

NAME: SAYAN GHOSH

STREAM: BCA(2ND YEAR)

SECTION: GAMMA; SEM: 3RD

PAPER NAME: DATA STRUCTURE AND ALGORITHM

PAPER CODE: BCAC 303

UNIVERSITY ROLL NO: 15201222165

TOPIC: “LINKED LIST”

PRES^ENTATION ON

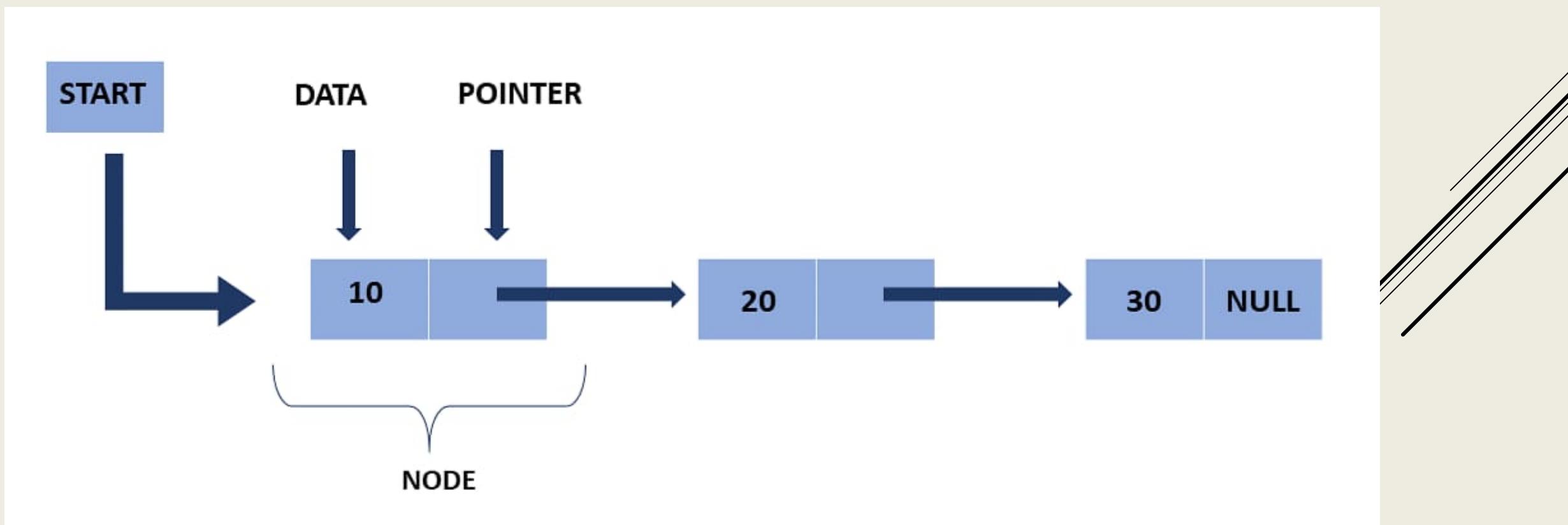
“LINKED LIST”

- **INTRODUCTION:** LINKED LIST IS THE MOST SOUGHT-AFTER DATA STRUCTURE WHEN IT COMES TO HANDLING DYNAMIC DATA ELEMENTS. A LINKED LIST CONSISTS OF A DATA ELEMENT KNOWN AS A NODE AND EACH NODE CONSISTS OF TWO FIELDS
- ONE FIELD HAS DATA AND IN THE SECOND FIELD ,THE NODE ,THE NODE HAS AN ADDRESS THAT KEEPS A REFERENCE TO THE NEXT NODE.

WHAT IS A LINKED LIST?....

- A LINKED LIST IS A LINEAR DATA STRUCTURE THAT STORES A COLLECTION OF DATA ELEMENTS DYNAMICALLY.
- NODES REPRESENT THOSE DATA ELEMENTS, AND LINKS OR POINTERS CONNECT EACH NODE.
- EACH NODE CONSISTS OF TWO FIELDS, THE INFORMATION STORED IN A LINKED LIST AND A POINTER THAT STORES THE ADDRESS OF ITS NEXT NODE.
- THE LAST NODE CONTAINS NULL IN ITS SECOND FIELD BECAUSE IT WILL POINT TO NO NODE.
- A LINKED LIST CAN GROW AND SHRINK ITS SIZE, AS PER THE REQUIREMENT.
- IT DOES NOT WASTE MEMORY SPACE.

