Algorithm for deleting a node from a singly linked list

**Algorithm** delete (front, nodeToDelete)

**In:** front node of linked list and node to delete

**Out:** *true* if the node was deleted, *false* otherwise

current = front
prev = null

while (current != null) and (current != nodeToDelete) do {
    prev = current
    current = current.next
}

if current == null then return *false*

else {
    if prev != null then prev.next = current.next
    else front = front.next
    return *true*
}


Doubly Linked List

Node object

front  tail

prev  element  next
Java Class for a Node of a Doubly Linked List

```java
public class LinearNodeDLL<T> {
    private LinearNodeDLL<T> next;
    private LinearNodeDLL<T> prev;
    private T element;

    public LinearNodeDLL( ) {
        next = null;
        prev = null;
        element = null;
    }

    public LinearNodeDLL (T dataItem) {
        next = null;
        prev = null;
        element = dataItem;
    }
}
```