CH - 4

Link-List

LIST OVERVIEW

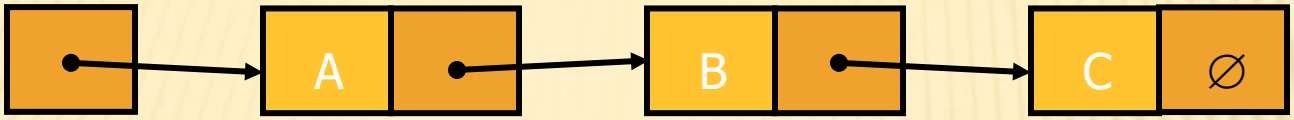
- Linked lists
  - Abstract data type (ADT)
- Basic operations of linked lists
  - Insert, find, delete, print, etc.
- Variations of linked lists
  - Circular linked lists
  - Doubly linked lists
- Use of both Method :
  - (LIFO –FIFO)

Types of Linked List

There are mainly three types of linked list

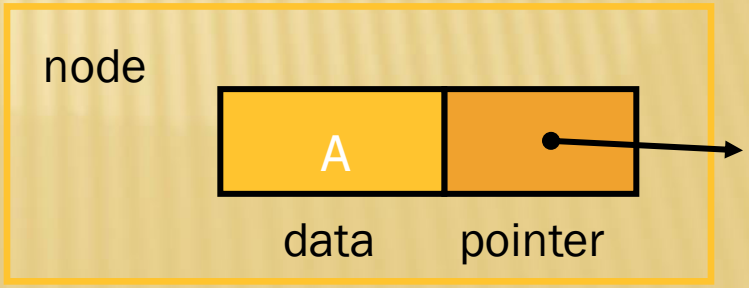
- **Singly linked list**
  - Each node has only one link part that contains the address of next node.
- **Circular linked list**
  - In this linked list the linked field of the last node contain the address of the first node of list
- **Doubly linked list**
  - In this linked list all nodes are linked together by multiple number of links which help in accessing both the successor and predecessor node from the given node position

LINKED LISTS

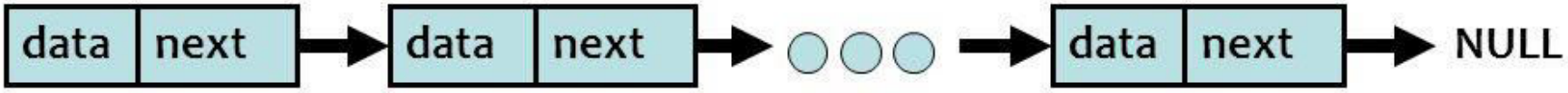


Head

- A *linked list* is a series of connected *nodes.*
- Each node contains at least
  - A piece of data (any type)
  - Pointer to the next node in the list
- *Head*: pointer to the first node
- The last node points to `NULL`



Linear Linked Lists : Definition



head head->next

```
struct node
        { int  data;  struct node   *next; };

// type name for new type is  "struct node"

struct node * head;  // declares the pointer for first node (head)
```

A SIMPLE LINKED LIST CLASS

- Operations of `List`
  - ***IsEmpty***: determine whether or not the list is empty.
  - ***InsertNode***: insert a new node at a particular position.
  - ***FindNode***: find a node with a given value.
  - ***DeleteNode***: delete a node with a given value.
  - ***DisplayList***: print all the nodes in the list

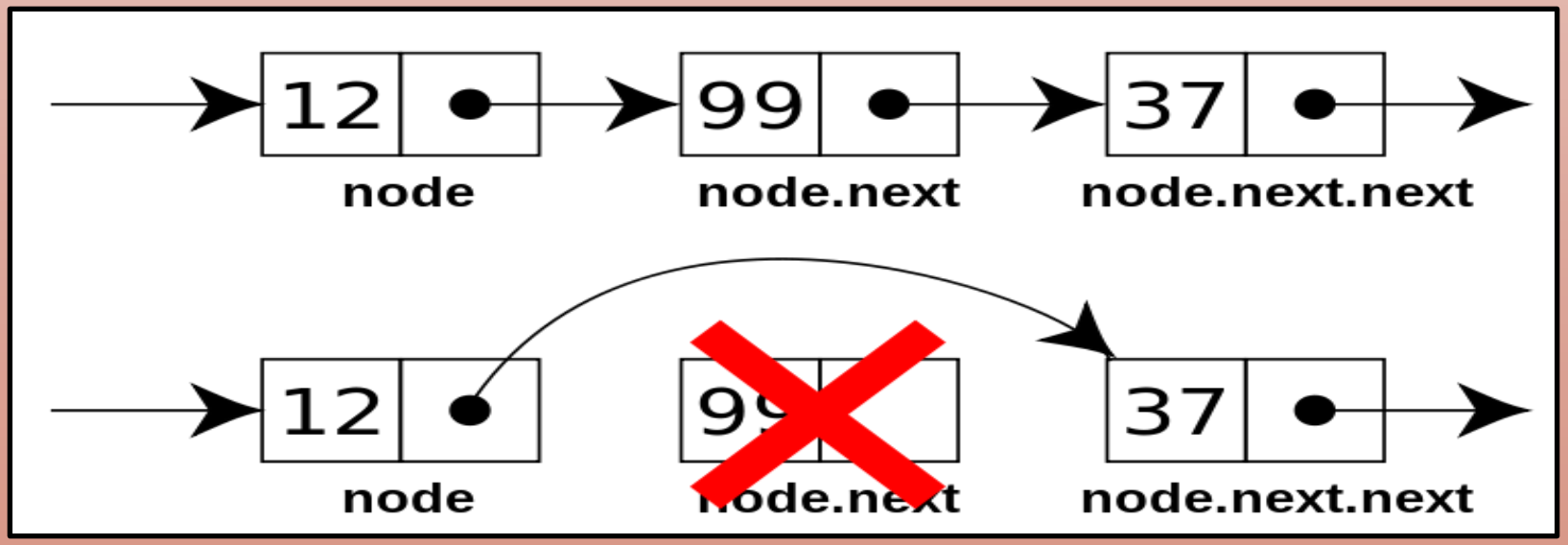
INSERTING A NEW NODE

- Possible cases of ***InsertNode***
  1. Insert into an empty list
  2. Insert in front
  3. Insert at back

- But, in fact, only need to handle two cases
  - Insert as the first node (Case 1)
  - Insert in the end of the list (Case 2)

DELETE NODE

- Possible cases of ***DeleteNode***
  1. Delete into an empty list
  2. Delete in front
  3. Delete at back
- But, in fact, only need to handle two cases
  - Delete as the first node (Case 4)
  - Delete in the end of the list (Case 5)



Operation of Linked List

Some basic operation of linked list are as follows:-

- **Insertion of a node**
  - Inserting a node before first node
  - Inserting a node after last node
  - Inserting a node at particular position
- **Deletion of a node**
  - deleting a node before first node
  - deleting a node after last node
  - deleting a node from particular position