REPRESENTATION OF A LINKED LIST: THIS REPRESENTATION OF A LINKED LIST DEPICTS THAT EACH NODE CONSISTS OF TWO FIELDS . THE FIRST FIELD CONSISTS OF DATA AND THE SECOND FIELD CONSISTS OF POINTERS THAT POINT TO ANOTHER NODE.



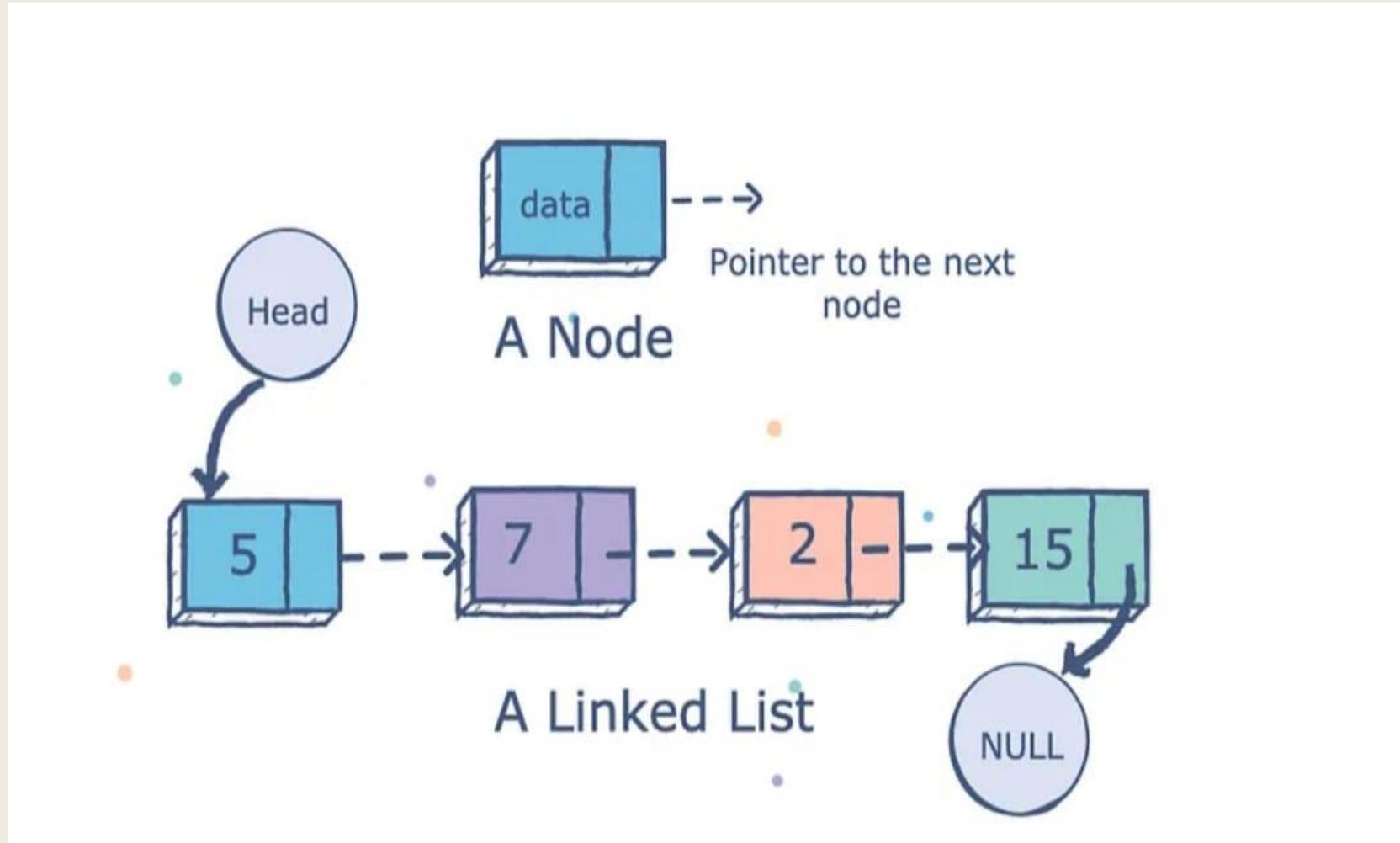
TYPES OF LINKED LIST:

**THE LINKED LIST MAINLY HAS
THREE TYPES, THEY ARE:**

- 1. SINGLY LINKED LIST**
- 2. DOUBLY LINKED LIST**
- 3. CIRCULAR LINKED LIST**



STRUCTURE OF LINKED LIST:



