DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE & Affiliated to Anna University, Chennai)
Re-Accredited by NAAC with 'A' Grade
Accredited by NBA for AERO, BME, CSE, ECE, EEE, IT & MECH.
PERAMBALUR-621212, TAMILNADU, INDIA.
Website: www.dsengg.ac.in

LABORATORY COURSE PLAN

LAB COURSE TITLE	**DATA STRUCTURES LABORATORY**			
LAB COURSE CODE	**U23ITP32**			
LAB COURSE STRUCTURE	**LECTURE**	**TUTORIAL**	**PRACTICAL**	**CREDIT**
	0	**0**	**4**	**2**
REGULATION	**BRANCH**	**YEAR**	**SEMESTER**	**ACADEMIC YEAR**
2023	**IT-C**	**II**	**III**	**2025-2026**
COURSE INCHARGE				

SYLLABUS

COURSE OBJECTIVE:

The main learning objective of this course is to prepare the students for:

1. To implement Linear and Non – Linear Data Structures
2. To understand the different Operations of Search Trees
3. To implement Graph Traversal Algorithms
4. To get familiarized to Sorting and Searching Algorithms

LIST OF EXPERIMENTS

1. Array implementation of Stack and Queue ADTs
2. Array implementation of List ADT
3. Linked list implementation of List, Stack and Queue ADTs
4. Applications of List, Stack and Queue ADTs

5. Implementation of Binary Trees and operations of Binary Trees
6. Implementation of Binary Search Trees
7. Implementation of AVL Trees
8. Implementation of Heaps using Priority Queues.
9. Graph representation and Traversal algorithms
10. Applications of Graphs
11. Implementation of searching and sorting algorithms
12. Hashing – any two collision techniques

TOTAL: 60 PERIODS

Text Book:

T1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition, Pearson Education, 1997.
T2. Reema Thareja, "Data Structures Using C", Second Edition, Oxford University Press, 2011.

Reference Book:

R1. Thomas H. Cormen, Charles E. Leiserson, Ronald L.Rivest, Clifford Stein, "Introduction to Algorithms", Second Edition, Mcgraw Hill, 2002.
R2. Aho, Hopcroft and Ullman, "Data Structures and Algorithms", Pearson Education,1983.
R3. Stephen G. Kochan, "Programming in C", 3rd edition, Pearson Education
R4. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, "Fundamentals of Data Structures in C", Second Edition, University Press, 2008

Virtual lab reference link

https://ds2-iiith.vlabs.ac.in/List%20of%20experiments.html
https://cse01-iiith.vlabs.ac.in/

EXP. NO.	NAME OF THE EXPERIMENTS	NO. OF PERIODS	CUMULATIVE PERIODS
CYCLE I			
1	Array implementation of Stack and Queue ADTs	4	4

2	Array implementation of List ADT	4	8
3	Linked list implementation of List, Stack and Queue ADTs	4	12
4	Applications of List, Stack and Queue ADTs	8	16
5	Implementation of Binary Trees and operations of Binary Trees	4	20
6	Implementation of Binary Search Trees	8	24
CYCLE II			
7	Implementation of AVL Trees	4	28
8	Implementation of Heaps using Priority Queues.	4	36
9	Graph representation and Traversal algorithms	8	44
10	Applications of Graphs	4	52
11	Implementation of searching and sorting algorithms	4	56
12	Hashing – any two collision techniques	4	60

COURSE OUTCOME

At the end of the course, the student should be able to:

CO 1: Develop and array implement of Stack and Queue ADTs
CO 2: Develop and array implement of List ADT
CO 3: Develop and implement List, Stack and Queue ADTs.
CO 4: Apply the concept of Binary Trees , Binary Search Trees, AVL Trees
CO 5: Develop and implement Heaps using Priority Queues
CO 6: Apply the concept of searching and sorting algorithms

Course Outcome Vs Program Outcome Mapping:

CO-PO mapping: 1 - low, 2 - medium, 3 - high, '-' - no correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	1	3	-	-	-	1	2	1	-	2	1
CO2	2	2	3	2	2	-	-	-	1	3	1	-	1	1
CO3	2	3	2	2	2	-	-	-	1	2	-	-	1	2
CO4	2	2	2	2	3	-	-	-	1	2	-	-	1	2
CO5	3	3	2	2	2	-	-	-	1	2	-	-	1	1
CO6	2	3	3	2	2	-	-	-	1	2	1	-	2	1
AVg.	2.33	2.67	2.5	1.83	2.33	-	-	-	1	2.17	1	-	1.33	1.33

