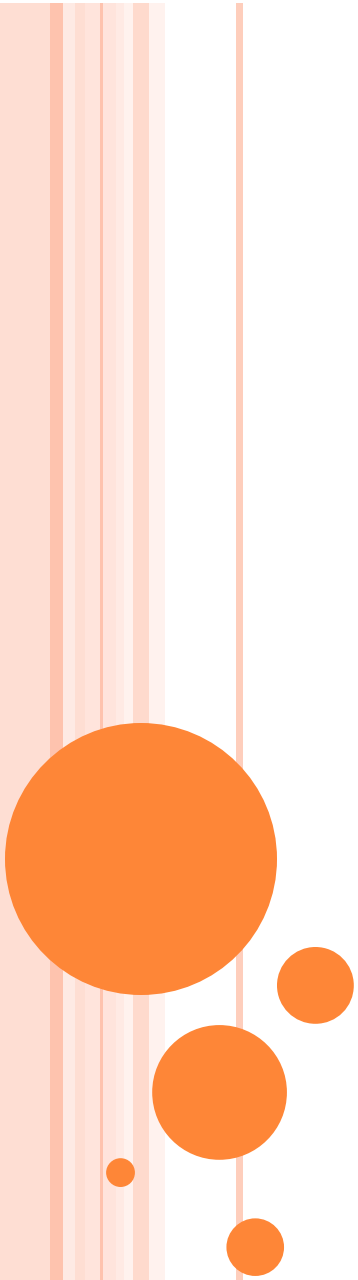


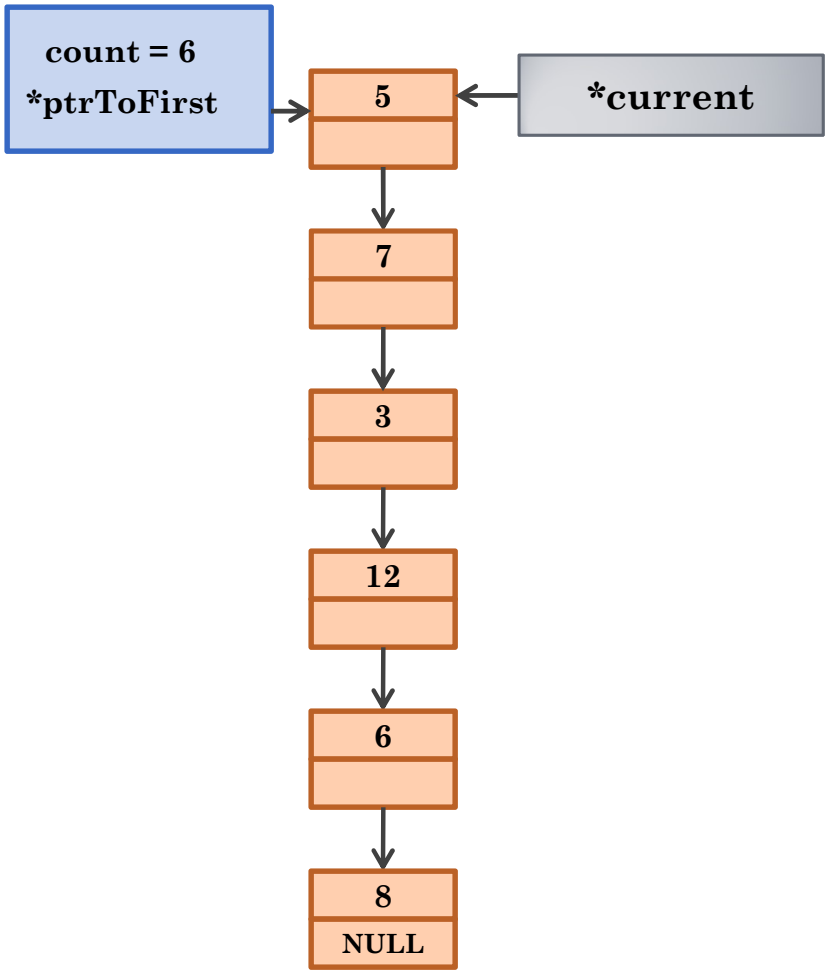
Singly Linked Lists (Part B)