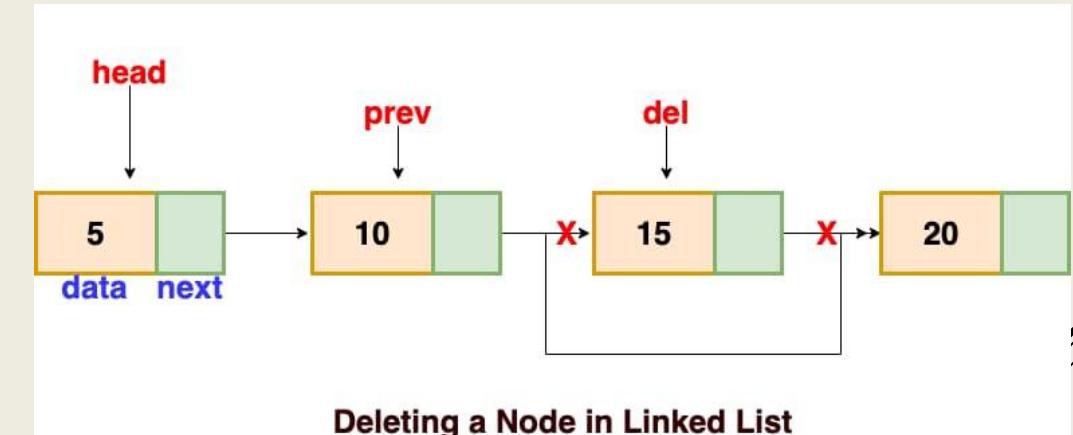
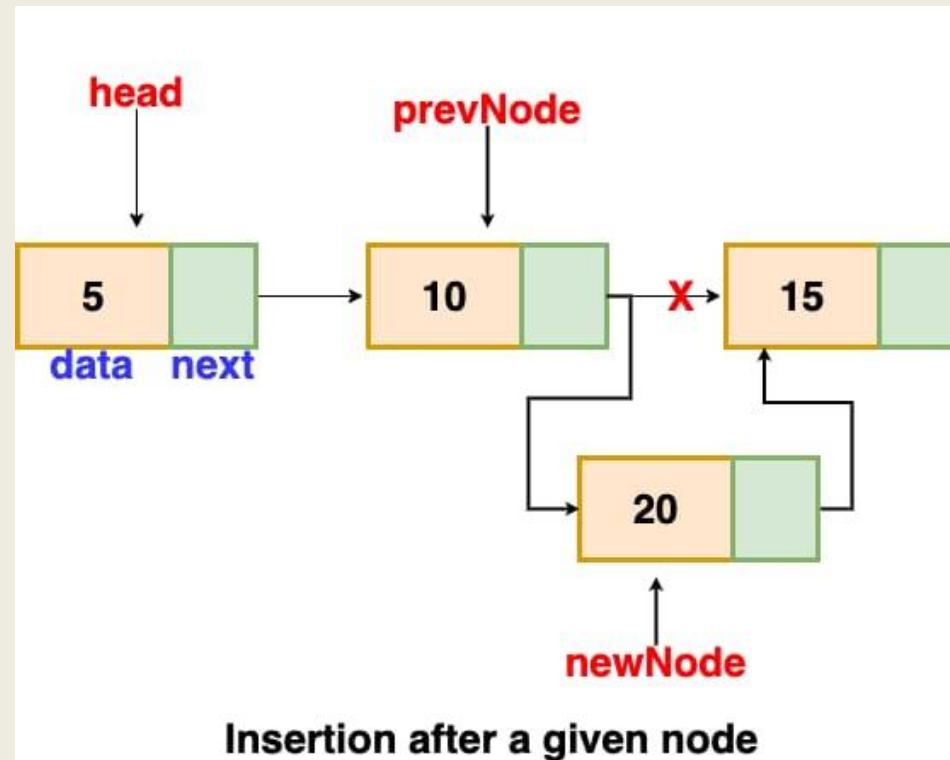
ARRAY VS LINKED LIST:

Arrays	Linked list
Fixed size: Resizing is expensive	Dynamic size
Insertions and Deletions are inefficient: Elements are usually shifted	Insertions and Deletions are efficient: No shifting
Random access i.e., efficient indexing	No random access → Not suitable for operations requiring accessing elements by index such as sorting
No memory waste if the array is full or almost full; otherwise may result in much memory waste.	Since memory is allocated dynamically(acc. to our need) there is no waste of memory.
Sequential access is faster [Reason: Elements in contiguous memory locations]	Sequential access is slow [Reason: Elements not in contiguous memory locations]