ADDITIONAL EXPERIMENTS		
EXP. NO.	**NAME OF THE EXPERIMENTS**	**Identified Resource link**
1	Matrix Data Structure	https://www.geeksforgeeks.org/matrix/?ref=lbp
2	Bitonic Sort	https://www.javatpoint.com/bitonic-sort
3	Treap Data Structure	https://www.javatpoint.com/treap-data-structure
4	Double Hashing	https://www.geeksforgeeks.org/double-hashing/?ref=lbp
5	Abstraction	https://www.geeksforgeeks.org/abstraction-in-cpp/?ref=ml_lbp

Google Class Code details: ozholpd

MODEL LAB DETAILS

BATCH	REGISTER NO.	MODE OF LAB CONDUCT	DATE	TIMING
1	810423205064-8104232050126	offline	29.12.2024	9.15-1.00

LIST OF EXPERIMENTS

SET-I

1.	a.) Code a C Program to implement the following various operations for **Stack ADT** using **array** implementation. i.) Push ii.) Pop iii.) Display. **(60)** b.) Code a C Program to check whether the given number is in the list using **linear search. (40)**
2.	a.) Code a C Program to implement the following various operations for **Queue ADT** using **array** implementation. i.) Insert ii.) Delete iii.) Display**.** **(60)** b.) Code a C Program to check whether the given number is in the list using **binary search**. **(40)**
3.	a.) Write a C Program to implement the following various operations for **List ADT** using **array** implementation. i.) Create ii.) Insert iii.) Erase iv.) Traverse **(60)** b.) Write a C Program to sort the following numbers using **Insertion sort:** 67, 53, 82, 7,11,3, 101 **(40)**

4.	a.) Write a C Program to implement the following various operations for **List ADT** using **array** implementation. i.) Create ii.) Traverse iii.) Search. **(60)** b.) Write a C Program to sort the following numbers using **Bubble sort:** 57, 83, 62, 5,17,9, 111 **(40)**
5.	Code a C program to implement **Stack ADT** by using **linked list** for the following operations i.) Push b.) Pop c.) Display. **(60)** b.) Write a C Program to sort the following numbers using **Selection sort:** 241,37,78, 16, 15,7 **(40)**
6.	a.) Code a C program to implement **Queue ADT** by using **linked list** for the following operations i.) Enqueue ii.) Dequeue iii.) Display **(60)** b.) Write a C Program to sort the following numbers using **Radix sort:** 41,37,6, 15,23, 99 **(40)**
7.	a.) Using C program to implement **List ADT** by using **linked list** for the following operations i.) Create ii.) Insert iii.) Traverse iv.) Search. **(60)** b.) Write a C Program to sort the following numbers using **Shell sort:** 14, 23, 8, 19, 63, 153 **(40)**
8.	a.) Code a C program to implement **List ADT** by using **linked list** for the following operations i.) Create ii.) Traverse iii.) Find. iv.) Delete **(60)** b.) Code a C program to implement the **hashing** with hash table of size 10 using **separate Chaining** for the following numbers 17,22,11,12,45,27,33,44 **(4**
9.	a.) Write a C program to convert an **infix** expression to **postfix** expression using **Stack ADT. (60)** b.) Code a C program to implement the **hashing** with hash table of size 10 using **open addressing** for the following numbers 8,12,71,98,36, 105, 48 **(40)**
10.	a.) Write a C program to implement the addition of two **polynomials** using linked list.

	(60) b.) Code a C Program to check whether the given number is in the list using **linear search.** **(40)**
11.	a.) Write a C program to construct a **binary tree** and display the nodes using inorder, preorder and postorder traversals. **(60)** b.) Code a C Program to check whether the given number is in the list using **binary search**. **(40)**
12.	a.) Create a node for **Binary Tree** and implement the following operations in the tree Create, Search and Preorder traversal and Postorder traversal using C program. **(60)** b.) Write a C Program to sort the following numbers using **Insertion sort**: 16, 43, 92, 9, 71, 88, 7,3 **(40)**
13.	a.) Write a C Program for **Binary Search Tree** with the following operations Create, Insert, Delete and View. **(60)** b.) Write a C Program to sort the following numbers using **Bubble sort:** 117, 83, 4, 3,17,9, 29 **(40)**
14.	a.) Write a C Program for **Binary Search Tree** with the following operations Create, Insert, Findmin and Findmax. **(60)** b.) Write a C Program to sort the following numbers using **Selection sort:** 24,17,48, 6, 19,8 **(40)**
15.	a.) Create a node and perform the following operations Insertion with single rotation and double rotation, Display in **AVL Tree** using C program. **(60)** b.) Write a C Program to sort the following numbers using **Radix sort:** 45,73, 86, 9,93, 19 **(40)**
16.	a.) Code a C program for the implementation of priority queue using the following **heap** operations using **Minheap**. i.) Insert ii.) Delete **(60)** b.) Write a C