CS A250 – C++ Programming II

Deleting a Node

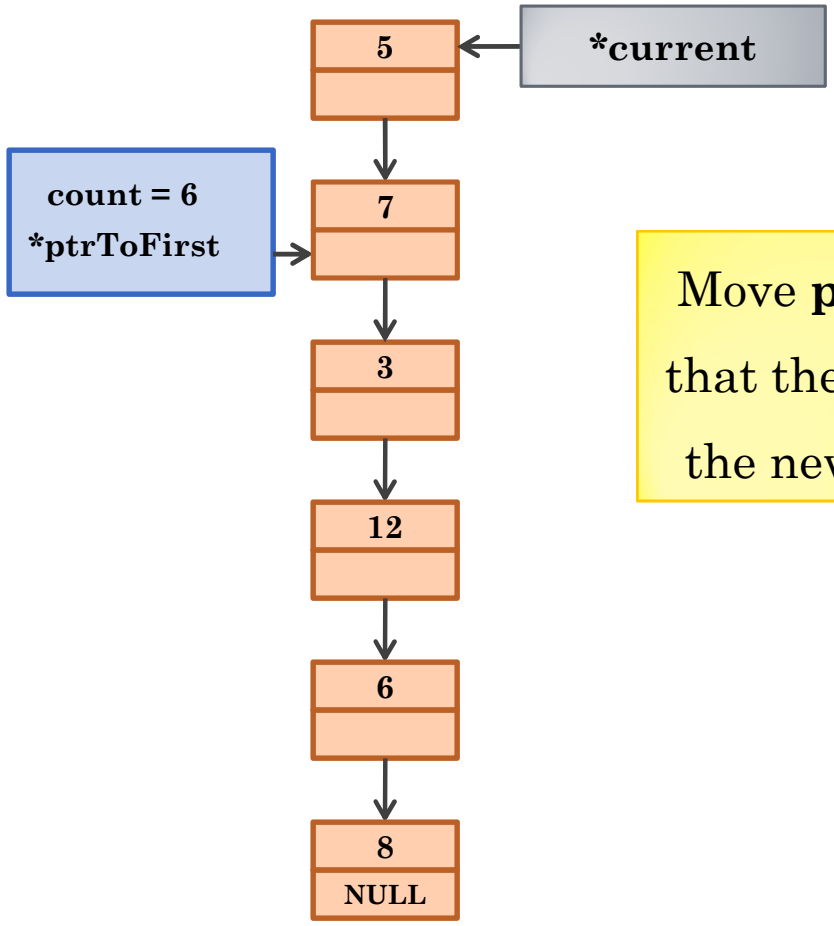
- To **delete a node** in the list, you need to first **find** the node and then delete it.
- Cases to consider:
  1. The **list is empty**
     - Output a **cerr** message
  2. The node to delete is the **first node**
     - You will need to connect the **list object** to the **second** node in the list, which will become the new first node in the list.
  3. The node to delete is **after the first node**
     - You will need 2 pointers, one that moves forward and another that follows.
  4. The node is **not found**

Deleting the First Node



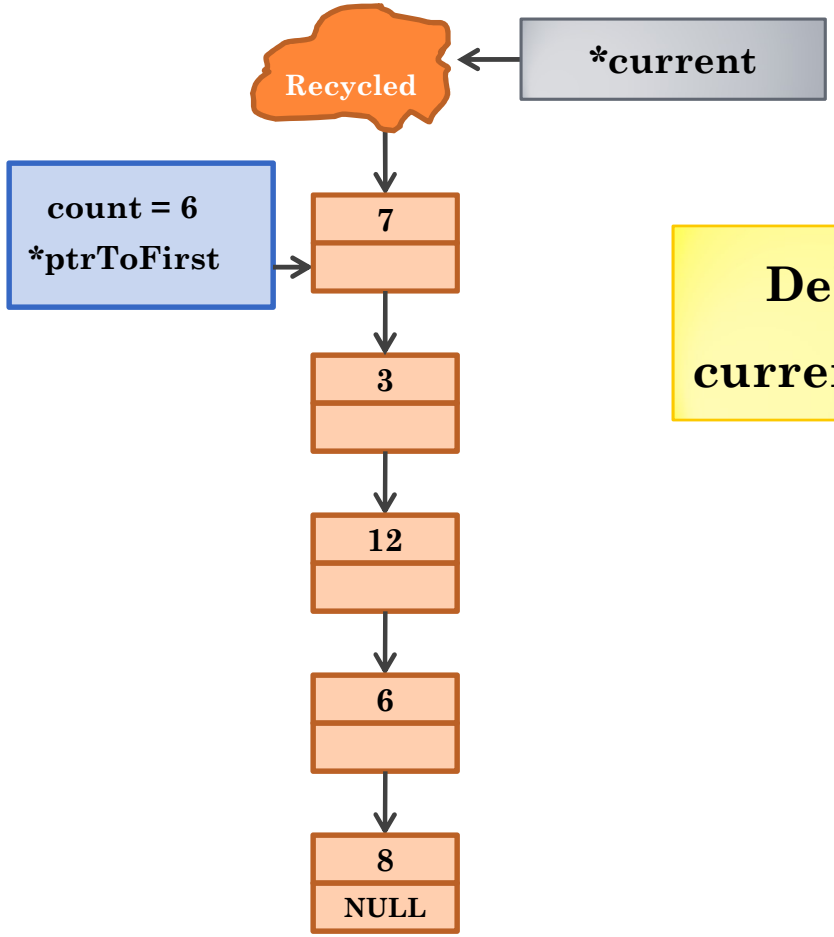
Looking for (5)

Deleting the First Node (cont.)



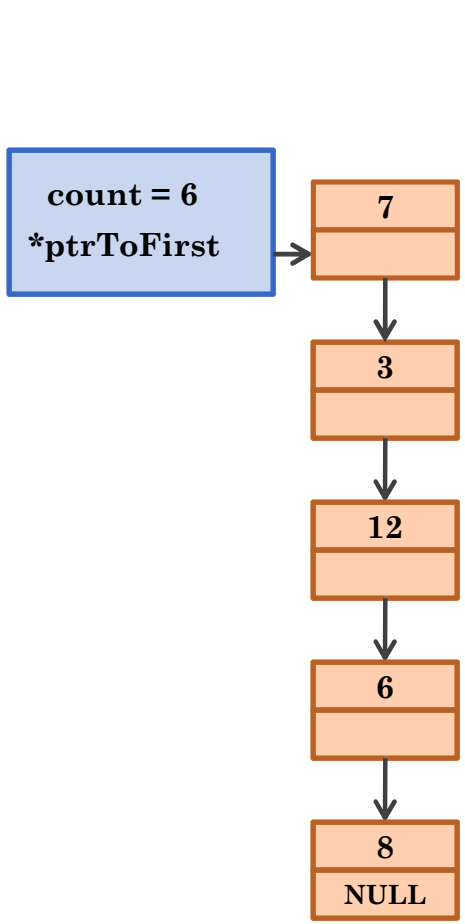
Move **ptrToFirst** forward, so that the **second** node becomes the new first node in the list.

