and initialize pointers and access the value and address of variables using pointers.
- Write a program to demonstrate pointer arithmetic.

5. User Defined Functions

- Write a program to find the square of a number using a user-defined function with parameters and return type.
- Write a program to calculate the sum of two numbers using a user-defined function without parameters and return type.

6. Structures

- Write a program to define a structure for a student (containing fields for roll number, name, and marks) and demonstrate initialization and access of structure members.
- Write a program to compare two structure variables for equality.

7. Array of Structures

- Write a program to create an array of structures for students and display their details.

8. Unions

- Write a program to define a union for different data types (int, float, char) and demonstrate initialization and access of union members.
- Write a program to show the difference between structures and unions in terms of memory usage.

Evaluation Scheme for Lab Examination:

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	07
	Execution	08
Program -2 from Part B	Writing the Program	07
	Execution	08
Practical Record		05
Viva-Voce		05
Total		40

Title of Subject: DATA STRUCTURES USING 'C' (Major – 3)		
COURSE CODE: 24MJCOMP2L	CIA Marks: 20	
SEMESTER: II	SEE Marks: 80	
Contact Hours: (L:T:P): 4-0-0	Credit: 04	Duration of Exam: 03
<p>Course Outcomes:After completing this course satisfactorily, a student will be able to:</p> <ol style="list-style-type: none"> 1. Describe how arrays, linked structures, stacks, queues, trees are represented in memory and used by algorithms 2. Describe common applications for arrays, linked structures, stacks, queues, trees 3. Write programs that use arrays, linked structures, stacks, queues, trees, 4. Demonstrate different methods for traversing trees 5. Compare alternative implementations of data structures with respect to performance 6. Describe the concept of recursion; give examples of its use. 7. Discuss the computational efficiency of the principal algorithms for sorting, searching. 		
UNIT – I		12 Hours
<p>Introduction to Data structures:Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Algorithm Specification, Performance Analysis, Performance Measurement</p> <p>Recursion: Definition; Types of recursions; Recursion Technique Examples - Fibonacci numbers, GCD, Factorial, Comparison between iterative and recursive functions.</p>		
UNIT – II		12 Hours
<p>ArraysBasic Concepts, Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing line arrays; Inserting and deleting elements; Sorting: Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching: Sequential Search, Binary search; Iterative and Recursive searching; multidimensional Arrays, representation of multidimensional Arrays, Sparse matrices,</p>		
UNIT – III		12 Hours
<p>Stacks: Basic Concepts, Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression, Tower of Hanoi.</p> <p>Queues: Basic Concepts, Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;</p>		
UNIT – IV		10 Hours
<p>Dynamic Memory Allocation:Static & Dynamic memory allocation; Memory allocation and de-allocation functions - malloc, calloc, realloc and free.</p> <p>Linked list: Basic Concepts, Definition and Representation of linked list, Types of linked lists - Singly linked list, doubly liked list, Header liked list, Circular linked list; Representation of single Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.</p>		
UNIT – V		10 Hours
<p>Trees:Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; pre order, in order and Post order traversal; Reconstruction of a binary tree when any two of the traversals are given.</p>		

Text Books:

1. Kamthane: Introduction to Data Structure in CL Pearson Education 2005.
2. Langsam, Ausenstein Maoshe & M. Tanenbaum Aaron, Data Structure using C and C++ Pearson Education.

References Books:

1. Weiss: Data Structure and Algorithm Analysis in C, IInd Edition, Pearson Education.
2. Lipschutz: Schaum's outline series Data Structures, Tata McGraw Hill.
3. Tenenbaum: Data Structures using C, Pearson Education

Title of Subject: DATA STRUCTURES LAB (Major – 3)		
COURSE CODE: 24MJCOMP2P	CIA Marks: 10	
SEMESTER: II	SEE Marks: 40	
Contact Hours: (L:T:P): 0-0-4	Credit: 02	Duration of Exam: 03

