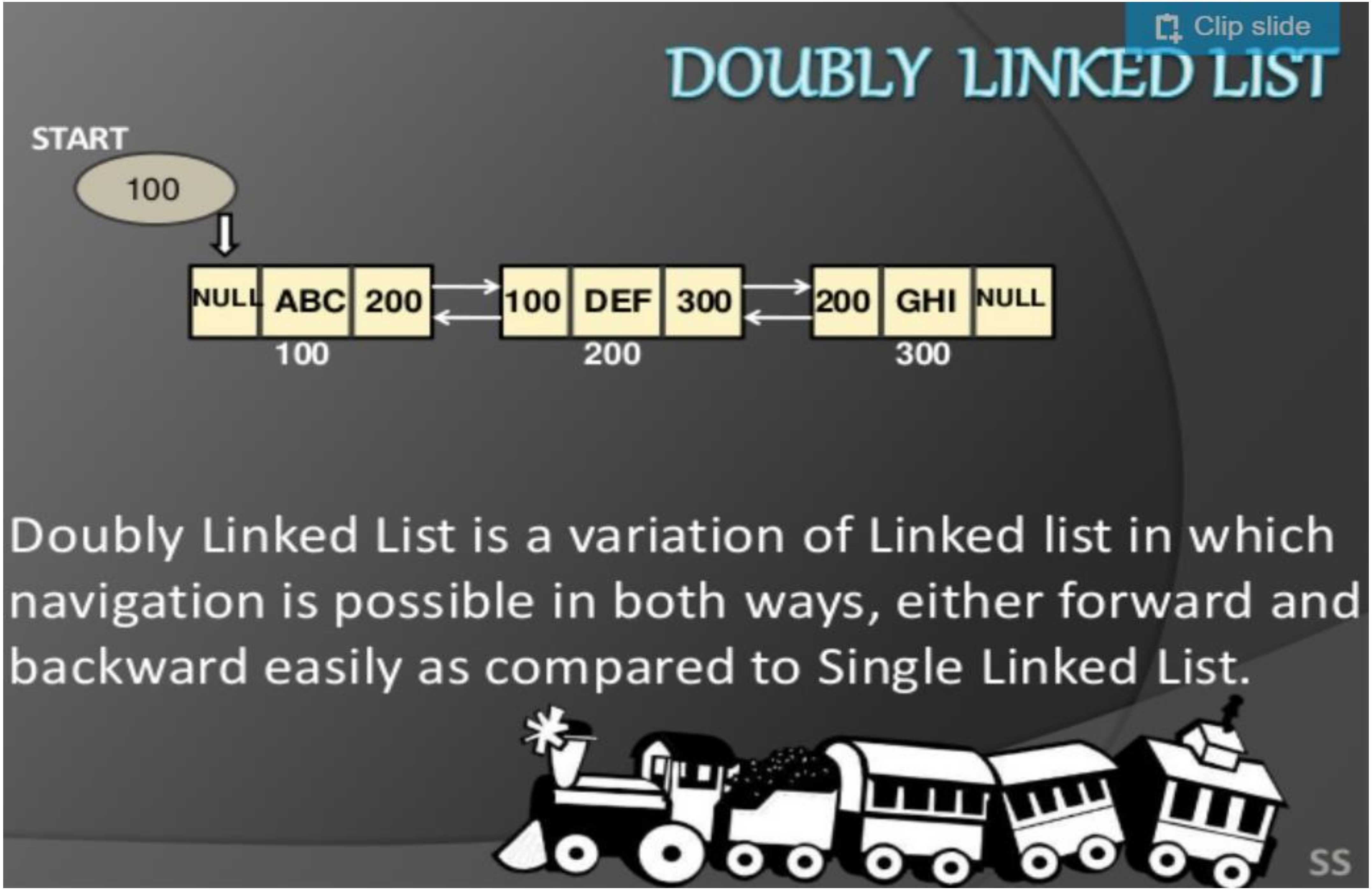
Doubly Linked List

Kiran Waghmare

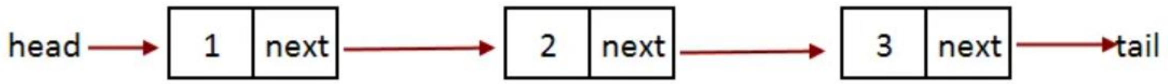


DOUBLY LINKED LIST

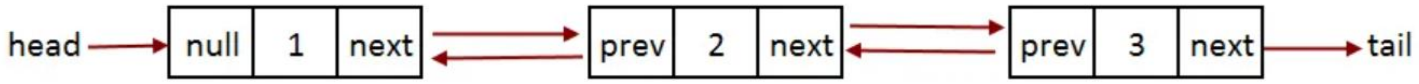
Doubly Linked List is a variation of Linked list in which navigation is possible in both ways, either forward and backward easily as compared to Single Linked List.

Singly Linked List vs Doubly Linked List

Singly Linked List	Doubly Linked List
Easy Implement	Not easy
Less memory	More Memory
Can traverse only in forward direction	Traverse in both direction, back and froth