Deleting the First Node (cont.)



Delete the node **current** is pointing to.

Deleting the First Node (cont.)

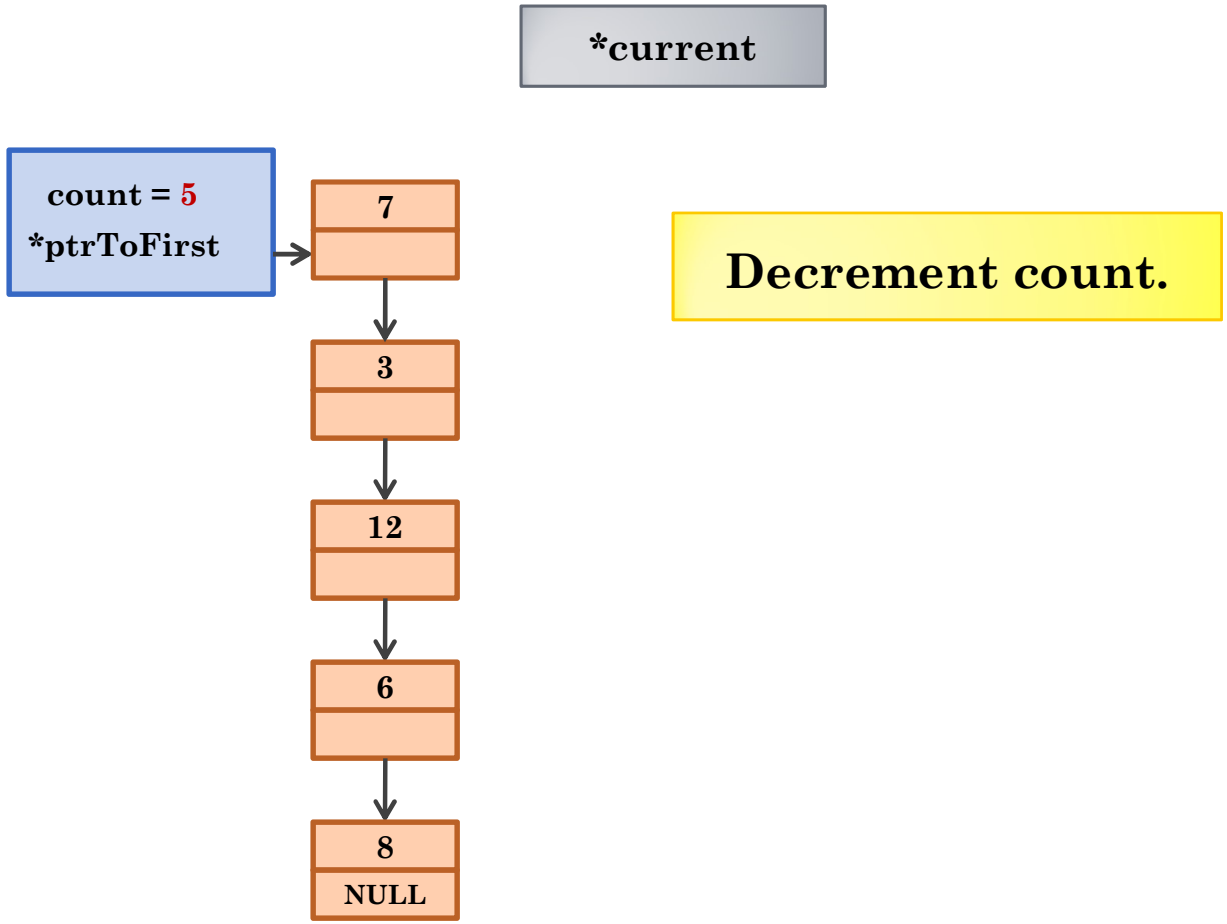


Reset **current** to **NULL,**

so that it is not a

dangling pointer.

Deleting the First Node (cont.)

