NAWAB SHAH ALAM KHAN COLEEG OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY IMPROVEMENT FOR FUTURE TRENDS (LIFT) - - -I

NAME OF THE LABORATORIES

S.No.	YEAR-SEM	NAME OF THE LAB
1	II B.TECH-ISEM	DATA STRUCTURES

A Guide for execution of Lab Courses

VISION OF THE INSTITUTE:

To be a leading institute of world class quality technical education with strong ethical values, preparing students for leadership in their fields for the dynamic and global careers, developing breakthrough environment for professional education and research.

MISSION OF THE INSTITUTE:

- M1: To provide adequate knowledge encompassing strong technical concepts and soft skills thereby inculcating sound ethics.
- M2: To provide a conducive environment to nurture creativity in teaching- learning process.
- M3: To identify and provide facilities which create opportunities for deserving students of all communities to excel in their chosen fields.
- M4: To strive and contribute to the needs of the society and the nation by applying advanced engineering and technical concepts

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION:

To produce quality IT professionals, with an ability to adapt to ever changing IT needs of local, national and international arena, through effective teaching & learning, interactions with alumni and industry

MISSION:

- M1: To provide a holistic learning environment for students through ethical practices.
- M2: To provide quality infrastructure through practical exposure to the latest technology requirements.
- M3: To train the students in soft skills to excel in placements and competitive exams at higher level the industry ready.
- M4: To have a healthy Industry - Institute interaction through faculty development programs, student internships, guest lectures and using latest teaching learning methodologies.
- M5: To provide effective platform to meet the industrial requirement and provide research-oriented environment for the faculty to meet the continuous societal needs.

PROGRAM SPECIFIC OUTCOMES (PSO's)

- Develop efficient information management systems using latest development tools catering to the globally changing requirements in multi-disciplinary domains
- Manage real time IT projects with consideration of human, financial, ethical and environmental factors and an understanding of policy implications.

1. **AIM OF THE LIFT**:

The main aim of the LIFT programme is to innovate, modify the existing facilities in labs, to create awareness among the students and develop Industry –Institution interactions and reach the standards in laboratories

2. **FUNCTIONS OF THE LIFT**:

I. To create better understanding concepts of LIFT and other lab related activities among the staff and lab technicians for better improvement.

II. To Arrange LIFT Presentations from each department about the lab activities by the staff handling the labs. (Lab Planners)

III. To Prepare GAP ANALYSIS: This involves collection of requirements from each lab of every department, information about expansion of labs, repairs and maintenance of labs etc.

IV. To arrange Industrial Visits/ Industrial training programs in coordination with concerned lab staff and Heads of the departments.

V. A Report on Shadow Engineering: This involves arrangement of Industrial and Practical learning, Submission of Industrial Visit report, Technical Survey reports and Market Survey of a product for development in laboratories.

VI. Verification of all the laboratories in every department by the LIFT Team along with the Principal and the concerned HODs, to check whether the activities are going according to LIFT guidelines, to check the Record Keeping, Lab Manuals and Viva sessions etc.

VII. Check for LEAD Experiments and its follow up.

VIII. Submission of proposals related to R&D, Project and Consultancy from lab staff to the Principal for further approvals.

LAB IMPROVEMENT FOR FUTURE TRENDS PROGRAMME (LIFT)

INDEX

CONTENTS:

1. OBJECTIVES AND RELEVANCE:

The main objective of the LIFT concept in lab course is to provide practical hands on experience for each student by providing them with good exposure to different experiments and to uplift the knowledge levels of the student, with different applications in various fields.

2. SCOPE:

The main scope of the LIFT lab course is to cover all the experiments as per the schedule given in the prescribed week wise periods. With this, a student can better understand the concepts and operating systems so that he could get better knowledge about each lab.

3. PREREQUISITES:

The basic level idea related to each experiment should be provided to the students before conducting main lab course. Following details are to be explained related to experiment:

1. Introduction to experiment – 30 min
2. The Operating of the equipment/instrument/software
3. Record of Experimental Results.
4. Sample Calculations / Executable Programs

4. SYLLABUS AS PER JNTUH:

The lab course should be planned as per the JNTUH syllabus. In this, LEAD experiments should also be included in the cycle of experiments.

5. (A) LAB SCHEDULE:

The lab schedule should be planned once in a week. The week wise scheduled experiment should be completed.

