

# Doubly Linked List

21 January 2026 20:02

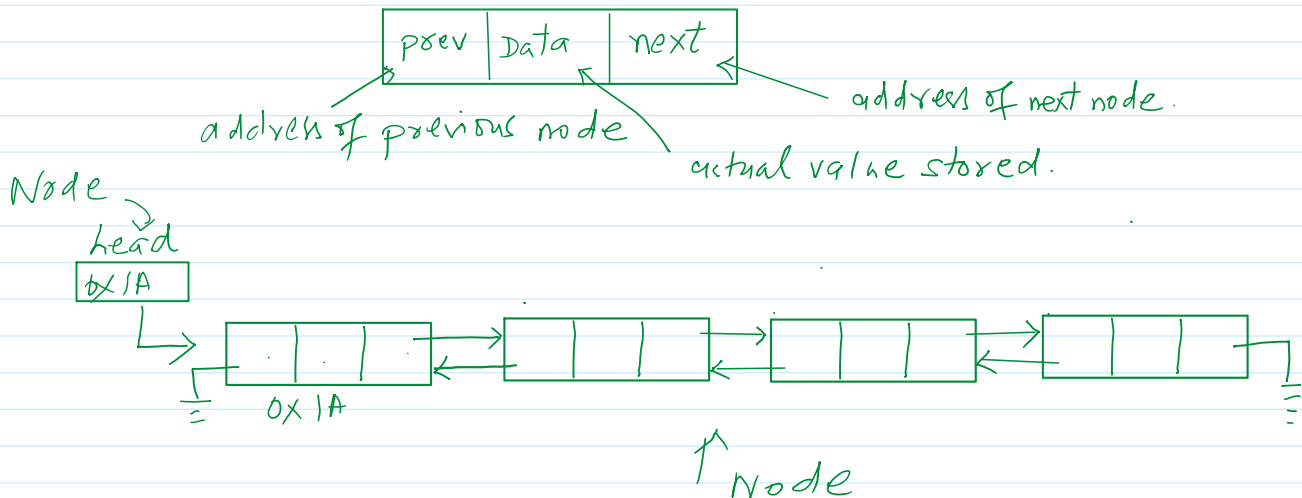
What is a Doubly Linked List?

A Doubly Linked List(DLL) is a linear dynamic data structure in which each node contains:

- Data
- Reference to previous node
- Reference to next node

Unlike a singly linked list traversal in doubly linked list is possible in both forward and backward direction.

Structure of a Node in Doubly Linked List:



Advantages of doubly linked list:

- Traversal in both directions
- Easy deletion of a node (no need to track previous separately)
- Efficient for applications like:
  - ◆ Undo/Redo operations
  - ◆ Browser navigation
  - ◆ Music playlist(next/previous song)
  - ◆ ...many more

Disadvantage of Doubly Linked List:

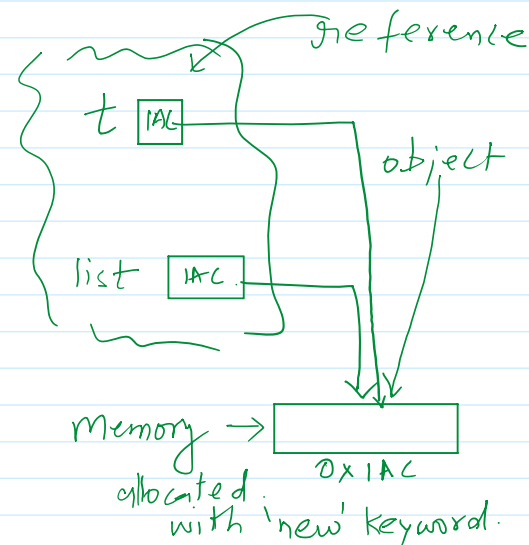
- Requires extra memory for previous reference
- More complex implementation than singly linked list

What is reference variable and object?

DoublyLinkedList list; *Reference variable*

list = new DoublyLinkedList();

DoublyLinkedList t = list;