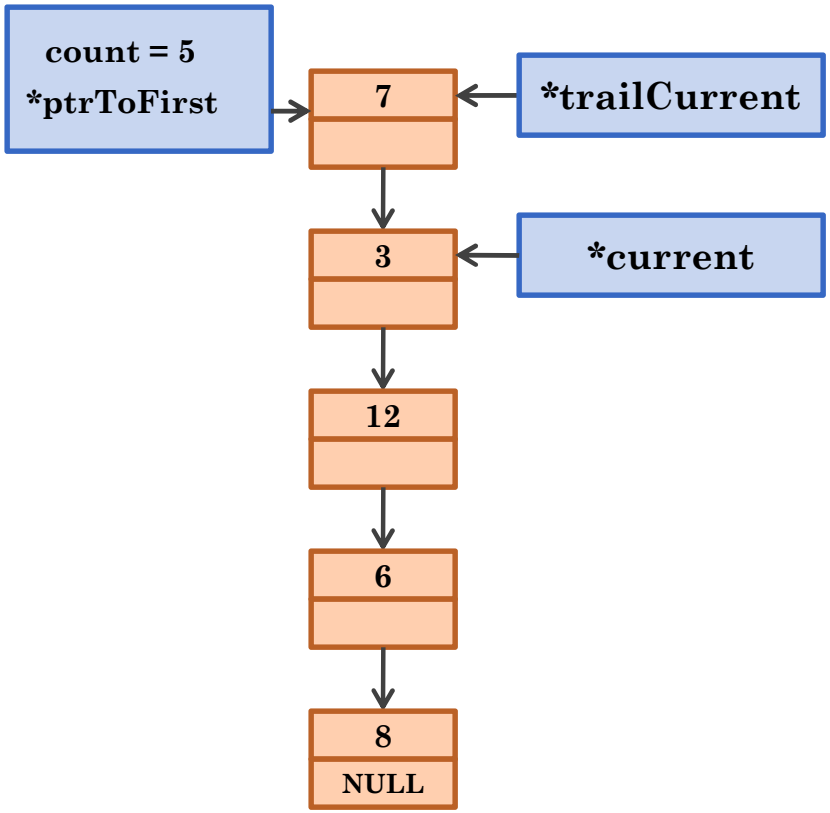


Deleting Other Nodes

- If the node you are looking for is **not** the first, you will need to use **2 pointers**.

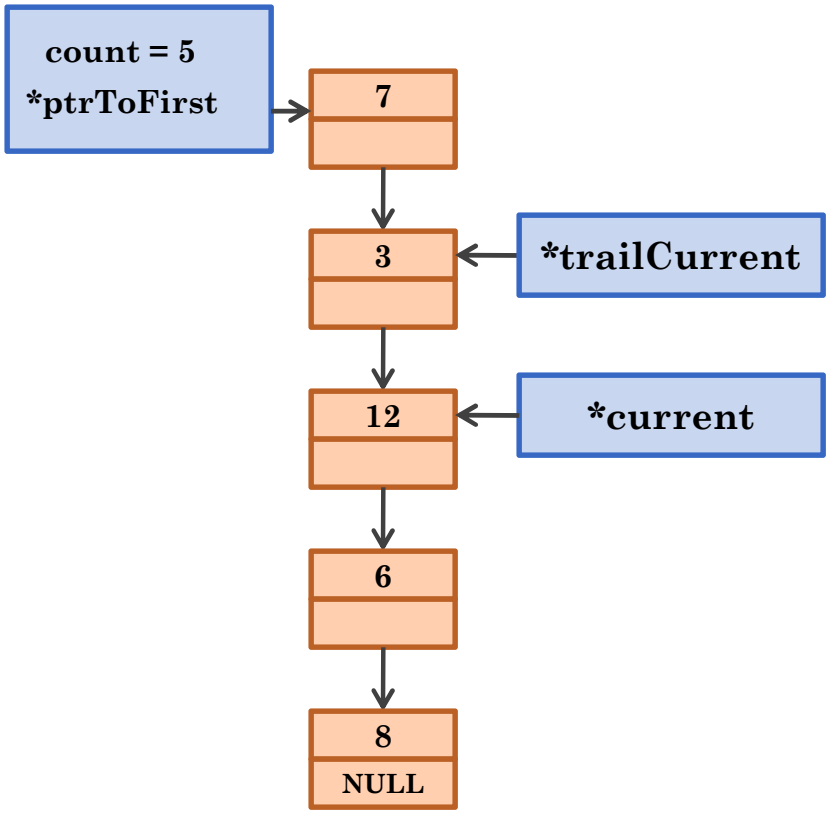
- The standard identifiers for these pointers are:
  - **current**
    - You can point this pointer to the second node in the list, because you already determined that the node to delete is not the first.
  - **trailCurrent**
    - You will be pointing this pointer to the first node in the list and always keep it behind **current**.

Deleting a Node (cont.)



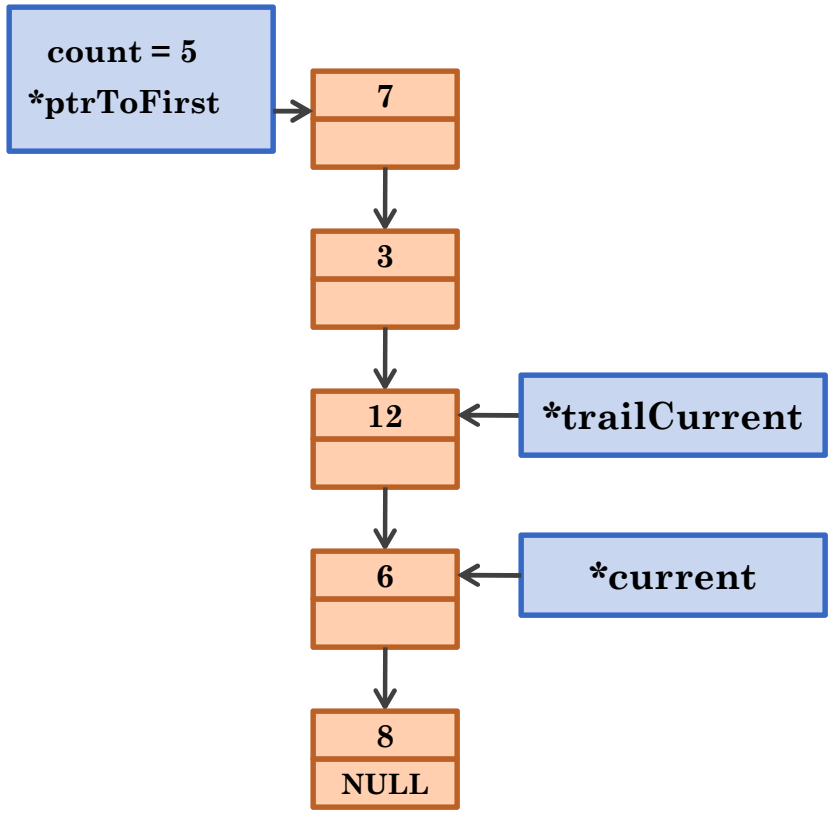
Looking for (6)