Singly Linked List



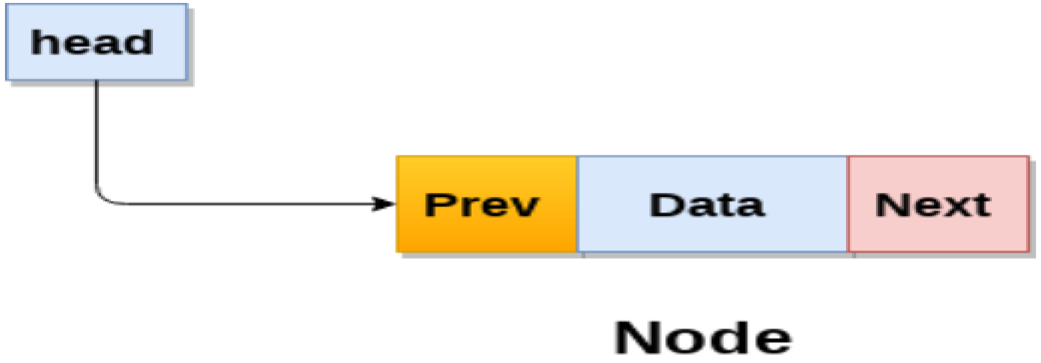
Doubly Linked List

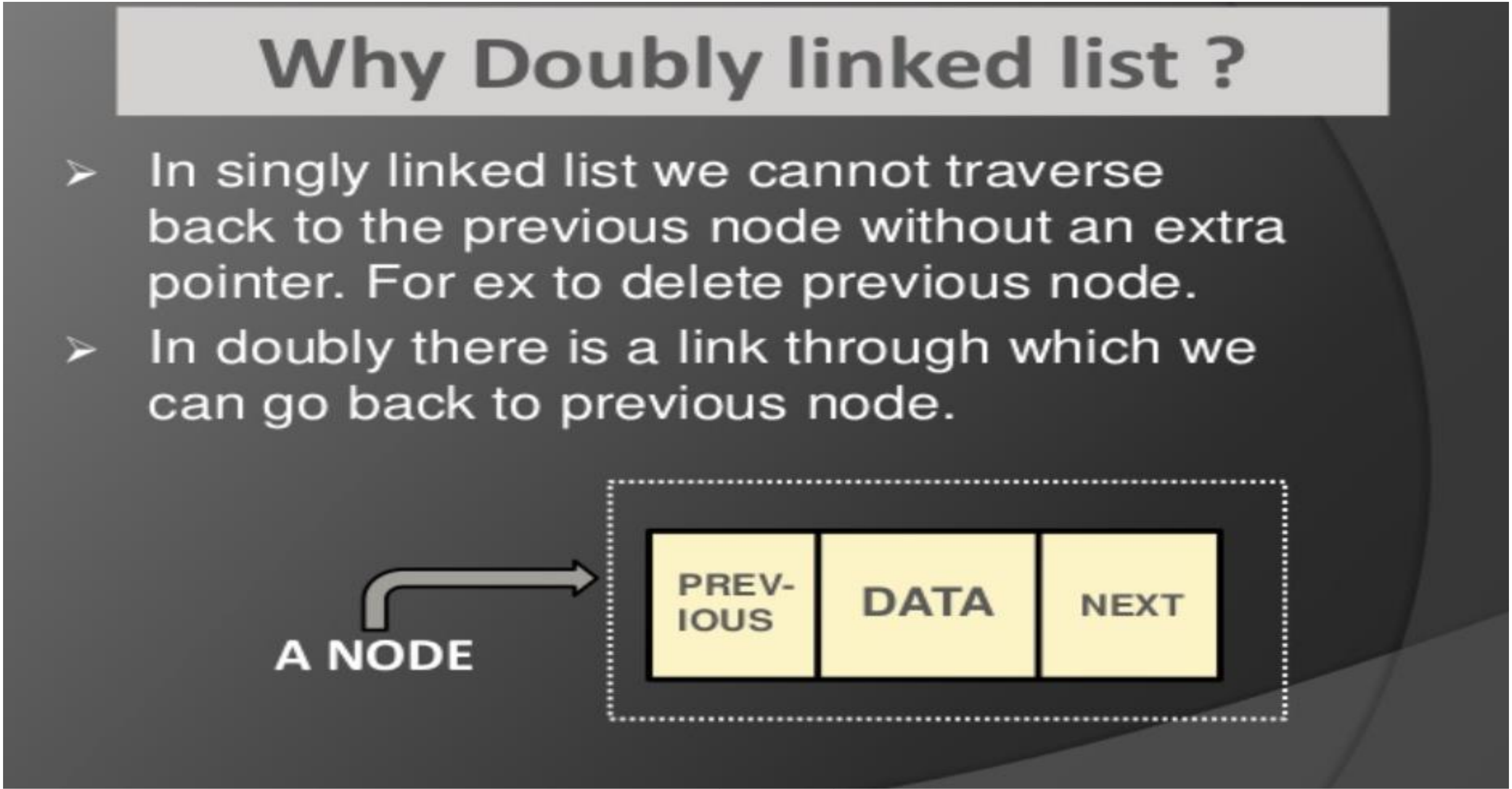
Doubly linked list

- 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.

- In a doubly linked list, a node consists of three parts:

1. Data
2. Pointer to the previous node
3. pointer to the next node

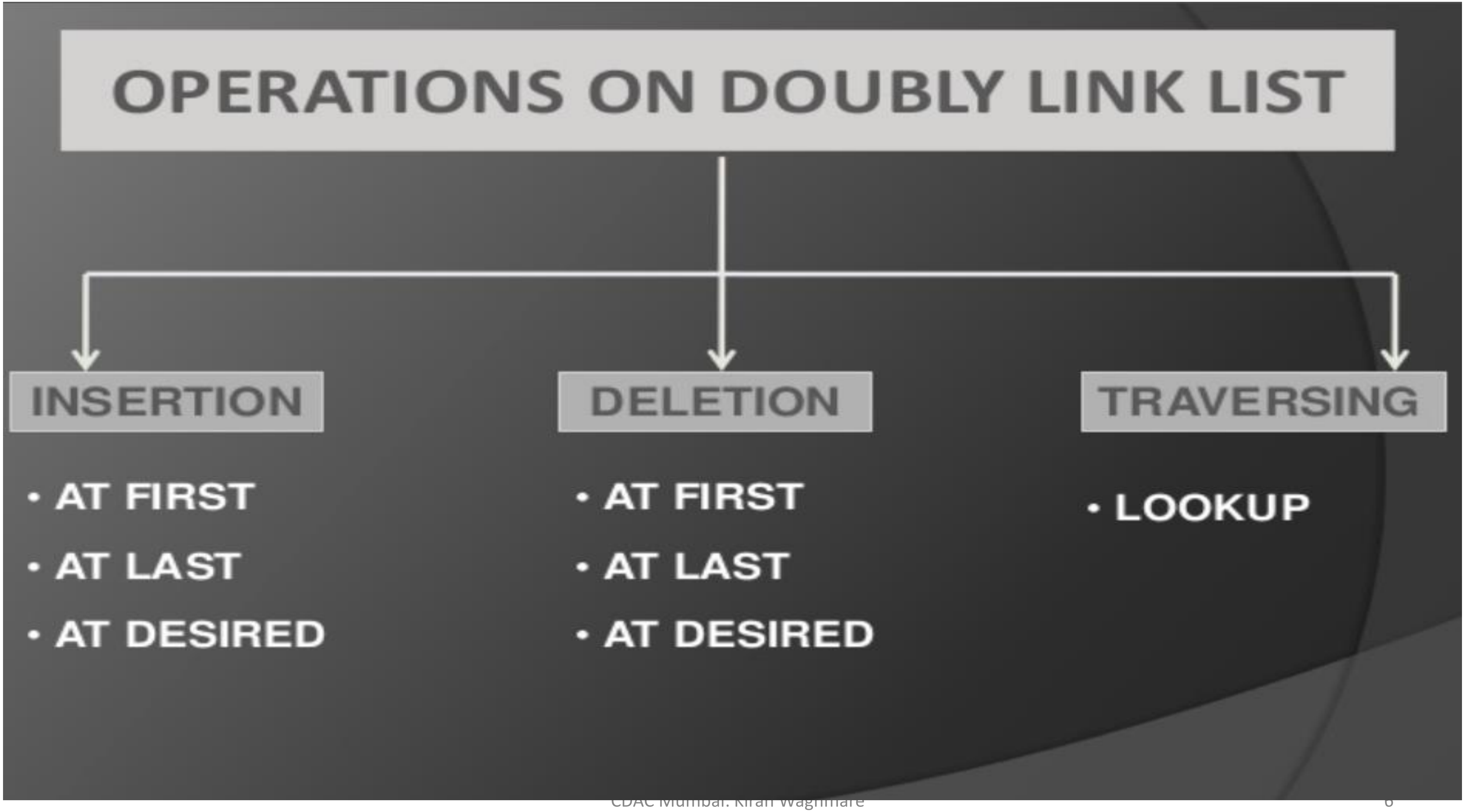




Why Doubly linked list ?

- In singly linked list we cannot traverse back to the previous node without an extra pointer. For ex to delete previous node.
- In doubly there is a link through which we can go back to previous node.