Batches	week-1	week-2	week-3	week-4	week-5	week-6
B1	Demo	Exp.1	Exp.2	Exp.10	Exp.9	Exp.8
B2	Demo	Exp.2	Exp.10	Exp.9	Exp.8	Exp.1
B3	Demo	Exp.10	Exp.9	Exp.8	Exp.1	Exp.2
B4	Demo	Exp.9	Exp.8	Exp.1	Exp.2	Exp.10
B5	Demo	Exp.8	Exp.1	Exp.2	Exp.10	Exp.9

(B) Scheme of Evaluation:

The scheme of evaluation for internal and external exams as follows:

LAB INTERNAL:							
Day to Day Evalution-15					**Internal Exam-10**		
Uniform	**Observation & Record**	**Performance Of the Experiment**	**Result**	**Viva**	**Write up**	**Execution & Results**	**Viva**
Marks-3	**Marks-3**	**Marks-3**	**Marks-3**	**Marks-3**	**Marks-4**	**Marks-3**	**Marks-3**
Total Marks-25							

LAB EXTERNAL:			
S.NO	Write up	Final Evaluation	Viva
1	**1. Aim** **2. Procedure** **3. Program** **4. Expected output.**	**Based on correctness of the program and Results**	**Based on understanding of Experiment and theoretical questions in the related subjects**
Marks	**20**	**20**	**10**
Total Marks-50			

6. SUGGESTED BOOKS:

The suggested books should be recommended to the students as per the JNTUH syllabus prescribed.

7. WEBSITES (USEFUL LINKS):

The useful links should be provided to the students, where they can get an easy access to the knowledge of the experiment.

SUBJECTWISE LAB PLANNER

Name of the Subject: DATA STRUCTURES LAB

CONTENTS:

1. OBJECTIVES AND RELEVANCE

To develop skills to design and analyze simple linear and non linear data structures, to strengthen the ability to identify and apply the suitable data structure for the given real world problems and to gain knowledge in practical applications of data structures.

2. SCOPE

Assess how the choice of data structure design methods impacts the performance of programs. Choose the appropriate data structure design method for a specified application. Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps0, binary search trees, and graphs and writing programs for these solutions.

3. PREREQUISITES

This subject recommends continuous practice of powerful modern language that combines the power, elegance and flexibility of C programming.

4. JNTUH Syllabus

Week1: Write a C program that uses functions to perform the following:

a) Create a singly linked list of integers.

b) Delete a given integer from the above linked list.

c) Display the contents of the above list after deletion.

Week2: Write a C program that uses functions to perform the following:

a) Create a doubly linked list of integers.

b) Delete a given integer from the above doubly linked list.

c) Display the contents of the above list after deletion.

Week3: Write a C program that uses stack operations to convert a given infix expression into its postfix equivalent. Implement the stack using an array.

Week 4: Write C programs to implement a double ended queue ADT using

i) array and ii) doubly linked list respectively.

Week 5: Write a C program that uses functions to perform the following:

a) Create a binary search tree of characters.

b) Traverse the above Binary search tree recursively in Post order.

Week 6: Write a C program that uses functions to perform the following:

a) Create a binary search tree of integers.

b) Traverse the above Binary search tree non recursively in order.

Week 7: Write C programs for implementing the following sorting methods to arrange a list of integers in Ascending order:

a) Insertion sort b) Merge sort

Week 8: Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order:

a) Quick sort b) Selection sort

Week 9: i) Write a C program to perform the following operation:

a) Insertion into a B-tree.

ii) Write a C program for implementing Heap sort algorithm for sorting a given list of integers in ascending order.

Week 10: Write a C program to implement all the functions of a dictionary (ADT) using hashing.

Week 11: Write a C program for implementing Knuth-Morris- Pratt pattern matching algorithm.

Week 12: Write C programs for implementing the following graph traversal algorithms:

a) Depth first traversal b) Breadth first traversal

<u>LEAD PROGRAMS</u>

1. Write a C program to perform operations of circular queue.
2. Write a C program to check whether the given matrix is sparse matrix or not.

MAIN LINKAGE OF DATA STRUCTURES THEORY WITH LAB EXPERIMENTS:

UNIT-I

EXPERIMENT NO.1

Write a C program that uses functions to perform the following:

a) Create a singly linked list of integers.

b) Delete a given integer from the above linked list.

