# Data structure and Algorithms

By

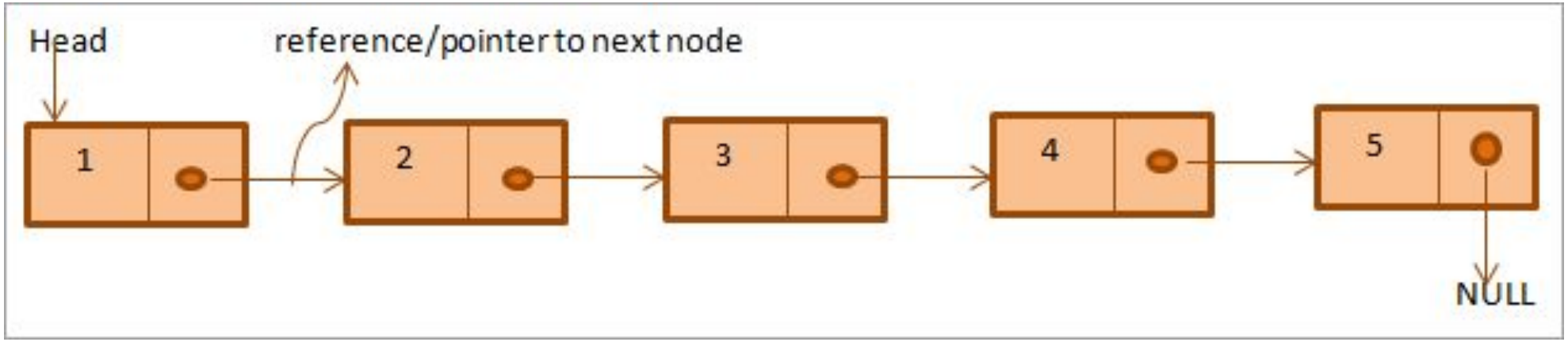
Engr Fatima Jaffar

# Linked List

- Linked list is a linear data structure that includes a series of connected nodes.
- includes a series of connected nodes.
- Linked list can be defined as the nodes that are randomly stored in the memory, not contiguous.

- A node in the linked list contains two parts,
  - i.e., first is the data part
  - and second is the address part.
- The last node of the list contains a pointer to the null.
- After array, linked list is the second most used data structure. In a linked list, every link contains a connection to another link.

# Representation of a linked list



# Why use linked list over array?

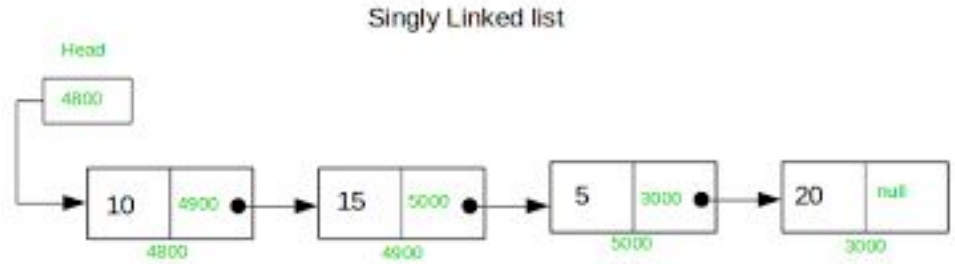
- The size of the array must be known in advance before using it in the program.
- Increasing the size of the array is a time taking process. It is almost impossible to expand the size of the array at run time.
- All the elements in the array need to be contiguously stored in the memory. Inserting an element in the array needs shifting of all its predecessors.

# Why use linked list over array?

- It allocates the memory dynamically. All the nodes of the linked list are non-contiguously stored in the memory and linked together with the help of pointers.
- In linked list, size is no longer a problem since we do not need to define its size at the time of declaration. List grows as per the program's demand and limited to the available memory space.

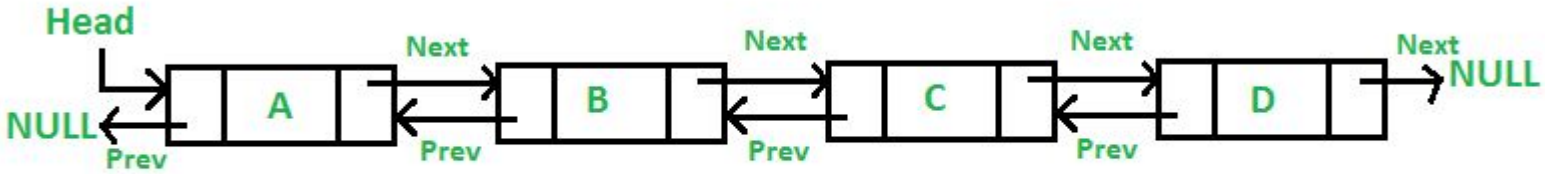
# Types of Linked List

- **Singly-linked list -** the collection of an ordered set of elements. A node in the singly linked list consists of two parts:
  - data part and
  - link part.
- Data part of the node stores actual information that is to be represented by the node
- the link part of the node stores the address of its immediate successor.



