



**KARNATAKA STATE AKKAMAHADEVI WOMEN  
UNIVERSITY, VIJAYAPURA**

**Curriculum for  
Bachelor of Science (B.Sc.)  
Computer Science  
As a Major Subject  
Category I: General Degree and  
Category II: Deep Specialization**

**Syllabus for I and II Semesters**

**BoS meeting proceedings dated 24-05-2024**

**2024-25**

## BSc I and II Semesters syllabus w.e.f. 2024-25 onwards

**Category I: General Degree and**

**Category II: Deep Specialization**

Sem	Course Code	Courses	Credits	Hour / Week		Examination			
				Theory	Lab	Term end Examination Max Marks	Formative Assessment Max Marks	Total Marks	Duration of exam Hours
1	24BSCCSMT1.1	Computer Fundamentals and Programming in C	4	04	--	80	20	100	03
	24BSCCSML1.2	Lab.: C Programming	2	--	04	40	10	50	03
2	24BSCCSMT2.1	Data Structures using C	4	04	--	80	20	100	03
	24BSCCSML2.2	Lab.: Data structures using C	2	--	04	40	10	50	03
		Total	12					400	

## BSc Computer Science Major -I Semester

<b>Course Code: 24BSCSMT1.1</b>	<b>Course Title: Computer Fundamentals and Programming in C</b>	
<b>Credits: 4</b>	<b>Hrs/Week: 4 Hrs</b>	<b>Contact Hours: 60</b>
<b>Formative assessment: 20 marks</b>	<b>Term end exam: 80 marks</b>	<b>Exam duration: 03 hrs</b>

**Course Outcomes (COs):** At the end of the course, students will be able to:

- Understanding computer hardware components and software concepts
- Illustrate the flowchart and design an algorithm for a given problem and to develop IC programs using operators
- Writing C programs using looping and conditional statements
- Write user defined functions to solve real world problems
- Understanding pointer concepts and writing c programs using pointers to access arrays, strings and functions
- Writing c programs to store and manipulate complex data using structures and union.

### Unit – 1

**12 Hrs**

**Fundamentals of Computers:** Introduction – History of Computers, Generations of Computers - Classification of Computers-Basic Anatomy of a Computer System-Input Devices - Processor-Output Devices-Memory Management – Types of Software- Overview of Operating System-Programming Languages, Translator, Programs-Problem Solving Techniques, Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.

### Unit – 2

**12 Hrs**

**Numbers System-** Binary, Decimal, Octal and Hexadecimal, Conversion from one number system to another, Computer Codes –BCD, Gray Code, ASCII and Unicode formats

**Overview of C:** Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C, C character Set; C tokens – keywords and identifiers, constants, and variables; Data types; Declaration & initialization of variables.

**Input and output with C:** Formatted I/O functions - printf and scanf, Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and puts functions.

### Unit – 3

**12 Hrs**

**C Operators & Expressions:** Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic expressions; Type conversion.

**Decision Making and Branching:** Introduction, Decision Making with IF Statement, Simple IF Statement, the IF-ELSE Statement, Nesting of IF-ELSE Statements, The ELSE IF Ladder, The Switch statement, The ? : Operator, The goto statement.

Decision Making and Looping: Introduction, The while Statement, The do statement, The for statement, Jumps in LOOPS.

#### **Unit – 4**

**12 Hrs**

**Arrays:** One-dimensional Arrays, Declaration of One-dimensional Arrays, Initialization of One-dimensional Arrays, Example programs- Bubble sort, Linear search, Binary search, Two-dimensional Arrays, Declaration of Two-dimensional Arrays, Initialization of Two-dimensional Arrays, Example programs-Matrix Multiplication, Transpose of a matrix.

**Strings:** Declaring & Initializing string variables; String handling functions - strlen, strcmp, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.

#### **Unit – 5**

**12 Hrs**

**User-defined Functions:** Need for functions, Elements of User-defined Functions, Definition of Functions, Return Values and their Types, Function Calls, Function Declaration, Category of Functions, No Arguments and no Return Values, Arguments but no Return values, Arguments with Return Values, No Arguments but Returns a Value, Passing Arrays to Functions, Recursion, The Scope, Visibility and Lifetime of variables.

**Pointers:** Introduction, Declaring Pointer Variables, Initialization of Pointer variables, accessing a Variable through its Pointer, Pointer Expressions, Pointer Increments and Scale Factor.

**Structures and Unions -** Structure Definition, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.

#### **Text Book:**

1. Fundamentals of Computers, E Balaguruswamy, Mc Graw Hill.
2. E. Balaguruswamy, Programming in ANSI C, 8th Edition, 2019, McGraw Hill Education.
3. C: The Complete Reference, By Herbert Schildt, 4th Edition By Herbert Schildt 2000 | Published: October 6, 2000.

