Inserting At Specific location in the SLL (After a Node)

We can use the following steps to insert a new node after a node in the single linked list.

Step 1: Create a **newNode** with given value.

Step 2: Check whether list is **Empty** (**head** == **NULL**)

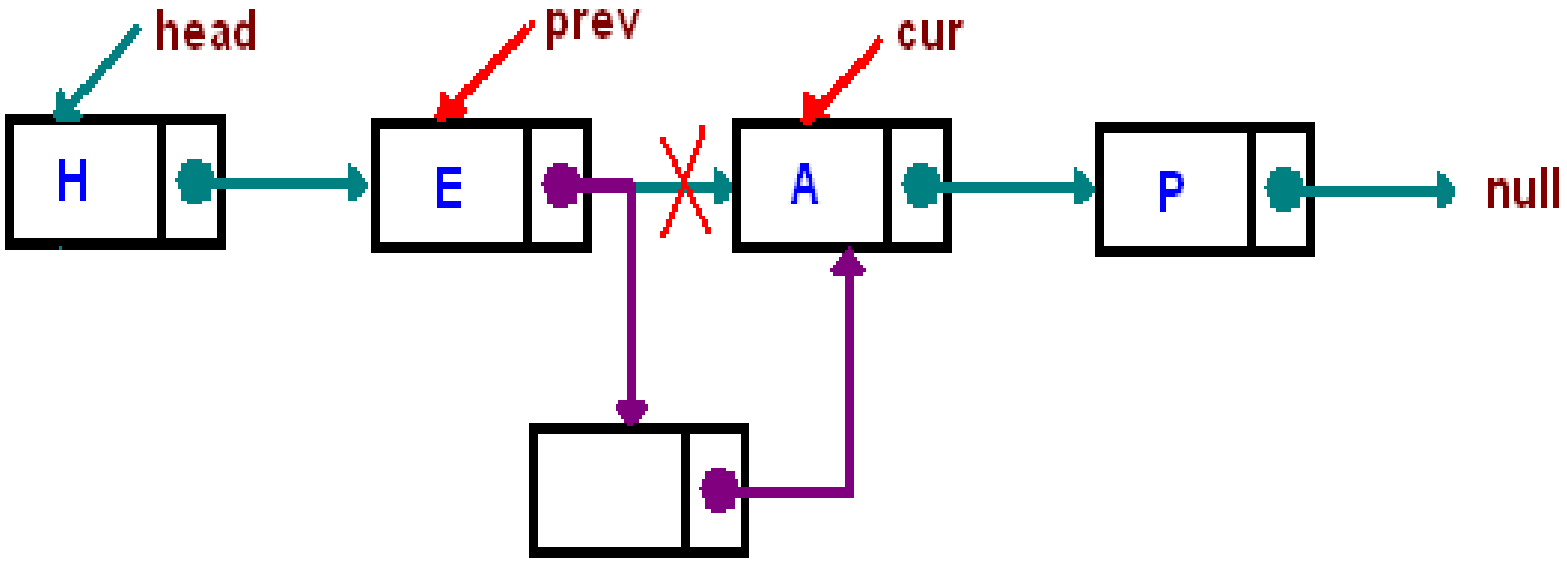
Step 3: If it is **Empty** then, set **newNode → next** = **NULL** and **head** = **newNode**.