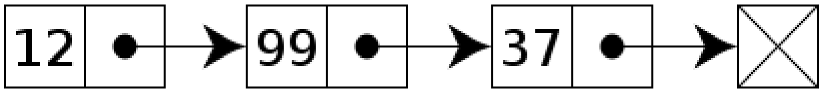
c) Display the contents of the above list after deletion.

. **OBJECTIVE:** The main objective is to understand singly linked lists and their operations

PRE REQUISITES: Basic understanding of nodes and pointers

DESCRIPTION:

Singly Linked Lists are a type of data structure. It is a type of list. In a singly linked list each node in the list stores the contents of the node and a pointer or the reference to the next node in the list. It does not store any pointer or reference to the previous node. It is called a singly linked list because each node only has a single link to another node. To store a single linked list, you only need to store a reference or pointer to the first node in that list. The last node has a pointer to nothingness to indicate that it is the last node.



A singly linked list whose nodes contain two fields: an integer value and a link to the next node

APPLICATIONS:

1) A singly-linked list is ideally suited to stacks, queues

2) Reduces access time and may expand in real time without memory overhead

EXPERIMENT NO.2

Write a C program that uses functions to perform the following:

a) Create a doubly linked list of integers.

b) Delete a given integer from the above doubly linked list.

c) Display the contents of the above list after deletion.

OBJECTIVE:

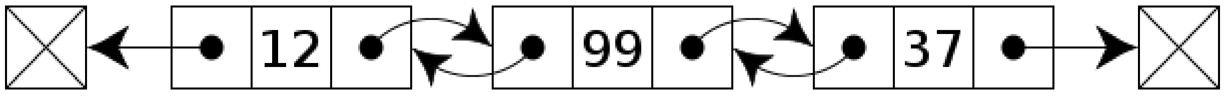
The main objective is to understand doubly linked list and their operations

PRE REQUISITES:

Basic understanding of singly linked lists and their operations

DESCRIPTION:

In a **doubly linked list**, each node contains, besides the next-node link, a second link field pointing to the *previous* node in the sequence. The two links may be called **forward(s)** and **backwards**, or **next** and **prev(previous)**.



A doubly linked list whose nodes contain three fields: an integer value, the link forward to the next node, and the link backward to the previous node

A technique known as XOR-linking allows a doubly linked list to be implemented using a single link field in each node.

APPLICATIONS:

1) The cache in your browser allows you to hit the BACK button (a linked list of URLs)
2) Undo functionality in Photoshop or Word (a linked list of state)
3) Allows you to hit the BACK button (a linked list of URLs)
4) A great way to represent a deck of cards in a game.

UNIT-II

EXPERIMENT NOs. 3 & 4

3) Write a C program that uses stack operations to convert a given infix expression into its postfix equivalent. Implement the stack using an array.

4) Write C programs to implement a double ended queue ADT using

i) array and ii)doubly linked list respectively.

OBJECTIVE:

The main objective is to understand stack and its operations, de queue and its operations

PRE REQUISITES:

Basics of arrays and linked list concepts

Basic introduction of ADT

Knowledge of different types of expressions

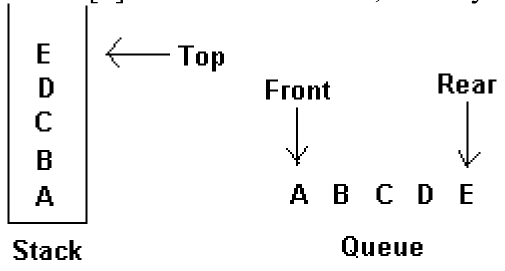
DESCRIPTION:

A *stack* is an ordered list in which all insertions and deletions are made at one end, called the *top*.

A *queue* is an ordered list in which all insertions take place at one end, the *rear*, while all deletions take place at the other end, the *front*.

Given a stack S=(a[1],a[2],. ..... a[n]) then we say that a1 is the bottommost element and element a[i]) is on top of element a[i-1], 1<i<=n.

When viewed as a queue with a[n] as the rear element, one says that a[i+1] is behind a[i], 1<i<=n.



STACK APPLICATIONS:

1) The simplest application of a stack is to reverse a word

2) Another application is an "undo" mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.

