

**COURSE MODULE FOR THE SESSION 2024- 2025 (ODD)**

<b>Faculty Name/s :</b>				<b>Academic Year: 2024 – 2025</b>			
<b>Department:</b> Computer Science& Engineering- Data Science							
Course Code	Course Title	Core/Elective	Prerequisite	Contact Hours			Total Hrs/ Sessions
				L	T	P	
BCSL305	Data Structures Laboratory	Core	C Programming Concepts	0	0	2	28

**Course Learning Objectives:**

This laboratory course enables students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Dynamic memory management
- Linear data structures and their applications such as stacks, queues and lists
- Non-Linear data structures and their applications such as trees and graphs

**Topics Covered as per Syllabus**

- Develop a Program in C for the following:
  - Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), The second field is the date of the Day (A integer), the third field is the description of the activity for a particular day (A dynamically allocated String).
  - Write functions create(), read() and display(); to create the calendar, to read the data from the keyboard and to print weeks activity details report on screen.
- Develop a Program in C for the following operations on Strings.
 

Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)

  - Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR. Support the program with functions for each of the above operations. Don't use Built-in fiunctions.
- Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)
  - Push an Element on to Stack
  - Pop an Element from Stack
  - Demonstrate how Stack can be used to check Palindrome
  - Demonstrate Overflow and Underflow situations on Stack
  - Display the status of Stack
  - Exit

Support the program with appropriate functions for each of the above operations
- Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, \*, /, % (Remainder), ^ (Power) and alphanumeric operands.

---

**Department of Computer Science & Engineering (Data- Science)**

---

5. Design, Develop and Implement a Program in C for the following Stack Applications a. Evaluation of Suffix expression with single digit operands and operators: +, -, \*, /, %, ^ b. Solving Tower of Hanoi problem with n disks.
6. Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations.
7. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Programme, Sem, PhNo a. Create a SLL of N Students Data by using front insertion. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion / Deletion at End of SLL d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack) e. Exit
8. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
9. Design, Develop and Implement a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes a. Represent and Evaluate a Polynomial  $P(x,y,z) = 6x^2 y^2 z - 4yz^5 + 3x^3 yz + 2xy^5 z - 2xyz^3$  b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations
10. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers . a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
11. Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method
12. Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a



**Department of Computer Science & Engineering (Data- Science)**

Program in C that uses Hash function  $H: K \rightarrow L$  as  $H(K) = K \bmod m$  (remainder method), and implement hashing technique to map a given key  $K$  to the address space  $L$ . Resolve the collision (if any) using linear probing.

**Course Outcomes**

**After studying this course, students will be able to**

- Analyze and Compare various linear and non-linear data structures
- Code, debug and demonstrate the working nature of different types of data structures and their applications
- Use appropriate searching and sorting algorithms for the give scenario.
- Apply the appropriate data structure for solving real world problems.

**The Correlation of Course Outcomes (CO's) and Program Outcomes (PO's)**

Subject Code:	BCSL305		TITLE: Data Structures and Laboratory				Faculty Name:							
List of Course Outcomes	Program Outcomes												Total	
	PO-1	PO- 2	PO- 3	PO- 4	PO- 5	PO- 6	PO- 7	PO-8	PO- 9	PO- 10	PO- 11	PO- 12		
	CO-1	2	2	-	-	-	-	-	-	-	-	1		5
	CO-2	1	-	2	-	-	-	-	-	-	-	1		4
	CO-3	1	2	2	-	-	-	-	-	-	-	1		6
	CO-4	1	1	3	-	-	-	-	-	-	-	1		6
Total	5	5	7	-	-	-	-	-	-	-	-	4	21	

**Note:** 3 = Strong Contribution 2 = Average Contribution 1 = Weak Contribution 0 = No Contribution

**The Correlation of Course Outcomes (CO's) and Program Outcomes (PO's)**

Subject Code:	BCSL305	TITLE: Data Structures and Laboratory	Faculty Name:		
List of Course Outcomes	Program Outcomes			Total	
	PSO-1	PSO-2			
CO-1	2	0		2	
CO-2	2	0		2	
CO-3	2	0		2	
CO-4	2	0		2	
Total	8	0		8	