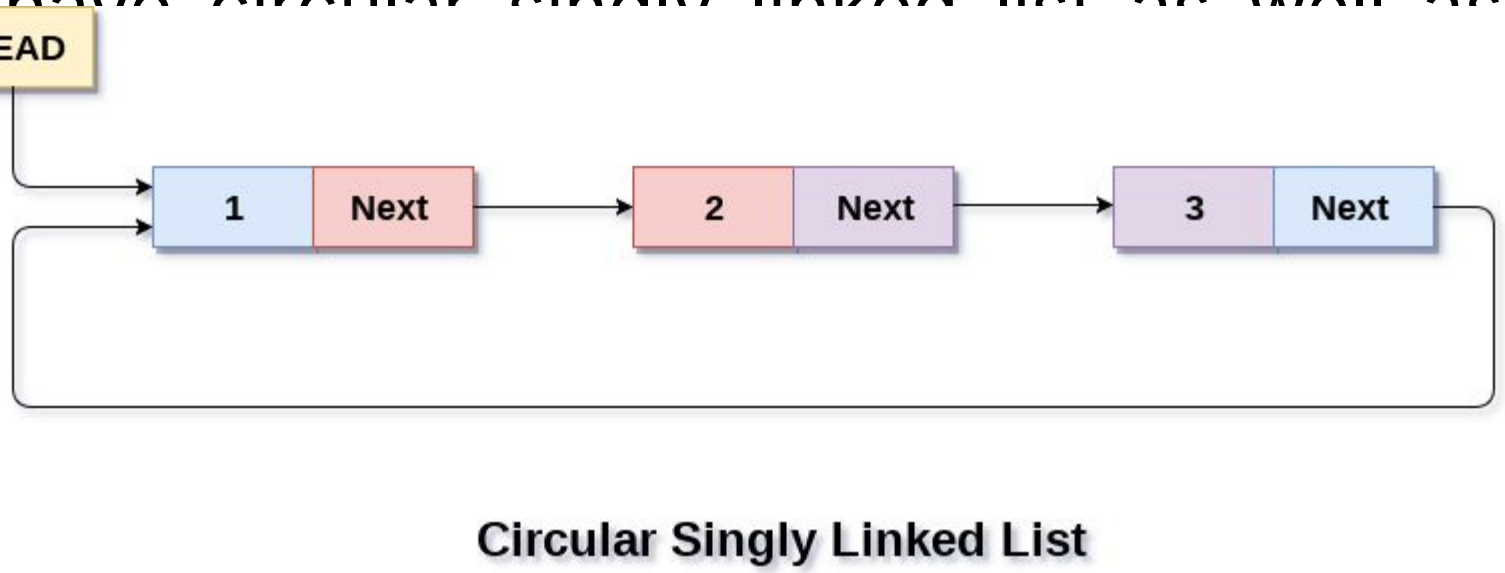
# Types of Linked List

- **Doubly linked list -** Doubly linked list is a complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence.
- Therefore, in a doubly-linked list, a node consists of three parts: node data, pointer to the next node in sequence (next pointer), and pointer to the previous node (previous pointe`



# Types of Linked Lists

- **Circular singly linked list -** In a circular singly linked list, the last node of the list contains a pointer to the first node of the list.
- We can have circular singly linked list as well as circular doubly




Circular Singly Linked List

- **Circular doubly linked list -** Circular doubly linked list is a more complex type of data structure in which a node contains pointers to its previous node as well as the next node. Circular doubly linked list doesn't contain NULL in any of the nodes.
- The last node of the list contains the address of the first node of the list.
- The first node of the list also contains the address of the last node in

# Applications of linked list

- A linked list can be used to represent the sparse matrix.
- The various operations like student's details, employee's details, or product details can be implemented using the linked list as the linked list uses the structure data type that can hold different data types.
- Using linked list, we can implement stack, queue, tree, and other various data structures.
- A linked list can be used to implement dynamic memory allocation. The dynamic memory allocation is the memory allocation done at the run-time

# Linked List in Java

- In Java, the linked list is implemented by the "***LinkedList***" class.
- This class belongs to the "***java.util***" package.

# Characterstics of Linked List

- This class is not synchronized.
- It allows duplicate values.
- Retains the insertion order.
- As elements are not required to be shifted while moving, the manipulation of elements in it is faster.
- This class can be used to implement a stack, queue, and list.

# How to create a linked list in java

- Before we move on to creating a linkedlist in Java,
  - let’s first discuss a linked list node in Java.
- As linked list consists of nodes.
- Thus in Java, we can represent a LinkedList as a class with its Node as a separate class.
- Hence this class will have a reference to the Node type

```
class LinkedList {
    Node head; // list head
      //node - linkedlist
    class Node {
        int data;
        Node next;


        Node(int d) { data = d; }  //constructor to create a new
node
    }
}
```

- **To create an object of type LinkedList, there are two main constructors as follows:**

**#1) LinkedList()**

**The general syntax for this constructor is:**

```
LinkedList<type> linkedList = new LinkedList<>();
```

- The above statement creates an empty LinkedList.
- **For example,**

```
LinkedList<Integer> l_list = new LinkedList<>();  //creates empty linked list with name l_list
```