Deleting a Node (cont.)



Looking for (6)

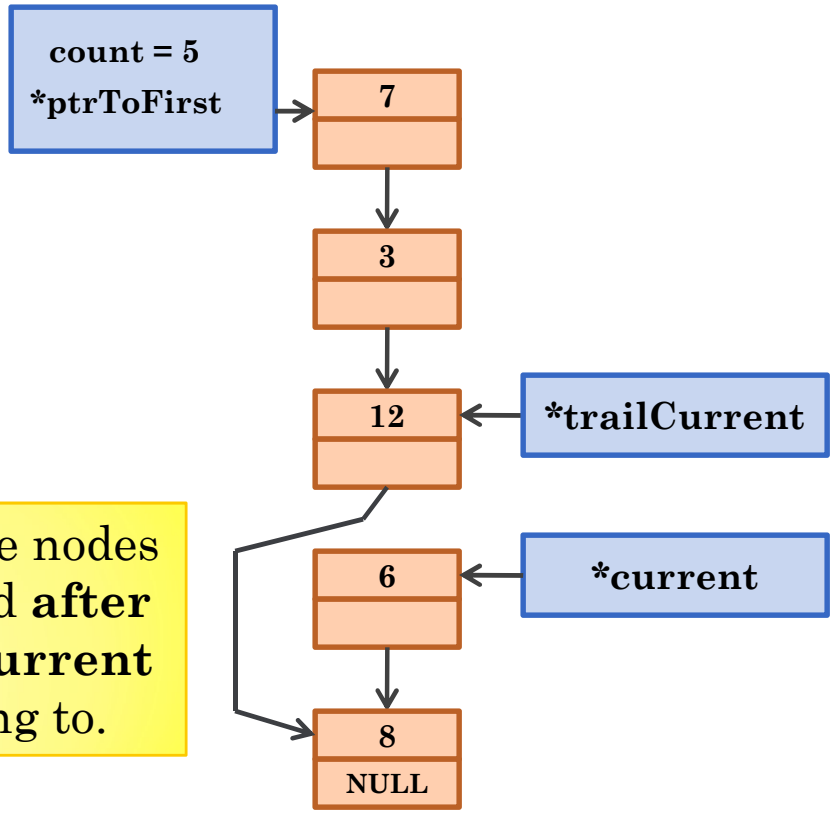
Deleting a Node (cont.)



Looking for (6)

Found!

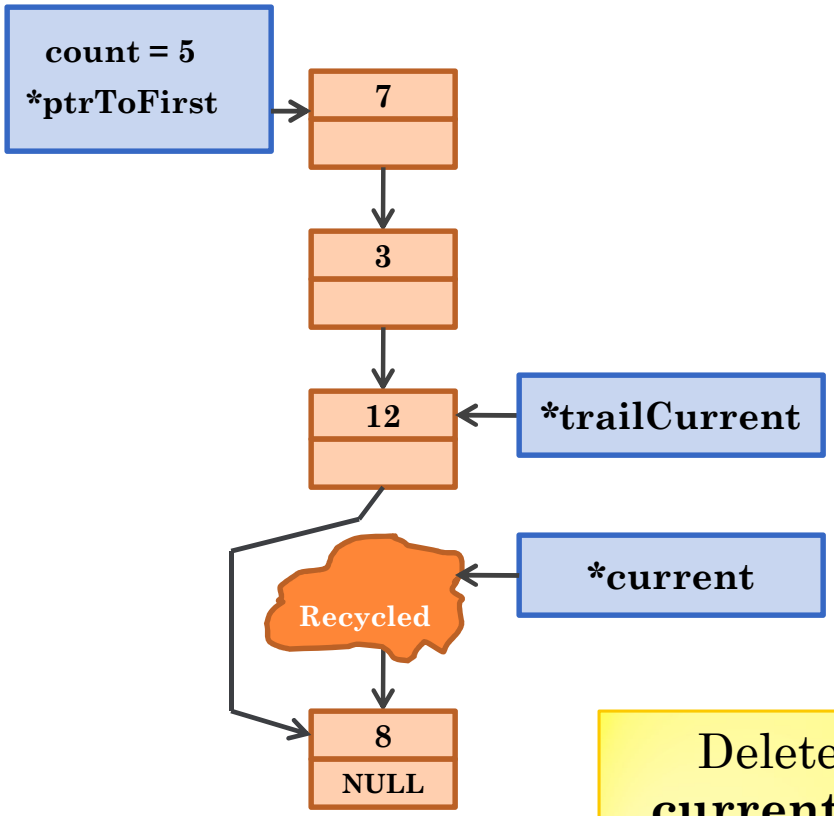
Deleting a Node (cont.)



