Data Structures Lab

Session 05

Course: Data Structures (CS2001) **Semester:** Fall 2022
Instructor: Eman Shahid **T.A:** N/A

Note:

- Maintain discipline during the lab.
- Listen and follow the instructions as they are given.
- Just raise hand if you have any problem.
- Completing all tasks of each lab is compulsory.
- Get your lab checked at the end of the session.

Doubly Link List

```
class Node {
        public:
                int key;
                int data;
                Node * next;
                Node * previous;

                 Node() {
                        key = 0;
                        data = 0;
                        next = NULL;
                        previous = NULL;
                }

                Node(int k, int d) {
                        key = k;
                        data = d;
                }
};

class DoublyLinkedList {
        public:
                Node * head;

                DoublyLinkedList() {
                        head = NULL;
                }

                DoublyLinkedList(Node * n) {
                        head = n;
        }
```

```
    appendNode();
    prependNode();
    insertNodeAfter();
    deleteNodeByKey();
    updateNodeByKey();
};
```

Circular Link List

```
class Node {
    public:
        int key;
        int data;
        Node * next;

        Node() {
            key = 0;
            data = 0;
            next = NULL;
        }

        Node(int k, int d) {
            key = k;
            data = d;
        }
};

class CircularLinkedList {
    public:
        Node * head;

        CircularLinkedList() {
            head = NULL;
        }


    appendNode();
    prependNode();
    insertNodeAfter();
    deleteNodeByKey();
    updateNodeByKey();
    print();
};
```

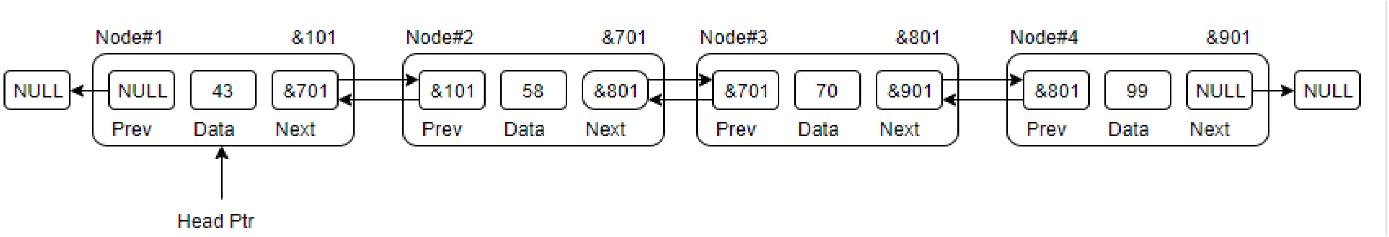
Circular Double Link List

```
class  node
{
public:
                int info;
                node *next;
                node *prev;
};

class double_clist
{
        public:
        node *create_node(int);

        insert_begin();
        insert_last();
        insert_pos();
        delete_pos();
         update();
        display();
        node  *start, *last;

        double_clist()
        {
                start = NULL;
                last = NULL;
        }
};
```

Reference:

1. Doubly Link List

Doubly Linked List is a variation of Linked list in which navigation is possible in both ways, either forward or backward easily as compared to Single Linked List.



- **Link** − each link of a linked list can store a data called an element.

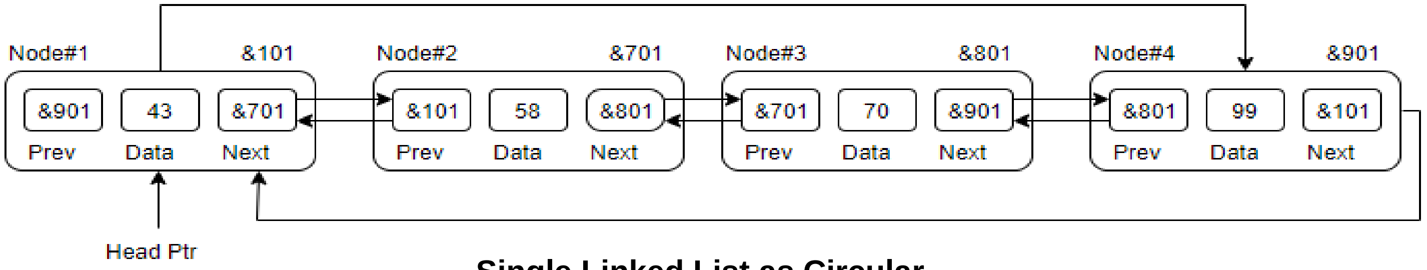
- **Next** − each link of a linked list contains a link to the next link called Next.
- **Prev** − each link of a linked list contains a link to the previous link called Prev.

2. **Circular Link List**

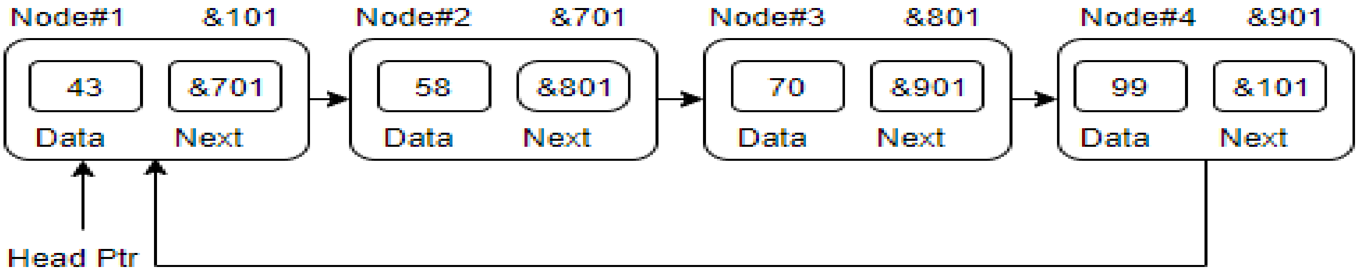
In doubly linked list, the next pointer of the last node points to the first node and the previous pointer of the first node points to the last node making the circular in both directions.

- The last link's next points to the first link of the list in both cases of singly as well as doubly linked list.
- The first link's previous points to the last of the list in case of doubly linked list.

Doubly Linked List as Circular



Single Linked List as Circular



Basic Operations

- Traverse
- Insert
- Display
- Append
- Delete
- Prepend
- Count

Lab6: Doubly Link List & Circular Link List	
Std Name:	**Std_ID:**

Lab1-Tasks	**Completed**	**Checked**
Task #1		
Task #2		
Task #3		
Task# 4		