#### **References**

1. C Programming Language, By Brian W. Kernighan, 2<sup>nd</sup> Edition AT & T Bell Laboratories Murray Hill, New Jersey
2. Kernighan B.W and Dennis M. Ritchie, The C Programming Language, 2nd Edition, 2015, Pearson Education India, ISBN: 978-93-3254-944-9.
3. Yashavant P. Kanetkar, “Let Us C”, 16th Edition, 2019, BPB Publications,
4. E- Resources <https://nptel.ac.in/courses/106/105/106105171/>=
5. P. K. Sinha & Priti Sinha: Computer Fundamentals, BPB Publications
6. Kamthane, Programming with ANSI and TURBO C, Pearson Education.

## BSc Computer Science Major I Semester

<b>Course Code: 24BSCSML1.2</b>	<b>Course Title: C Programming Lab.</b>	
<b>Credits: 2</b>	<b>Hrs/Week: 4 Hrs</b>	<b>Contact Hours: 45</b>
<b>Formative assessment: 10 marks</b>	<b>Term end exam: 40 marks</b>	<b>Exam duration: 03 hrs</b>

**Course Outcomes (COs):** At the end of the course, students will be able to:

- Understand basic structure of the C, declaration and usage of variables
- Write C programs using operators
- Write C programs using conditional and iterative statements
- Using pointers and allocate memory using dynamic memory management functions.
- Write C programs using Pointers to access arrays, strings and functions.
- Write C programs using pointers and allocate memory using dynamic memory management functions.
- Write C programs to solve real world problems using functions, structures and unions

### Steps involved in program development:-

To develop the program in high level language and translate it into machine level language following steps have to be practiced.

1. Writing and editing the C program.
2. Linking the program with the required library modules.
3. Compiling the program.
4. Executing the program.

### Following programs shall be written and executed in C

#### PART A

1. Write a C program to perform the Calculator operations, namely, addition, subtraction, multiplication, division and square of a number
2. Write a C Program to read radius of a circle and to find area and circumference of the circle
3. Write a C Program to read three numbers and find the largest of three numbers
4. Write a C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
5. Write a program to compute grade of students using if else ladder. The grades are assigned as follows:

<b>Marks</b>	<b>Grade</b>
marks < 60	F
50 ≤ marks < 60	C
60 ≤ marks < 70	B
70 ≤ marks < 80	B+
80 ≤ marks < 90	A
90 ≤ marks ≤ 100	A+

6. Write a C Program to find the roots of quadratic equation (demonstration of switch-case statement)
7. Write a C program to find GCD (greatest common divisor or HCF) and LCM (least common multiple) of two numbers.
8. Write a C Program to find the length of a string without using built in function
9. Write a C Program to demonstrate string functions.
10. Write a C program to concatenate two strings.

## **PART B**

11. Write a C Program to calculate Standard Deviation.
12. Write a C Program to find the ASCII value of a Character.
13. Write a C Program to find the frequency of a character in a string using function.
14. Write a C Program to read, display and add two  $m \times n$  matrices using functions
15. Write a C Program to read, display and multiply two  $m \times n$  matrices using functions
16. Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
17. Write a C Program to Search an element in array.
18. Write a C Program to Swap Two Numbers using Pointers
19. Write a C program using structures to input name, marks of 5 subjects of a student and display the name of the student, the total marks scored, percentage scored and the class of result.
20. Write a C program to read records of 10 employees and find their average salary (exercise array of structures & Nested structures concepts through this program)

## BSc Computer Science Major- II Semester wef 2024-25

<b>Course Code: 24BSCSMT2.1</b>	<b>Course Title: Data Structures using C.</b>	
<b>Credits: 4</b>	<b>Hrs/Week: 4 Hrs</b>	<b>Contact Hours: 60</b>
<b>Formative assessment: 20 marks</b>	<b>Term end exam: 80 marks</b>	<b>Exam duration: 03 hrs</b>

Course Outcomes (COs): At the end of the course, students will be able to:

1. Understand data structures stacks, queues, trees, and graphs concepts and writing algorithms to represent them in memory, store data in these structures.
2. Understand common applications for arrays, records, linked structures, stacks, queues, and trees
3. Write c programs to solve real world problems using arrays, records, linked structures, stacks, queues, trees, and graphs
4. Understand tree traversal
5. Describe the concept of recursion and writing programs to illustrate the same
6. Understand computational efficiency of the algorithms
7. Ability to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.

### Unit – 1

**12 Hrs**

**Introduction to data structures:** Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures.

Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - *malloc*, *calloc*, *realloc* and *free*.

Algorithm Specification, Performance Analysis, Performance Measurement

Recursion: Definition; Types of recursions; Recursion Technique Examples - GCD,

Binomial coefficient  ${}^n C_r$ , Towers of Hanoi; Comparison between iterative and recursive functions.

### Unit – 2

**12 Hrs**

**Arrays:** Arrays as abstract data types (ADT); Representation of Linear Arrays in memory;

Traversing linear 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.