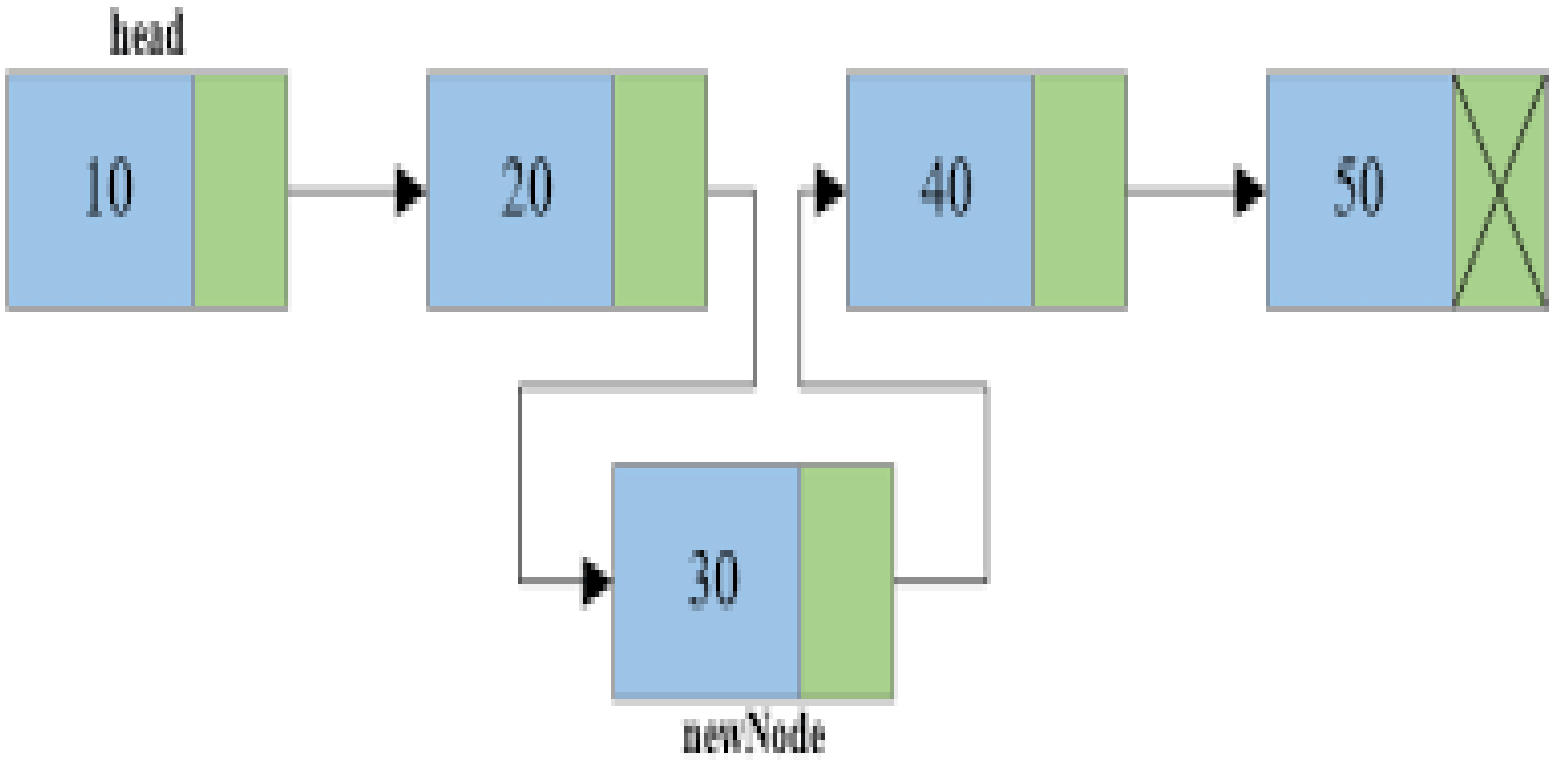
Step 4: If it is **Not Empty** then, define a node pointer **temp** and initialize with **head**.

Step 5: Keep moving the **temp** to its next node until it reaches to the node after which we want to insert the newNode (until **temp1 → data** is equal to **location**, here location is the node value after which we want to insert the newNode).

Step 6: Every time check whether **temp** is reached to last node or not. If it is reached to last node then display **'Given node is not found in the list!!! Insertion not possible!!!'** and terminate the function. Otherwise move the **temp** to next node.

Step 7: Finally, Set '**newNode → next** = **temp → next**' and '**temp → next** = **newNode**


head
prev
cur
H
E
A
P
null


head
10
20
40
50
30
newNode

Deletion in SLL

In a single linked list, the deletion operation can be performed in three ways. They are as follows...

1. Deleting from Beginning of the list
2. Deleting from End of the list
3. Deleting a Specific Node

Deleting from Beginning of the SLL

We can use the following steps to delete a node from beginning of the single linked list...

Step 1: Check whether list is **Empty** (**head** == **NULL**)

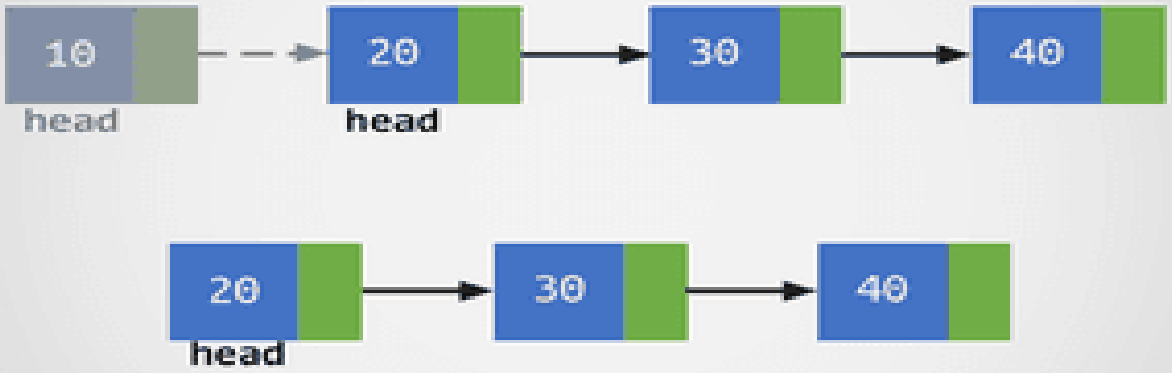
Step 2: If it is **Empty** then, display **'List is Empty!!! Deletion is not possible'** and terminate the function.

