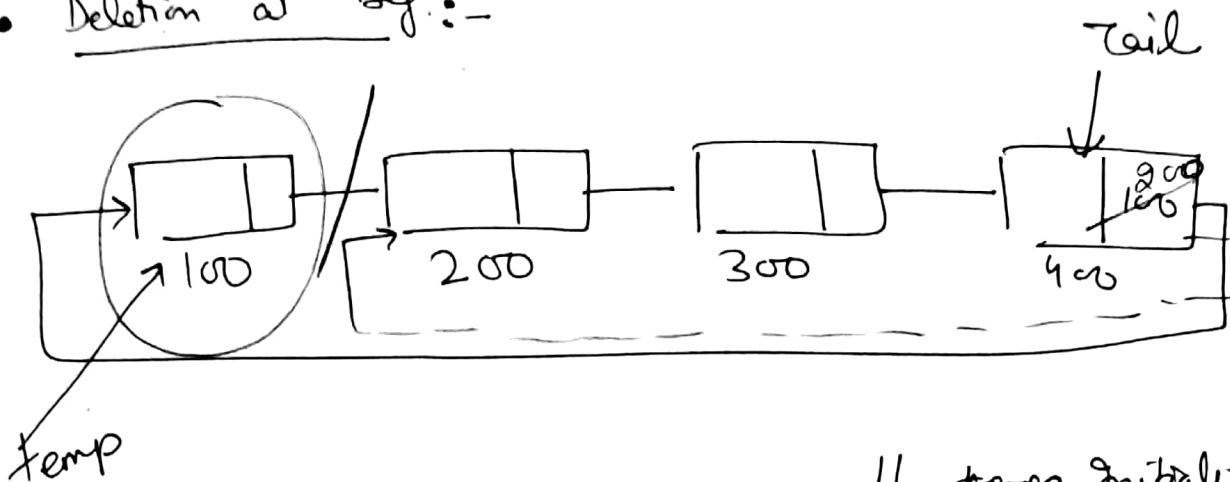


Lecture :- 20

Deletion in Circular Linked List :-

• Deletion at Beg. :-



$temp = tail \rightarrow next$ // ~~temp~~ initialize a pointer.

It will start pointing at first node if list exist.

if ($tail == 0$)

{
list is empty
nothing to delete
}

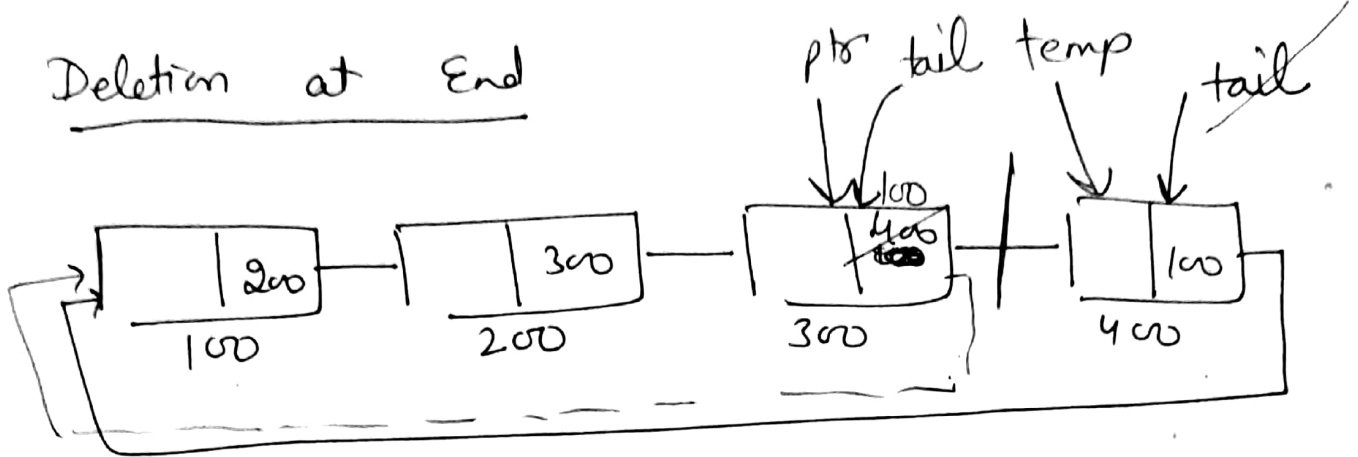
Else if ($temp \rightarrow next == temp$)

{
 $tail = 0$; // Only one node in the list & that we have to delete.
 $free(temp)$;
}

Else

{
 $tail \rightarrow next = temp \rightarrow next$;
 $free(temp)$;
}

Deletion at End



$temp = tail \rightarrow next$ // temp now points at first node

- Check one special case is no list
- or list have only one node.
- same as we did in deletion at Beg.

while ($temp \rightarrow next \neq tail \rightarrow next$)

{

$ptr = temp$; // initialise a pointer

$temp = temp \rightarrow next$; // update temp.