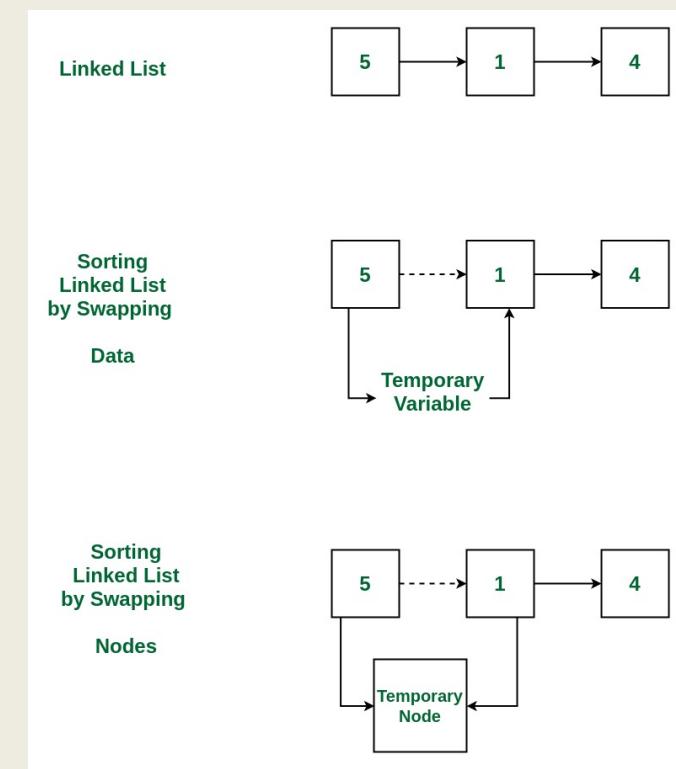
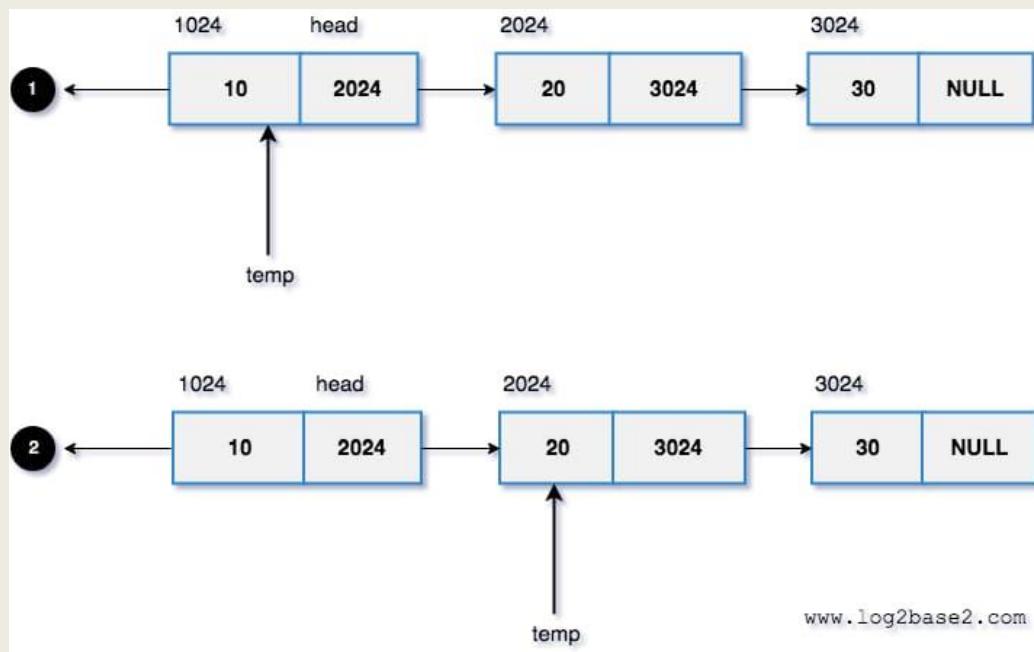
BASIC OPERATION ON LINKED LIST:

- ▶ The basic operations in the linked lists are insertion, deletion, searching and display an element at a given key. These operations are performed on Singly Linked Lists as given below –
- **Insertion** – Adds an element at the beginning of the list.
- **Deletion** – Deletes an element at the beginning of the list.
- **Display** – Displays the complete list.
- **Search** – Searches an element using the given key

INSERTION OPERATION & DELETION OPERATION



SEARCH & SORT OPERATION IN LINKED LIST



END