2 Compiler’s syntax check for matching braces is implemented by using stack.

Arithmetic expression evaluation

QUEUE APPLICATIONS:

1) Used by operating system in scheduling jobs and spooling (queue of print jobs)

2) Disk I/O calls

UNIT III

EXPERIMENT NO. 12

. Write C programs for implementing the following graph traversal algorithms:

a) Depth first traversal b) Breadth first traversal

OBJECTIVE: The main objective is to understand graph traversal algorithms.

PRE REQUISITES: Basic understanding of nodes, vertices and graphs.

DESCRIPTION:

Breadth-first traversal

In graph theory, breadth-first traversal is a strategy for searching in a graph when search is limited to essentially two operations: (a) visit and inspect a node of a graph; (b) gain access to visit the nodes that neighbor the currently visited node. The BFS begins at a root node and inspects all the neighboring nodes. Then for each of those neighbor nodes in turn, it inspects their neighbor nodes which were unvisited, and so on.

Depth first traversal

Depth-first search (DFS) is a general technique for traversing a graph

A DFS traversal of a graph G

Visits all the vertices and edges of G

Determines whether G is connected

Computes the connected components of G

Computes a spanning forest of G

APPLICATIONS

1) Finds the shortest path between two nodes *u* and *v*

2) To solve complex routing problems such as routing airlines among the airports

3) To route messages over computer networks

UNIT IV

EXPERIMENT NOs. 7, 8 & 9

7) Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order

a) Insertion sort b) Merge sort

8) Write C programs for implementing the following sorting methods to arrange a list of integers in ascending order
a) Quick sort b) Selection sort

9) ii) Write a C program for implementing Heap sort algorithm for sorting a given list of integers in ascending order.

OBJECTIVE:

To arrange the list of integers in an ascending order by using various sorting techniques.

PREREQUISITES:

It requires the knowledge of arrays

DESCRIPTION:

Sorting is the process of placing elements from a collection in some kind of order. Sorting makes searching process easy. Various sorting techniques are available like Insertion sort, Merge sort, Quick sort, Selection sort, Heap sort, etc. to arrange elements in particular order.

APPLICATIONS:
1) A list of words could be sorted alphabetically or by length
2) A list of cities could be sorted by population, by area, or by zip code

EXPERIMENT NO. 10

Write a C program to implement all the functions of a dictionary (ADT) using hashing.

OBJECTIVE:

The main objective is to perform the main operations of a dictionary using hashing.

PREREQUISITES:

Student has to know about ADT and hashing technique.

DESCRIPTION:

The dictionary ADT models a searchable collection of key-element entries. The main operations of a dictionary are searching, inserting, and deleting items. Multiple items with the same key are allowed.

APPLICATIONS:

1) word-definition pairs
2) credit card authorizations
3) DNS mapping of host names

<u>UNIT-V</u>

EXPERIMENT NOs. 5, 6 & 9

5) Write a C program that uses functions to perform the following:

a) Create a binary search tree of characters.

b) Traverse the above Binary search tree recursively in Post order.

6) Write a C program that uses functions to perform the following:

a) Create a binary search tree of integers.

b) Traverse the above Binary search tree non recursively in order.

9) i) Write a C program to perform Insertion into a B-tree.

OBJECTIVE:

To represent and access the elements in a binary search tree, B-tree

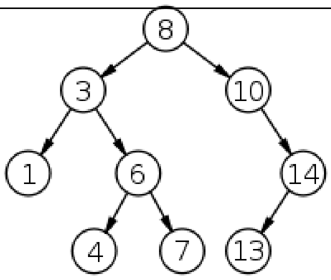
PREREQUISITES:

Basic knowledge of tree data structure and its traversal operations.

DESCRIPTION:

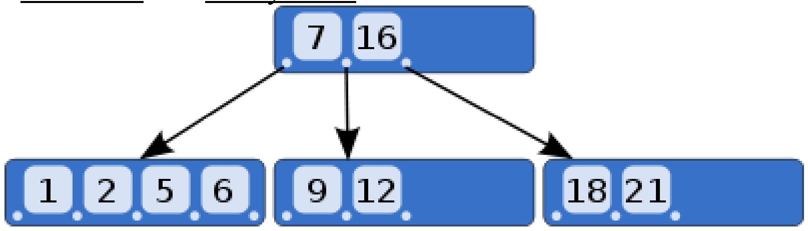
Binary search tree (BST)

A special type of binary tree called a binary search tree (BST) is often useful. A BST (with no duplicate elements) has the property that, for every node in the tree, the value of any node in its left sub tree is less than the value of the node, and the value of any node in its right sub tree is greater than the value of the node.