	Program to sort the following numbers using Shell sort: 6, 23, 18, 11, 69, 15 **(40)**
17.	a.) Code a C program for the implementation of priority queue using the following heap operations using **Maxheap** i.) Insert ii.) Delete **(60)** b.) Code a C program to implement the hashing with hash table of size 10 using **separate Chaining** for the following numbers 7,12,41,61,75,2,5,8 **(40)**
18.	a.) Implement the **DFS** to perform depth first search on the graph using C. **(60)** b.) Code a C program to implement the hashing with hash table of size 10 using **open addressing** for the following numbers 11,32,63,18,46,5,123 **(40)**
19.	a.) Implement the **BFS** to perform breadth first search on the graph using C. **(60)** b.) Code a C Program to check whether the given number is in the list using **linear search. (40)**

SET-II

1.	Execute the push and pop operations performed on a stack of size 5 using C program.
2.	Implement First In First Out order of queue ADT using arrays in C.
3.	Write a C program to implement a sequential data structure - list ADT using array.
4.	Write a C program to implement stack ADT using linked list implementation
5.	Write a C program to implement queue ADT using linked list implementation
6.	Insert elements in a node with pointers and delete few elements using list ADT in C program.
7.	How will you apply stack in converting Infix to postfix using C? Write a program to implement the same.
8.	Write a C program to implement insertion and deletion on both sides of a list / application of queue ADT – DEQueue
9.	Extract the coefficient of a given exponent and write a program to perform addition using lists in C.
10.	Write a C program to implement an application of list ADT - Polynomial subtraction.
11.	Explain programmatically using C, how to perform the different traversals in a binary tree.

12.	How to implement the search operation on a binary search tree, after performing insertion operation using C.
13.	Implement a C program with the criteria, every descendant node's value in the left subtree of n is less than the value of n and every descendant node's value in the right subtree is greater than the value n.
14.	Write a C program to implement a self balancing tree with nodal height difference of 1 unit.
15.	To insert 1 to 10 numbers in a tree with a balancing factor of 1 maximum, frame a code using C.
16.	Implement MIN Heap property using C.
17.	Substantiate the importance of MAX Heap in insertion and deletion by programming using C
18.	A person wants to visit some places. He starts from a vertex and then wants to visit every place connected to this vertex and so on. What traversal methodology suits best for him and implement it using C? (Breadth First Search).
19.	A person wants to visit some places. He starts from a vertex and then wants to visit every vertex till it finishes from one vertex, backtracks and then explore other vertex from same vertex. What traversal methodology suits best for him and implement it using C? (Depth First Search)
20.	Describe programmatically using C, how to find the shortest path as an application of graphs.
21.	How to locate an element in an array using a) linear search (5+20+10) b) binary search (10+20+15)
22.	Write a simple C program to narrate the execution of sorting using insertion sort / selection sort / bubble sort / shell sort
23.	Try coding the hashing function using linear probing in C language.

VIVA QUESTIONS

1) What is data structure?
2) Differentiate between file and structure storage structure.
3) When is a binary search best applied?
4) What is a linked list?
5) How do you reference all the elements in a one-dimension array?

6) In what areas do data structures are applied?

7) What is LIFO?

8) What is a queue?

9) What are binary trees?

10) Which data structures are applied when dealing with a recursive function?

11) What is a stack?

12) Explain Binary Search Tree

13) What are multidimensional arrays?

14) Are linked lists considered linear or non-linear data structures?

15) How does dynamic memory allocation help in managing data?

16) What is FIFO?

17) What is an ordered list?

18) What is merge sort?

19) Differentiate NULL and VOID

20) What is the primary advantage of a linked list?

21) What is the difference between a PUSH and a POP?

22) What is a linear search?

23) How does variable declaration affect memory allocation?

24) What is the advantage of the heap over a stack?

25) What is a postfix expression?

26) What is Data abstraction?

27) How do you insert a new item in a binary search tree?

28) How does a selection sort work for an array?

29) How do signed and unsigned numbers affect memory?

30) What is the minimum number of nodes that a binary tree can have?

PREPARED BY **VERIFIED BY**

AP/IT **HOD/IT**

APPROVED BY

PRINCIPAL