Looking for (6)

Connect the nodes **before** and **after** the node **current** is pointing to.

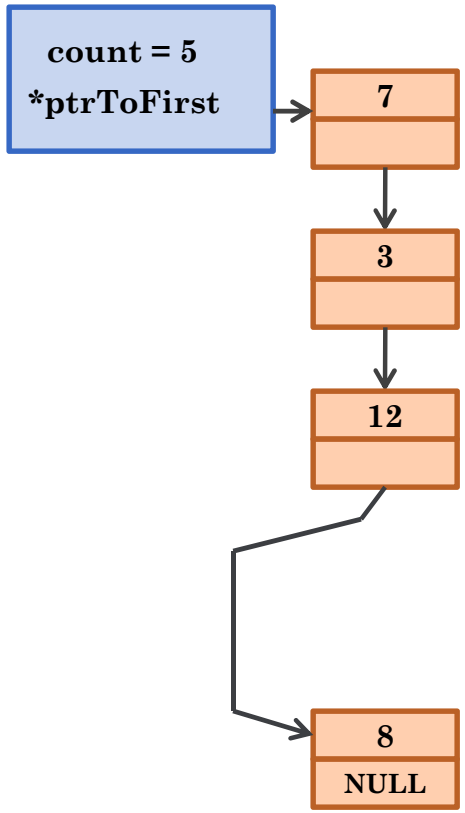
Deleting a Node (cont.)



Looking for (6)

Delete the node **current** is pointing to.

Deleting a Node (cont.)



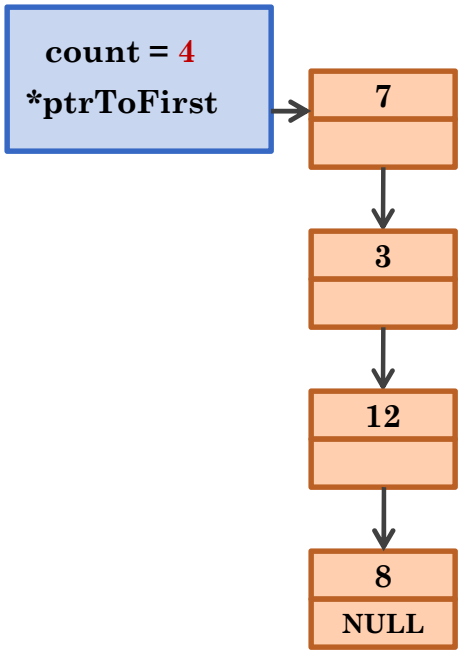
Looking for (6)

*trailCurrent

*current

Reset both **current** and **trailCurrent** to **NULL.**

Deleting a Node (cont.)



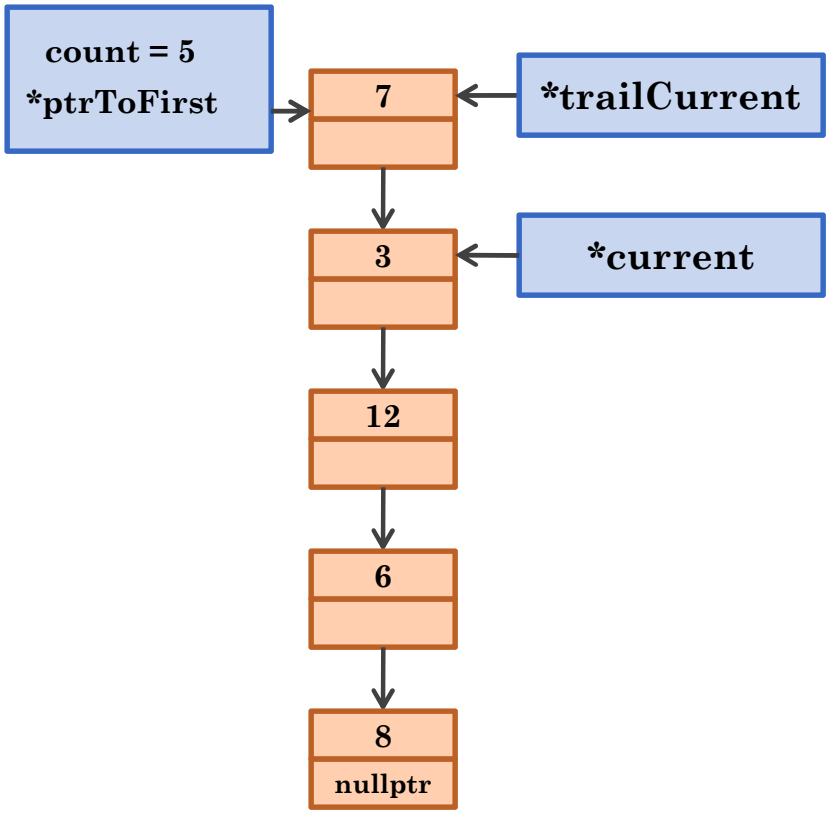
Decrement the **count**.

Deletion completed.