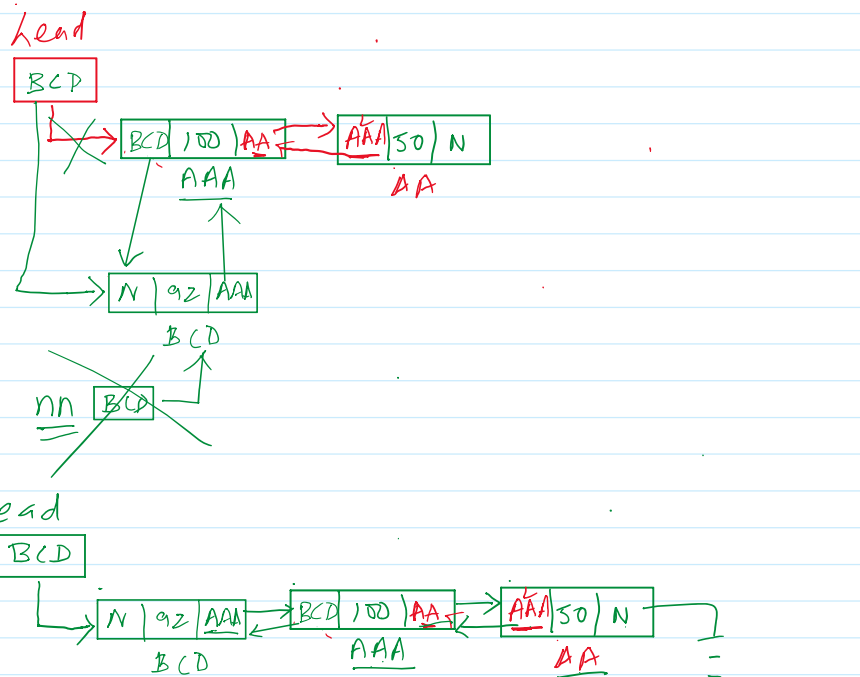
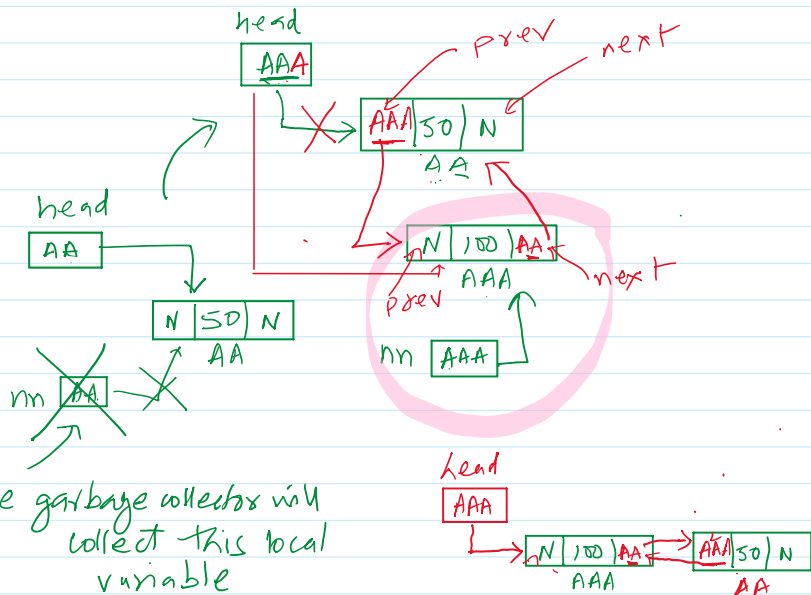
The variable name is called the reference, and the memory which they points is called object, therefore the variable and the memory together is called object, but when the same memory is pointed by another variable is called reference variable.

```

package LinkedList;
import static java.lang.System.out;
public class DoublyLinkedList {
    private Node head; //reference variable
    public DoublyLinkedList() {
        head = null;
    }

    public void insertAtHead(int data){
        Node nn = new Node(data);
        → if (head != null){
            → head.setPrev(nn);
            → nn.setNext(head);
        }
        → head = nn;
    }
    public void display(){
        if(head == null){
            out.print("\nList is empty");
        }
        else{
            for(Node t = head; t!=null; t = t.getNext()) //t is a reference variable
            {
                out.print("\n"+t);
            }
        }
    }
}

```



```

public void insertAtHead(int data){
    → Node nn = new Node(data);
    → if (head != null){
        → head.setPrev(nn);
        → nn.setNext(head);
    }
    → head = nn;
}

```

$t = \text{BCD} \rightarrow \text{AAA} \rightarrow \text{AA} \rightarrow \text{N}$

```

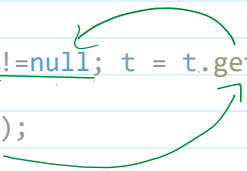
public void display(){
    if(head == null){
        out.print("\nList is empty");
    }
    else{

```

```

        out.print("\nList is empty");
    }
    else{
        for(Node t = head; t!=null; t = t.getNext()) //t is a reference variable
        {
            →out.print("\n"+t);
        }
    }
}

```


  
 [Node = 92]
   
 [Node = 100]
   
 [Node = 50]

```

package LinkedList;
public class Node {
    private int data;
    private Node prev;
    private Node next;
    public Node(int data) {
        this.data = data;
        this.next = this.prev = null;
    }
    public int getData() {
        return data;
    }
    public void setData(int data) {
        this.data = data;
    }
    public Node getPrev() {
        return prev;
    }
    public void setPrev(Node prev) {
        this.prev = prev;
    }
    public Node getNext() {
        return next;
    }
    public void setNext(Node next) {
        this.next = next;
    }
    @Override
    public String toString() {
        return "Node [data=" + data + "]";
    }
}

```

```

package LinkedList;
import static java.lang.System.out;
public class DoublyLinkedList {
    private Node head; //reference variable
    public DoublyLinkedList() {
        head = null;
    }
}

```

```

    public void insertAtHead(int data){
        Node nn = new Node(data);
        if (head != null){
            head.setPrev(nn);
            nn.setNext(head);
        }
        head = nn;
    }
    public void display(){
        if(head == null){
            out.print("\nList is empty");
        }
        else{
            for(Node t = head; t!=null; t = t.getNext()) //t is a reference variable
            {
                out.print("\n"+t);
            }
        }
    }
}

```

```

package LinkedList;
import java.util.Scanner;
public class MainClass {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DoublyLinkedList ddl = new DoublyLinkedList();
        while(true){
            System.out.println("\nPress 1 to insert from start");
            System.out.println("Press 2 to display");
            System.out.println("Press 0 to exit");
            System.out.println("Enter your choice: ");
            int ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("Enter data to insert from start: ");
                    ddl.insertAtHead(sc.nextInt());
                    break;
                case 2:
                    ddl.display();
                    break;
                case 0:
                    System.out.println("Good bye");
                    System.exit(ch);
                default:
                    System.out.println("Wrong option selected!");
            }
        }
    }
}

```

Try this code and write down public void insertAtEnd(int data) method.