A NODE
PREV-
IOUS
DATA
NEXT



OPERATIONS ON DOUBLY LINK LIST

INSERTION

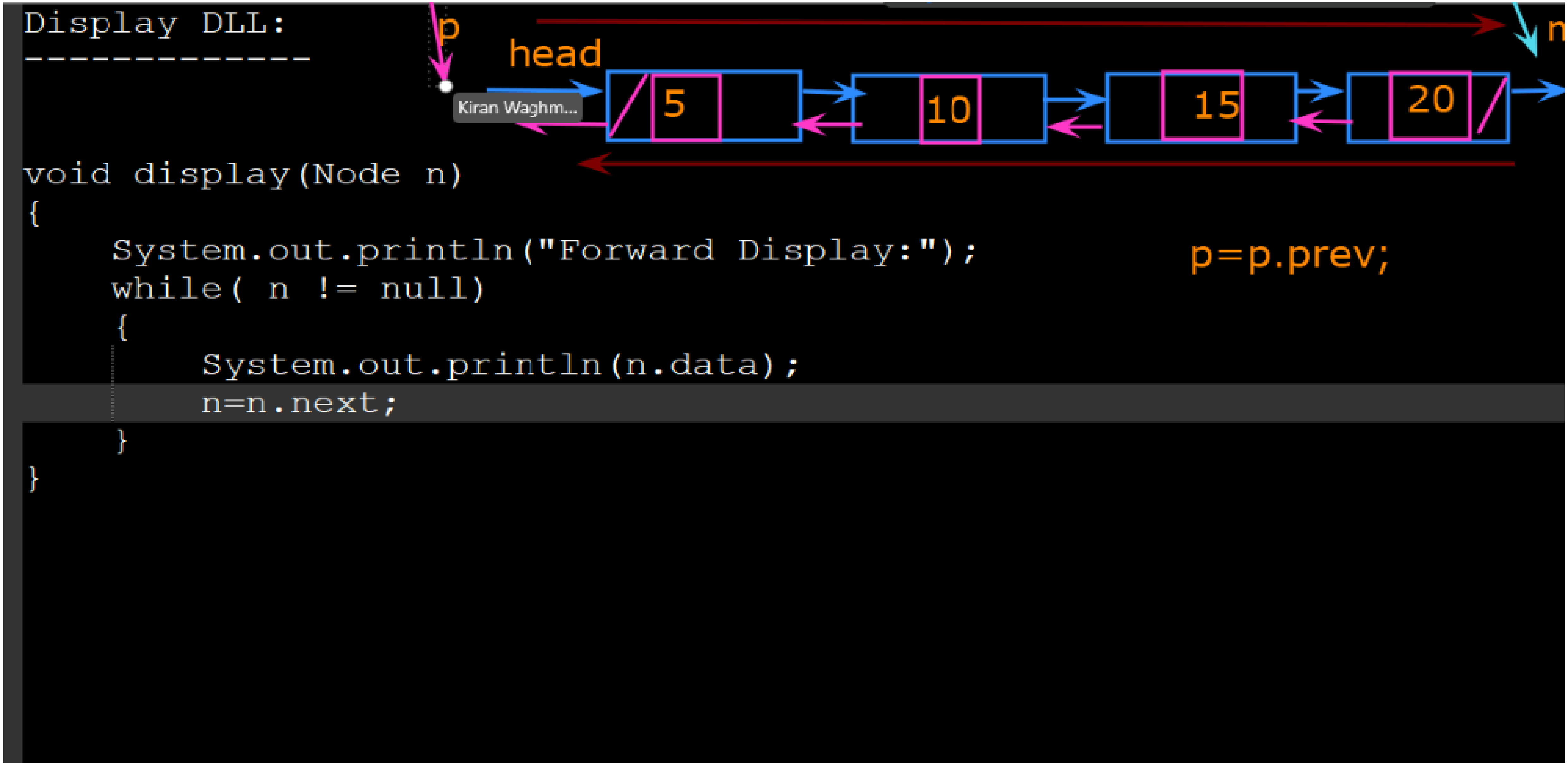
- AT FIRST
- AT LAST
- AT DESIRED

DELETION

- AT FIRST
- AT LAST
- AT DESIRED

TRAVERSING

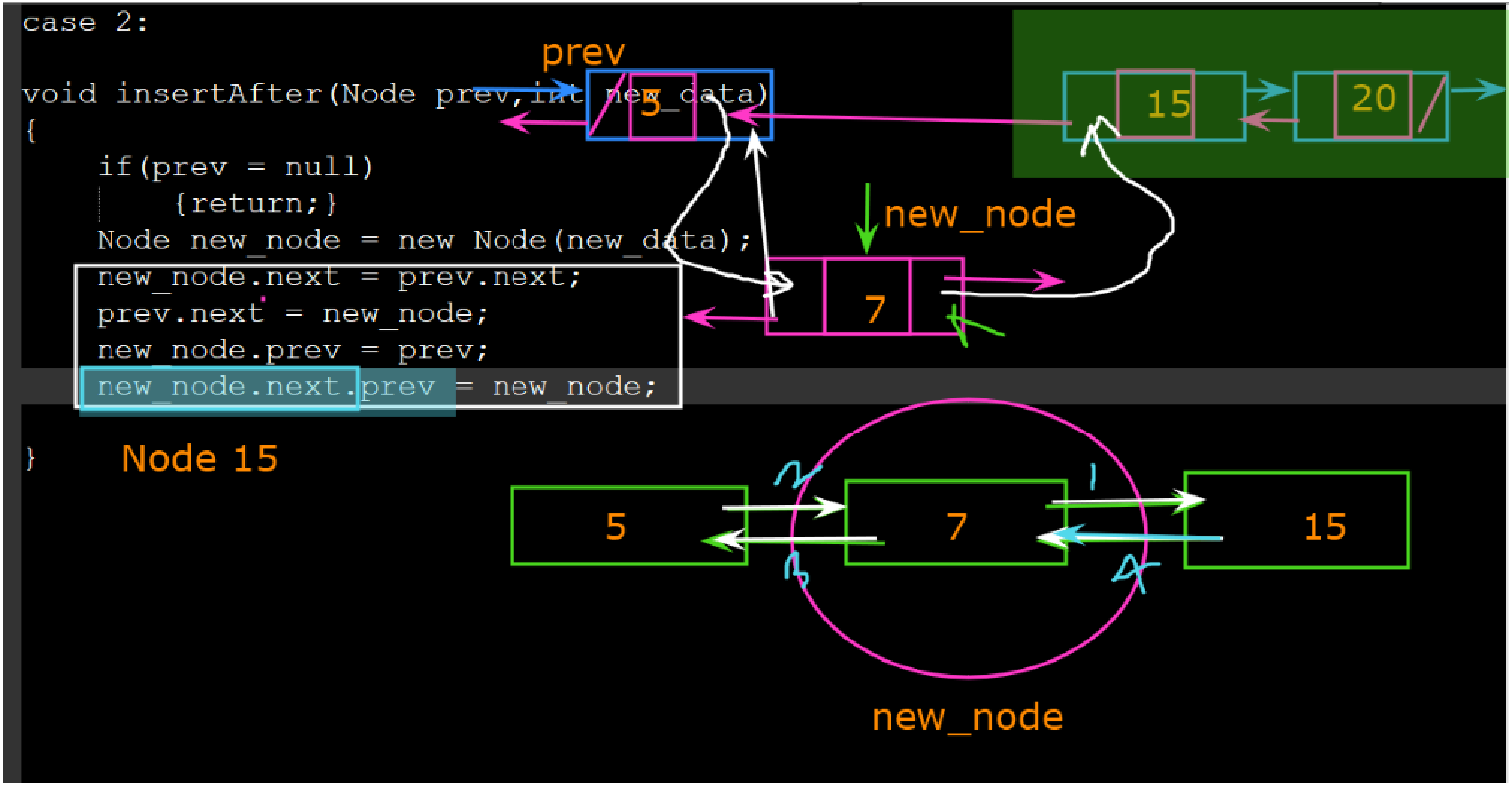
- LOOKUP



```
Display DLL:
-------------

void display(Node n)
{
     System.out.println("Forward Display:");
     while( n != null)
     {
          System.out.println(n.data);
          n=n.next;
     }
}
```

