Delete node with only pointer given to it

A **simple solution** is to traverse the linked list until you find the node you want to delete. But this solution requires a pointer to the head node which contradicts the problem statement.

The **fast solution** is to copy the data from the next node to the node to be deleted and delete the next node. Something like this:

It is important to note that this approach will only work if it is **guaranteed** that the given pointer does **not** point to the **last node**. Because if it is the last node, then you don't have a next node to copy the data from.

```
struct Node *temp  = node_ptr->next;
node_ptr->data  = temp->data;
node_ptr->next  = temp->next;
free(temp);
```

Below is the implementation of the above code:

C++ | Java

```
// Java program to del the node in
// which only a single pointer is
// known pointing to that node
class LinkedList {

    static Node head;
    static class Node {

        int data;
        Node next;
        Node(int d)
        {
            data = d;
            next = null;
        }
```

```
    void printList(Node node)
    {
        while (node != null) {
            System.out.print(node.data + " ");
            node = node.next;
        }
    }

    void deleteNode(Node node)
    {
        Node temp = node.next;
        node.data = temp.data;
        node.next = temp.next;
        System.gc();
    }

    // Driver code
    public static void main(String[] args)
    {
        LinkedList list = new LinkedList();
        list.head = new Node(1);
        list.head.next = new Node(12);
        list.head.next.next = new Node(1);
        list.head.next.next.next = new Node(4);
        list.head.next.next.next.next = new Node(1);

        // print the list
        System.out.println("Before deleting ");
        list.printList(head);

        /* I m deleting the head itself.
        You can check for more cases */
        list.deleteNode(head);
        System.out.println("");
        System.out.println("After deleting ");
        list.printList(head);
    }
}
```


Track Progress

89 of 132 Complete. (68%)

```
Before deleting
```

Time complexity: O(1) since performing constant operations and modifying only single pointer to delete node

Auxiliary Space: O(1)

Mark as Read

Report An Issue

If you are facing any issue on this page. Please let us know.