**Department of Computer Science**

**Lecture Outline**

**Data Structures – 4th CSE**

**Lecture : Linked Lists**

|  |
| --- |
| **Class Lectures on Linked Lists:****Lecture 1 : Introduction to Linked Lists :**[**https://youtu.be/u3H20LlaL2Y**](https://youtu.be/u3H20LlaL2Y)**Lecture 2 : Creation of a Linked List Node:**[**https://youtu.be/YqXohOxBr6M**](https://youtu.be/YqXohOxBr6M)**Lecture 3: Add at Beginning and End of Linked List:** [**https://youtu.be/XDb2HXAriG4**](https://youtu.be/XDb2HXAriG4)**Lecture 4: Delete at Beginning and End of a Linked List:** [**https://www.youtube.com/watch?v=GYzoLbaggdA**](https://www.youtube.com/watch?v=GYzoLbaggdA)**Lecture 5: Add/Delete at a Particular Position in a Linked List:** [**https://www.youtube.com/watch?v=GYzoLbaggdA**](https://www.youtube.com/watch?v=GYzoLbaggdA)**Lecture 6: Print, Count and Search in a Linked List:** [**https://youtu.be/n8eA5-Gw5No**](https://youtu.be/n8eA5-Gw5No) |

**All Programming to be done in C language.**

|  |  |
| --- | --- |
| 1. Linked Lists
 | * 1. What is a list?
	2. Types of Linked Lists ( Singly, Doubly, Circular Linked List) --Read(<https://www.studytonight.com/data-structures/introduction-to-linked-list>)
	3. Advantages of Linked List over Dynamic Arrays
	4. Comparison of Linked List vs Arrays

Read (<https://www.geeksforgeeks.org/linked-list-vs-array/>)* 1. Defining the self referential structure for a Singly Linked List
	2. Creating a single node for a linked list
	3. Typecasting the allocated data
	4. Watch (<https://www.youtube.com/watch?v=K7VIKlUdo20>)
	5. Implementing Different Functions for Linked Lists
		1. Create
		2. Add Node
		3. Delete Node
		4. Print List
		5. Count number of elements in the list (Size of List)
		6. Add Node at particular position
		7. Delete Node at particular position
		8. Insert/Delete node at beginning
		9. Insert/Delete Node at the end
		10. Sort elements of Linked List
		11. Search Linked List for a particular element
		12. Check if List is Empty
		13. Find Nth node of a Linked List
		14. Find some examples here (<https://www.edureka.co/blog/linked-list-in-c/>)
		15. Additional Resource ( <https://www.learn-c.org/en/Linked_lists>)
	6. Learning Implementation of Linked List by:
		1. Declaring start pointer to Linked List as a global variable
		2. Passing start pointer of Linked List to every function either by value or by reference
 |
| 1. Doubly Linked List
 | * 1. Defining the self referential structure for a Doubly Linked List
	2. Replicating the functionb56s created for Singly Linked List for Doubly Linked List
 |
| 1. Circularly Linked List
 | * 1. Defining the self referential structure for a Circular Linked List
	2. Replicating the functions created for Singly Linked List for Circular Linked List
	3. Creating Doubly Circular Linked Lists as an exercise
 |
| 1. Applications of Linked Lists
 | * 1. **Implementation of stacks and queues**
	2. **Implementation of graphs : Adjacency list representation of graphs**
	3. Dynamic memory allocation where linked list of free blocks is used.
	4. Maintaining directory of names
	5. Performing arithmetic operations on long integers
	6. **\*Manipulation of polynomials by storing constants in the node of linked list. Implementing polynomial arithmetic for Two polynomials ( Add/Subtract/Multiply in code using Linked Lists.**

**(**<https://www.daniweb.com/programming/software-development/code/216987/polynomial-addition-using-linked-list> )**Topics a,b to be discussed later in detail.** |