Step 3: If it is **Not Empty** then, define a Node pointer **'temp'** and initialize with **head**.

Step 4: Check whether list is having only one node (**temp → next** == **NULL**)

Step 5: If it is **TRUE** then set **head** = **NULL** and delete **temp** (Setting **Empty** list conditions)

Step 6: If it is **FALSE** then set **head** = **temp → next**, and delete **temp**.



Delete first element in linked list

Deleting from End of the SLL

We can use the following steps to delete a node from end of the single linked list...

Step 1: Check whether list is **Empty** (**head** == **NULL**)

Step 2: If it is **Empty** then, display **'List is Empty!!! Deletion is not possible'** and terminate the function.

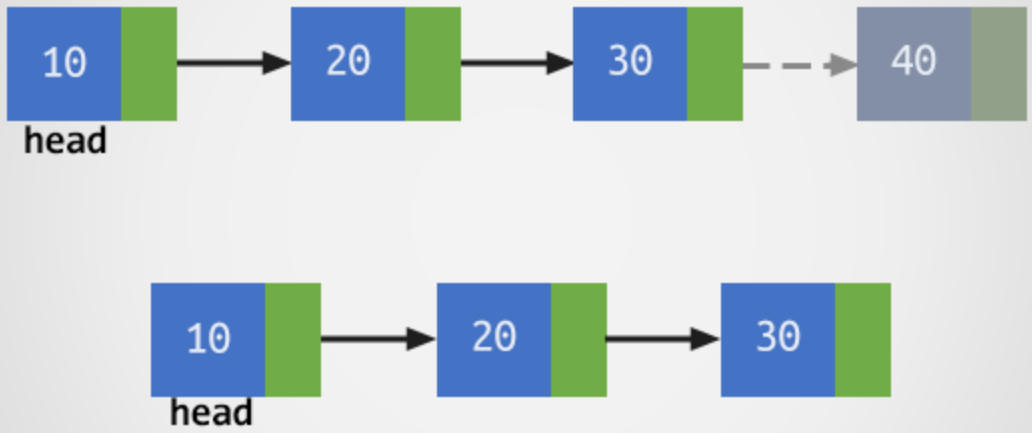
Step 3: If it is **Not Empty** then, define two Node pointers **'temp1'** and '**temp2'** and initialize '**temp1**' with **head**.

Step 4: Check whether list has only one Node (**temp1 → next** == **NULL**)

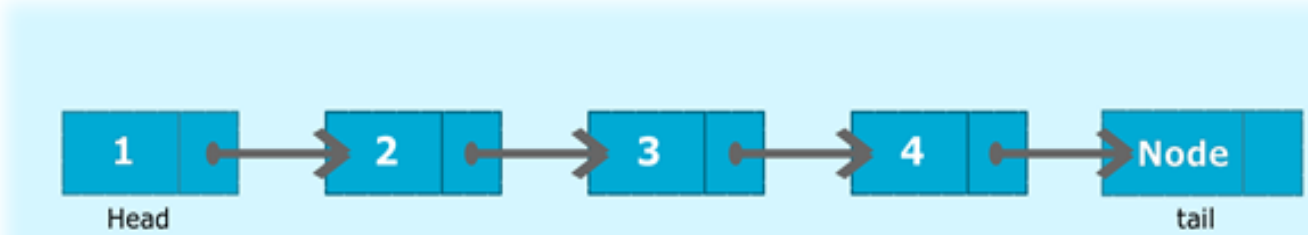
Step 5: If it is **TRUE**. Then, set **head** = **NULL** and delete **temp1**. And terminate the function. (Setting **Empty** list condition)

Step 6: If it is **FALSE**. Then, set '**temp2 = temp1** ' and move **temp1** to its next node. Repeat the same until it reaches to the last node in the list. (until **temp1 → next** == **NULL**)

Step 7: Finally, Set **temp2 → next** = **NULL** and delete **temp1**.