### Unit – 3

**12 Hrs**

**Linked list:** Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Header linked list, Circular linked list; Representation of Linked list in Memory;

Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection

#### **Unit – 4**

**12 Hrs**

**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 using stack; Application of stack in function calls.

**Queues:** Basic Concepts – Definition and Representation of queues; Types of queues

- Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;

#### **Unit – 5**

**12 Hrs**

**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; preorder, inorder and post order traversal;

#### **Text Book**

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures, ISBN: 0-914894-20X.
2. An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill

#### **References**

3. Data Structures: A Pseudo-code approach with C -By Gilberg & Forouzan Publisher Thomson Learning
4. Kamathane, Introduction to Data structures, Pearson Education
5. Robert Kruse, C.L Tondo and Bruce Leung, “Data Structure and Programming in C”, Pearson Education
6. Yedidyah Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, “Data Structure using C and C++”, Pearson Education 2nd Edition
7. Robert Kruse, C.L Tondo and Bruce Leung, “Data Structure and Programming in C”, Pearson Education
8. Alfred V Aho, John E Hopcroft and Jeffery D Ullman, “ Data Structures and Algorithms”, Pearson Education

## BSc Computer Science Major II Semester

<b>Course Code: 24BSCSML2.2</b>	<b>Course Title: Data Structure Using C Lab.</b>	
<b>Credits: 2</b>	<b>Hrs/Week: 4 Hrs</b>	<b>Contact Hours: 45</b>
<b>Formative assessment: 10 marks</b>	<b>Term end exam: 40 marks</b>	<b>Exam duration: 03 hrs</b>

**Course Outcomes (COs):** At the end of the course, students will be able to:

- Ability to identify the appropriate data structure for given problem
- Ability to write recursive functions
- Write programs to allocate memory dynamically using C memory functions
- To design programs for solving problems using different data structures.
- To evaluate various search and sorting techniques

### PART A

1. Program to find GCD using recursive function
2. Program to generate n Fibonacci series using recursive function.
3. Program to implement Towers of Hanoi.
4. Program to create a file to store student record and read them for display.
5. To insert an element in an array –at beginning, at end and in middle.
6. To concatenate two strings using arrays.
7. Program to sort the given list using selection sort technique.
8. Program to sort the given list using bubble sort technique.
9. Program to sort the given list using insertion sort technique.
10. Program to sort the given list using quick sort technique.

### PART B

11. Program to sort the given list using merge sort technique.
12. Program to search an element using linear search technique.
13. Program to search an element using recursive binary search technique.
14. Program to implement Stack operations using arrays and structure concept..
15. Program to convert an infix expression to postfix.
16. Program to implement queue operations using arrays and structure concept.
17. Write a program that uses functions to perform the following operations on circular linked List i) Creation ii) Insertion iii) Deletion iv) Traversal using array implementation
18. Write a program that uses functions using dynamic memory allocation to perform the following operations on singly linked list i) Creation ii) Insertion iii) Deletion iv) Traversal.
19. Program to illustrate binary tree traversal.
20. Program to implement binary search tree using array.
21. Write a program to perform the following operations: a) Insert an element into a binary search tree. b) Delete an element from a binary search tree. c) Search for a key element in a binary search tree.

**Pedagogy for theory papers:** Lecture/ PPT/ Videos/ Animations/ Role Plays/ Think-Pair-Share/ Predict-ObserveExplain/ Demonstration/ Concept mapping/ Case Studies examples/ Tutorial/ Activity/ Flipped Classroom/ Jigsaw/ Field based Learning/ Project Based Learning/ Mini Projects/ Hobby Projects/ Forum Theatre/ Dance/ Problem Based Learning/ Game Based Learning/ Group Discussion/ Collaborative Learning/ Experiential Learning / Self Directed Learning etc.

<b>Formative Assessment Scheme (Theory) Max marks: 20</b>		
	Internal Test 1	: 05
	Internal Test 2	: 05
	Seminar	: 05
	assignment	: 05
<b>Formative assessment scheme shall be followed as prescribed by the University from time to time</b>		

<b>Formative Assessment Scheme (Lab.) Max marks: 10</b>		
	Internal Lab Test 1	: 05
	Internal Lab Test 2	: 05
<b>Formative assessment scheme shall be followed as prescribed by the University from time to time</b>		

#### **Evaluation Scheme for Lab Examination**

Assessment Criteria		Marks
Program – 1 from Part B	Writing the Program	10
	Execution and Formatting	05
Program -2 from Part B	Writing the Program	10
	Execution and Formatting	05
Viva Voice based on C Programming		05
Practical Record		05
Total		<b>40</b>

## Semester End Examination

Time: 3Hrs.

Max. Marks: 80

Instruction to Candidates: Part A All questions are Compulsory  
Part B Answer any five full questions

### PART A

Q 1

(10\*2=20 M)

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

### PART B

(5\*12=60M)

Q 2	a)	6M
	b)	6M
Q 3	a)	6M
	b)	6M
Q 4	a)	6M
	b)	6M
Q 5	a)	6M
	b)	6M
Q 6	a)	6M
	b)	6M
Q 7	a)	6M
	b)	6M
Q 8	a)	6M
	b)	6M