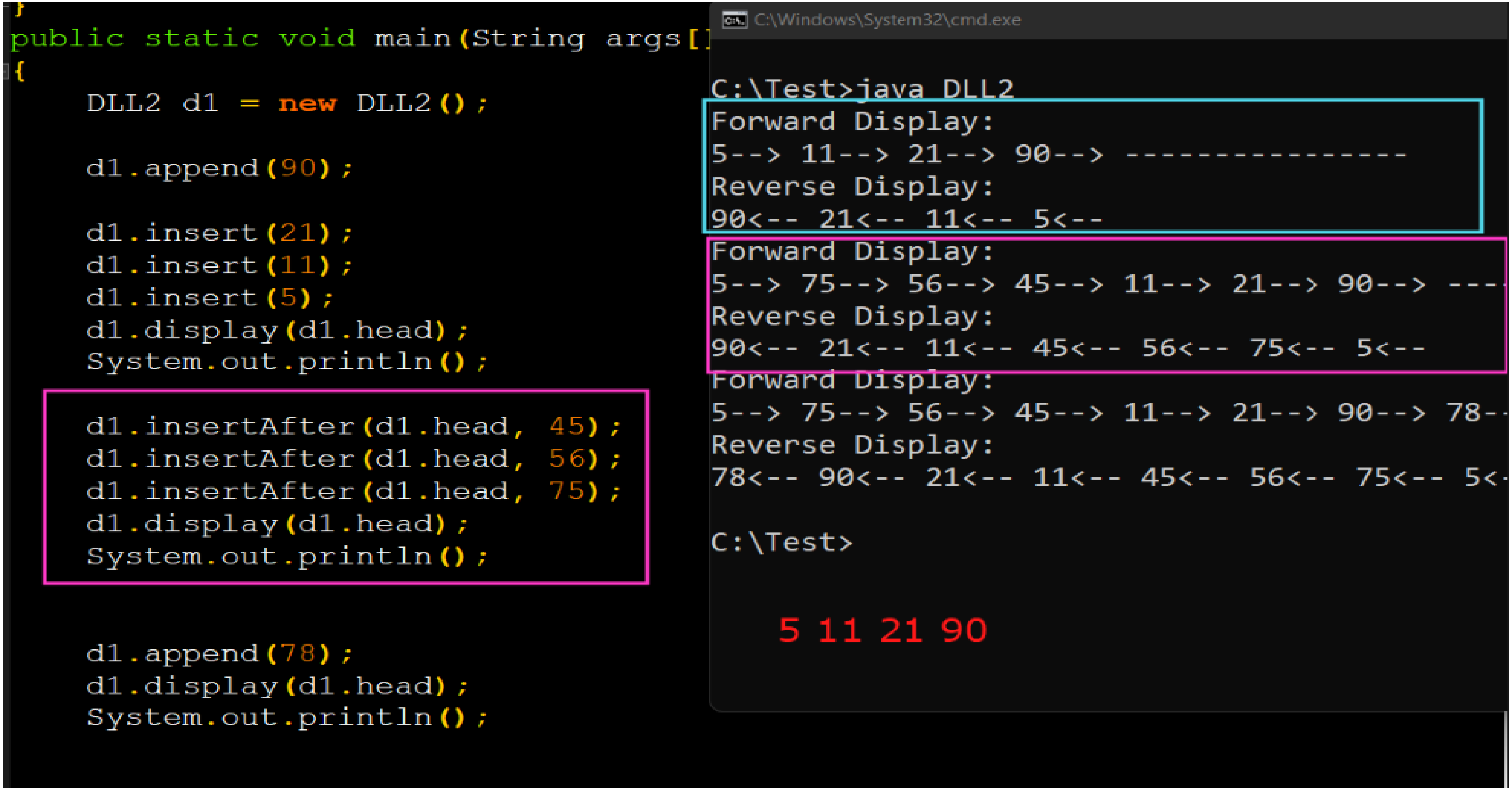
p=p.prev;



```
case 2:

void insertAfter(Node prev,int new_data)
{
     if(prev = null)
          {return;}
     Node new_node = new Node(new_data);
     new_node.next = prev.next;
     prev.next = new_node;
     new_node.prev = prev;
     new_node.next.prev = new_node;

}
```



```java
}
public static void main(String args[]
{
    DLL2 d1 = new DLL2();

    d1.append(90);

    d1.insert(21);
    d1.insert(11);
    d1.insert(5);
    d1.display(d1.head);
    System.out.println();

    d1.insertAfter(d1.head, 45);
    d1.insertAfter(d1.head, 56);
    d1.insertAfter(d1.head, 75);
    d1.display(d1.head);
    System.out.println();


    d1.append(78);
    d1.display(d1.head);
    System.out.println();
```

C:\Windows\System32\cmd.exe

```
C:\Test>java DLL2
Forward Display:
5--> 11--> 21--> 90--> ----------------
Reverse Display:
90<-- 21<-- 11<-- 5<--
Forward Display:
5--> 75--> 56--> 45--> 11--> 21--> 90--> ---
Reverse Display:
90<-- 21<-- 11<-- 45<-- 56<-- 75<-- 5<--
Forward Display:
5--> 75--> 56--> 45--> 11--> 21--> 90--> 78--
Reverse Display:
78<-- 90<-- 21<-- 11<-- 45<-- 56<-- 75<-- 5<-

C:\Test>
```

5 11 21 90

Thanks