Delete last element in linked list


1
2
3
4
Node
Head
tail


1
2
3
4
Head
tail
Node

Deleting a Specific Node from the SLL

We can use the following steps to delete a specific node from the single linked list.

Step 1: Check whether list is **Empty** (**head** == **NULL**)

Step 2: If it is **Empty** then, display **'List is Empty!!! Deletion is not possible'** and terminate the function.

Step 3: If it is **Not Empty** then, define two Node pointers **'temp1'** and '**temp2**' and initialize '**temp1**' with **head**.

Step 4: Keep moving the **temp1** until it reaches to the exact node to be deleted or to the last node. And every time set '**temp2 = temp1**' before moving the '**temp1**' to its next node.

Step 5: If it is reached to the last node then display **'Given node not found in the list! Deletion not possible!!!'**. And terminate the function.

Step 6: If it is reached to the exact node which we want to delete, then check whether list is having only one node or not

Step 7: If list has only one node and that is the node to be deleted, then set **head** = **NULL** and delete **temp1** (**free(temp1)**).

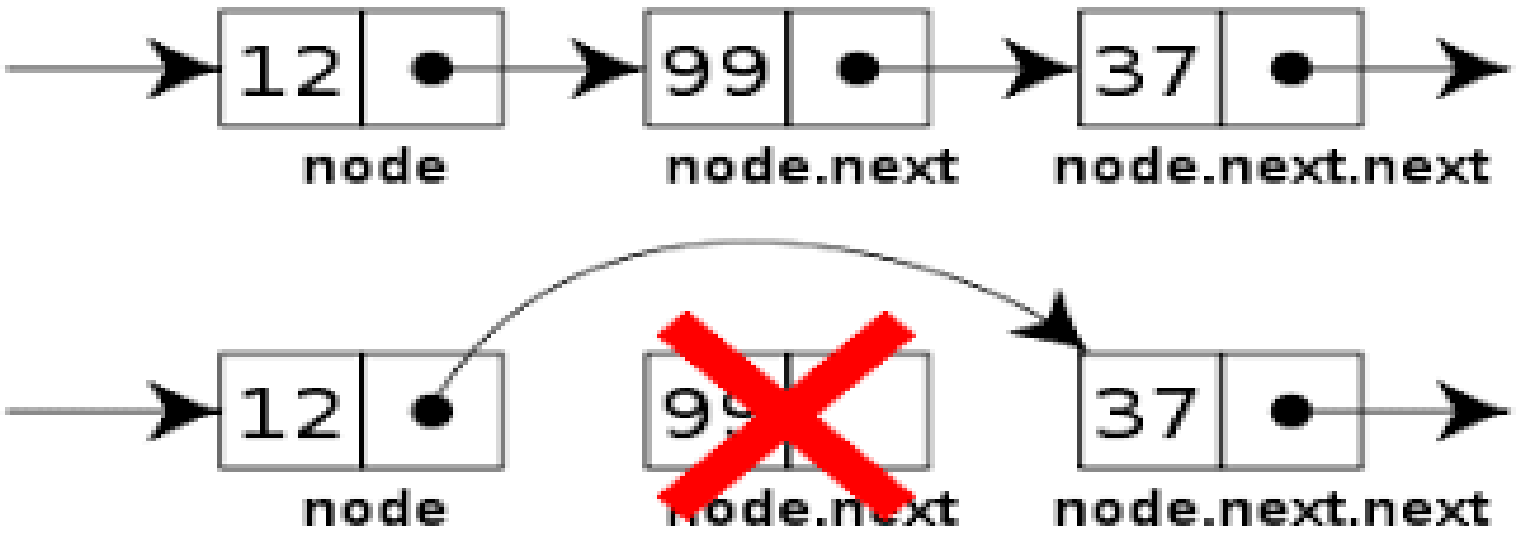
Step 8: If list contains multiple nodes, then check whether **temp1** is the first node in the list (**temp1 == head**).

Step 9: If **temp1** is the first node then move the **head** to the next node (**head = head → next**) and delete **temp1**.

Step 10: If **temp1** is not first node then check whether it is last node in the list (**temp1 → next == NULL**).

Step 11: If **temp1** is last node then set **temp2 → next** = **NULL** and delete **temp1** (**free(temp1)**).

Step 12: If **temp1** is not first node and not last node then set **temp2 → next** = **temp1 → next** and delete **temp1** (**free(temp1)**).


12
99
37
node
node.next
node.next.next
12
37
node
node.next.next