Deleting the Last Node in the List

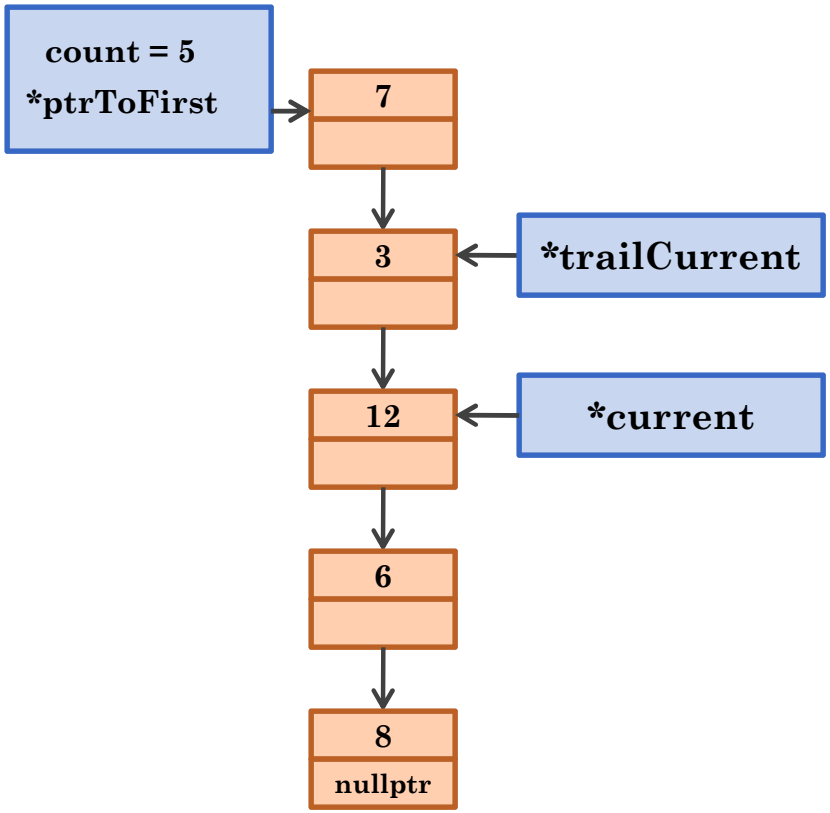
- If the node you are looking for is the **last node** in the list, you need to make sure that the **new last node** will point to **NULL.**

Deleting a Node (cont.)



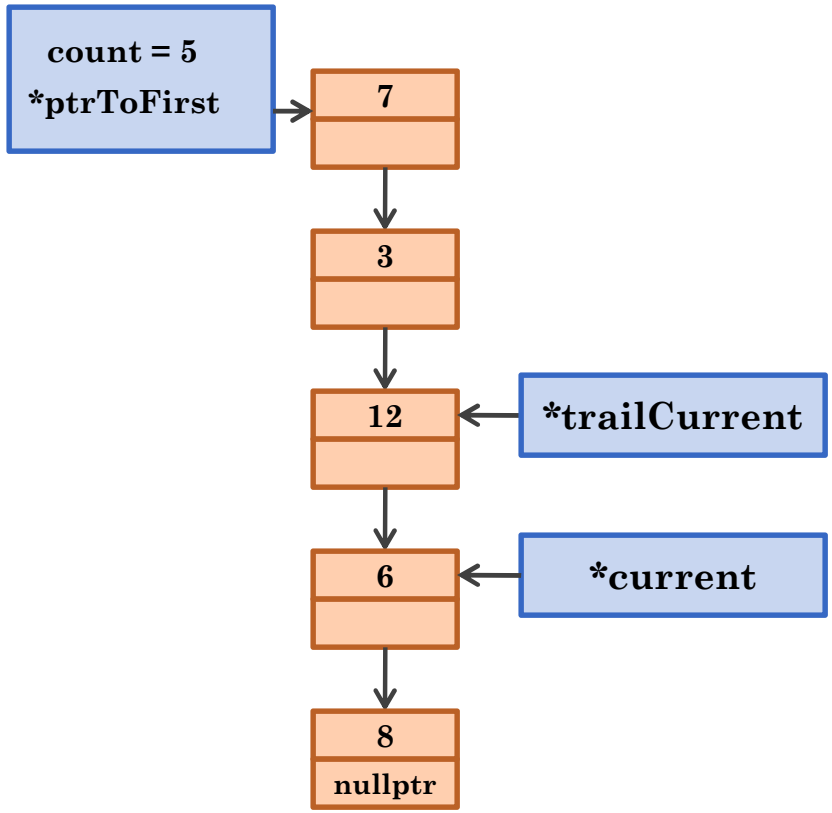
Looking for (8)

Deleting a Node (cont.)