Part-A

1. **Fibonacci Numbers using Recursion**
 - Write a program to generate Fibonacci numbers using a recursive function.
2. **GCD using Recursion**
 - Write a program to find the greatest common divisor (GCD) of two numbers using recursion.
3. **Factorial using Recursion**
 - Write a program to calculate the factorial of a number using a recursive function.
4. **Comparison of Iterative and Recursive Techniques**
 - Write programs to compute Fibonacci numbers, GCD, and factorial using both iterative and recursive methods. Compare their performance.
5. **Array Operations**
 - Write a program to demonstrate basic operations on arrays (traversing, inserting, and deleting elements).
6. **Sorting Algorithms**
 - Implement Selection Sort, Bubble Sort, Quick Sort, and Insertion Sort on an array and compare their performance.
7. **Searching Algorithms**
 - Implement Sequential Search and Binary Search on an array using both iterative and recursive approaches.
8. **Multidimensional Arrays and Sparse Matrices**
 - Write a program to perform operations on multidimensional arrays and demonstrate the representation of sparse matrices.
9. **Stack Operations**
 - Implement stack operations (push, pop, display) and use the stack to convert infix expressions to postfix.
10. **Evaluation of Postfix Expressions**
 - Write a program to evaluate a postfix expression using a stack.

Part-B

1. **Queue Operations**
 - Implement simple queue operations (enqueue, dequeue, display) and demonstrate the use of circular queues.
2. **Tower of Hanoi using Stack**
 - Write a program to solve the Tower of Hanoi problem using a stack.
3. **Memory Allocation Functions**

- Write a program to demonstrate the use of malloc, calloc, realloc, and free functions for dynamic memory allocation.
- 4. Singly Linked List Operations**
 - Implement operations on a singly linked list (traversing, searching, inserting, and deleting nodes).
- 5. Doubly Linked List Operations**
 - Implement operations on a doubly linked list (traversing, searching, inserting, and deleting nodes).
- 6. Circular Linked List**
 - Write a program to create and perform operations on a circular linked list.
- 7. Binary Tree Traversals**
 - Write a program to create a binary tree and perform pre-order, in-order, and post-order traversals.
- 8. Binary Search Tree (BST) Operations**
 - Implement insertion, deletion, and searching in a binary search tree.
- 9. Heap Tree**
 - Write a program to implement a heap tree and perform heap operations.
- 10. Reconstruction of Binary Tree**
 - Write a program to reconstruct a binary tree when given any two of the traversals (pre-order, in-order, post-order).

Evaluation Scheme for Lab Examination:

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	07
	Execution	08
Program -2 from Part B	Writing the Program	07
	Execution	08
Practical Record		05
Viva-Voce		05
Total		40

**BCA/B.Sc Degree Examination,
SEP – QP - Pattern**

Time: 3 Hours

Max. Marks: 80

Section – A

Note: Answer all sub questions

Each question carries TWO mark.

(10 x 2 = 20)

1.
 - a)
 - b)
 - c)
 - d)
 - e)
 - f)
 - g)
 - h)
 - i)
 - j)

Section – B

Note : Answer any Four questions

Each question carries FIVE marks.

(4 x 5 =20)

- 2.
- 3.
- 4.
- 5.
- 6.
- 7.

Section – C

Note : Answer any Four questions

Each question carries TEN marks.

(4 x 10 =40)

- 8.
- 9.
- 10.
- 11.
- 12.12.
- 13.13.

Note : 1. For Section –A , Two questions from each Unit.

2. For Section – B , One question from each Unit, and Q-7 must be from Unit 2 to 5.

3. For Section – C , One question from each Unit, and Q-13 must be from Unit 2 to 5.

**BCA/B.Sc Degree Examination,
SEP – Scheme for Practical Examination**

1. Writing Two Programs	: 14 Marks (for each 7 marks)
2. Execution of Two programs	: 16 Marks (for each 8 marks)
3. Practical record	: 05 Marks
4. Viva Voce	: 05 Marks
Total	: 40 Marks