A binary search tree of size 9 and depth 3, with root 8 and leaves 1, 4, 7 and 13

B-tree
A B-tree is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree is a generalization of a binary search tree in that a node can have more than two children (Comer 1979, p. 123). Unlike self-balancing binary search trees, the B-tree is optimized for systems that read and write large blocks of data. It is commonly used in databases and file systems.



A B-tree of order 2

APPLICATIONS:

1. Used in *many* search applications where data is constantly entering/leaving,
2. To maintain ordered collection of data

EXPERIMENT NO: 11 Write a C program for implementing Knuth-Morris- Pratt pattern matching algorithm.

OBJECTIVE:
To search for occurrences of a "word" W within a main "text string" S

PREREQUISITES:
Basic knowledge of searching algorithms

DESCRIPTION:
In computer science, the **Knuth–Morris–Pratt string searching algorithm** (or **KMP algorithm**) searches for occurrences of a "word" W within a main "text string" S by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters.

APPLICATIONS:
Searches for occurrences of a "word" W within a main "text string" S

LEAD:

1. Write a program to perform operations of circular queue.
2. Write a program to check the given matrix is sparse matrix or not using C.

5(A) LAB SCHEDULE:

CYCLE 1:

Batches	week-1	week-2	week-3	week-4	week-5	week-6	week-7	week-8
B1(501 to 512)	Demo	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	test	Exp.6
B2(513 to 524)	Demo	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	test	Exp.6
B3(525 to 536)	Demo	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	test	Exp.6
B4(537 to 548)	Demo	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	test	Exp.6
B5(549 to 560)	Demo	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	test	Exp.6

(B) VIVA SCHEDULE: The viva schedule should be planned prior to the lab experiment.

CYCLE 1: **ROUND -** 1

Batches	week-1	week-2	week-3	week-4	week-5	week-6	week-7	week-8	week-9	week-10	week-11	week-12	week-13	week-14
B1,B2,B3	viva					viva					Viva			
B4,B5,B1		viva					viva					viva		
B2,B3,B4			viva					viva					viva	
B5,B1,B2				Viva					Viva					viva
B2,B3,B4					viva					viva				

(C) Scheme of Evaluation:

The scheme of evaluation for internal and external exams is as follows:

LAB INTERNAL:							
Day to Day Evalution-15					**Internal Exam-10**		
Uniform	**Observation & Record**	**Performance Of the Experiment**	**Result**	**Viva**	**Write up**	**Execution & Results**	**Viva**
Marks-3	**Marks-3**	**Marks-3**	**Marks-3**	**Marks-3**	**Marks-4**	**Marks-3**	**Marks-3**
Total Marks-25							

6. SUGGESTED BOOKS:

1. C and Data Structures, Third Edition, P.Padmanabham, BS Publications.
2. C and Data Structures, Prof. P.S.Deshpande and Prof. O.G. Kakde, Dreamtech Press.
3. Data structures using C, A.K.Sharma, 2nd edition, Pearson.
4. Data Structures using C, R.Thareja, Oxford University Press.
5. C and Data Structures, N.B.Venkateswarlu and E.V.Prasad, S.Chand.
6. C Programming and Data Structures, P.Radha Krishna, Hi-Tech Publishers.

LAB EXTERNAL:			
S.NO	Write up	Final Evaluation	Viva
1	**1. Aim 2. Procedure 3. Program 4. Expected output.**	**Based on correctness of the program and Results**	**Based on understanding of Experiment and theoretical questions in the related subjects**
Marks	**20**	**20**	**10**
Total Marks-50			

7. WEBSITES

http://www.c4learn.com/data-structure

http://www-gs.informatik.tu-cottbus.de/cs1_v06.pdf

http://www.users.pjwstk.edu.pl/~msyd/wyka-eng/complexity2.pdf

http://www.cs.cmu.edu/~adamchik/15 -
121/lectures/Stacks%20and%20Queues/Stacks%20and%20Queues.html

http://www.cprogramming.com/

ftp://ftp.awl.com/cseng/authors/gray/Graphs%20Chapter/Gray_CH13_rev.pdf

http://www.sanfoundry.com/c-programming-examples-searching-sorting/

https://www.cs.auckland.ac.nz/software/AlgAnim/AVL.html