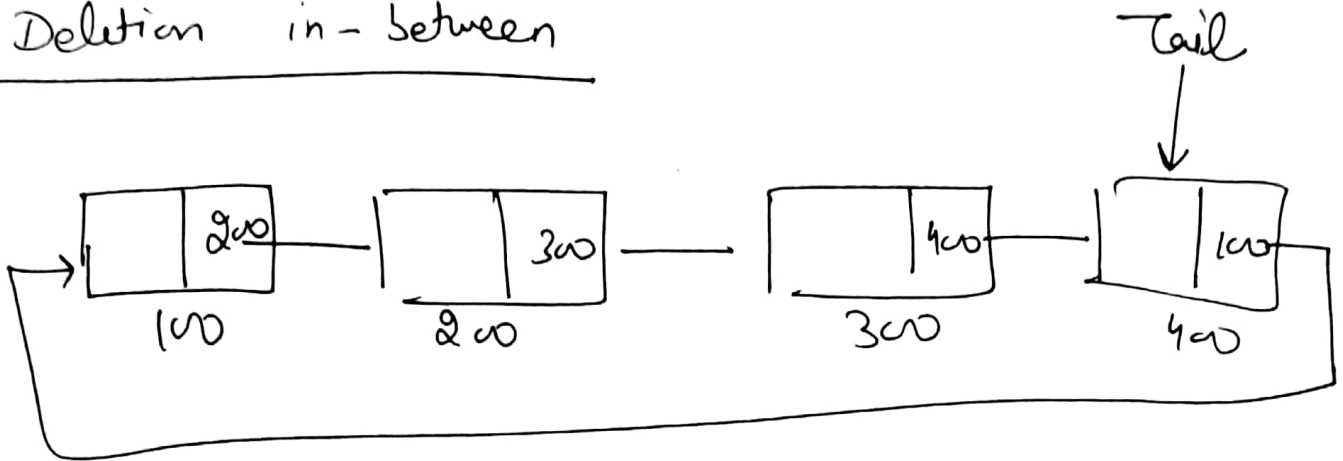
}

$ptr \rightarrow next = temp \rightarrow next$; // when temp reach at last node.

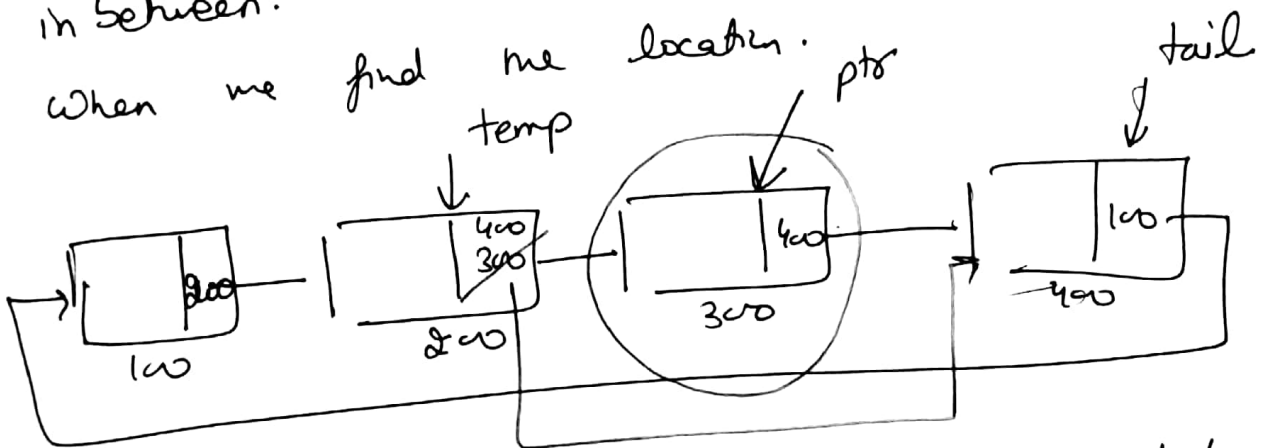
$tail = ptr$;

$free(temp)$;

Deletion in-between



- Read the position you want to delete
- Use the same logic as we did in insertion at in-between.
- When we find the location.



- the node we want to delete is pointed by ptr

$$\begin{cases} \text{temp} \rightarrow \text{next} = \text{ptr} \rightarrow \text{next} \\ \text{free}(\text{ptr}) \end{cases}$$