***Delete First Node of Singly Linked List***

Given a linked list, the task is to remove the first node of the linked list and update the head pointer of the linked list.

**Examples:**

***Input :****1 -> 2 -> 3 -> 4 -> 5 -> NULL*  
***Output :****2 -> 3 -> 4 -> 5 -> NULL*

***Input :****2 -> 4 -> 6 -> 8 -> 33 -> 67 -> NULL*  
***Output :****4 -> 6 -> 8 -> 33 -> 67 -> NULL*

To remove the first node, we need to make the second node as head and delete the memory allocated for the first node.

**Implementation:**

C++Java

// Java program to remove first node of

// linked list.

class GFG {

// Link list node /

static class Node {

int data;

Node next;

};

// Function to remove the first node

// of the linked list /

static Node removeFirstNode(Node head)

{

if (head == null)

return null;

// Move the head pointer to the next node

Node temp = head;

head = head.next;

return head;

}

// Function to push node at head

static Node push(Node head\_ref, int new\_data)

{

Node new\_node = new Node();

new\_node.data = new\_data;

new\_node.next = (head\_ref);

(head\_ref) = new\_node;

return head\_ref;

}

// Driver code

public static void main(String args[])

{

// Start with the empty list /

Node head = null;

// Use push() function to con

// the below list 8 . 23 . 11 . 29 . 12 /

head = push(head, 12);

head = push(head, 29);

head = push(head, 11);

head = push(head, 23);

head = push(head, 8);

head = removeFirstNode(head);

for (Node temp = head; temp != null; temp = temp.next)

System.out.print(temp.data + " ");

}

}

**Output**

23 11 29 12

**Time complexity:** **O(1)**  
**Auxiliary Space: O(1)**