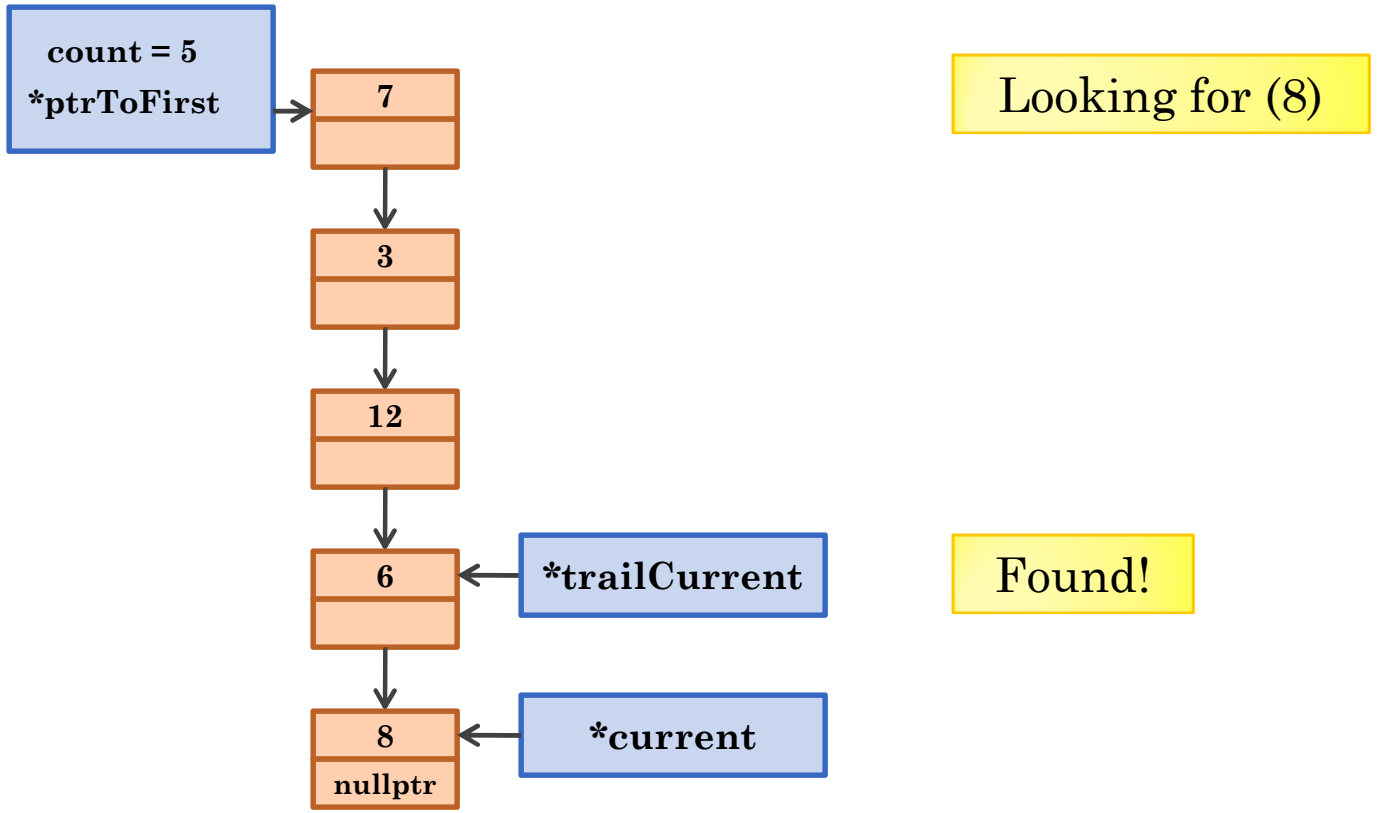
Looking for (8)

Deleting a Node (cont.)

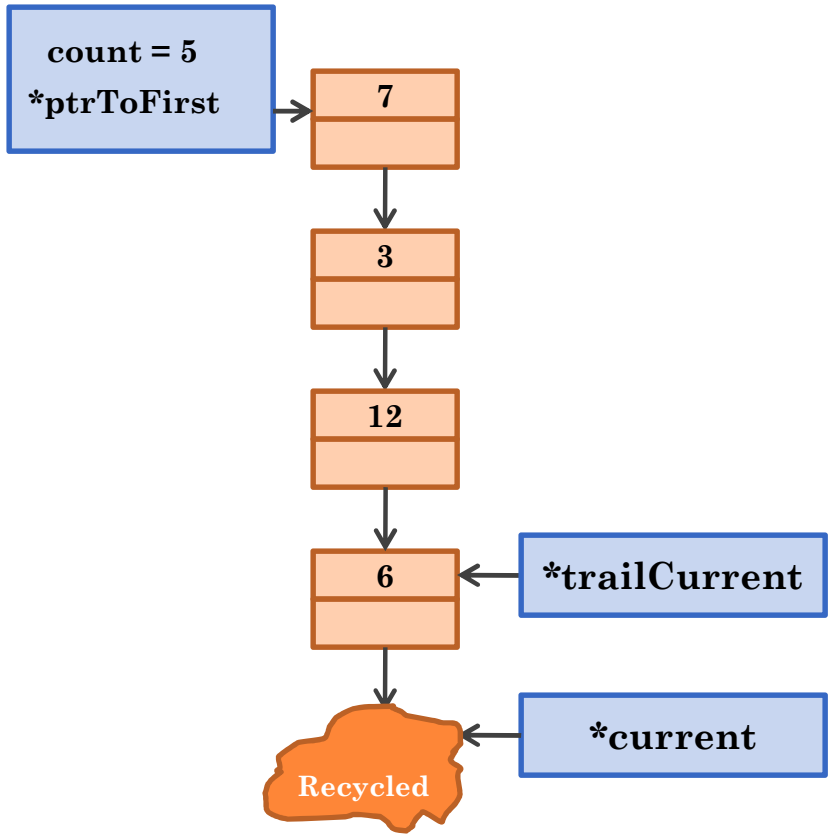


Looking for (8)

Deleting a Node (cont.)



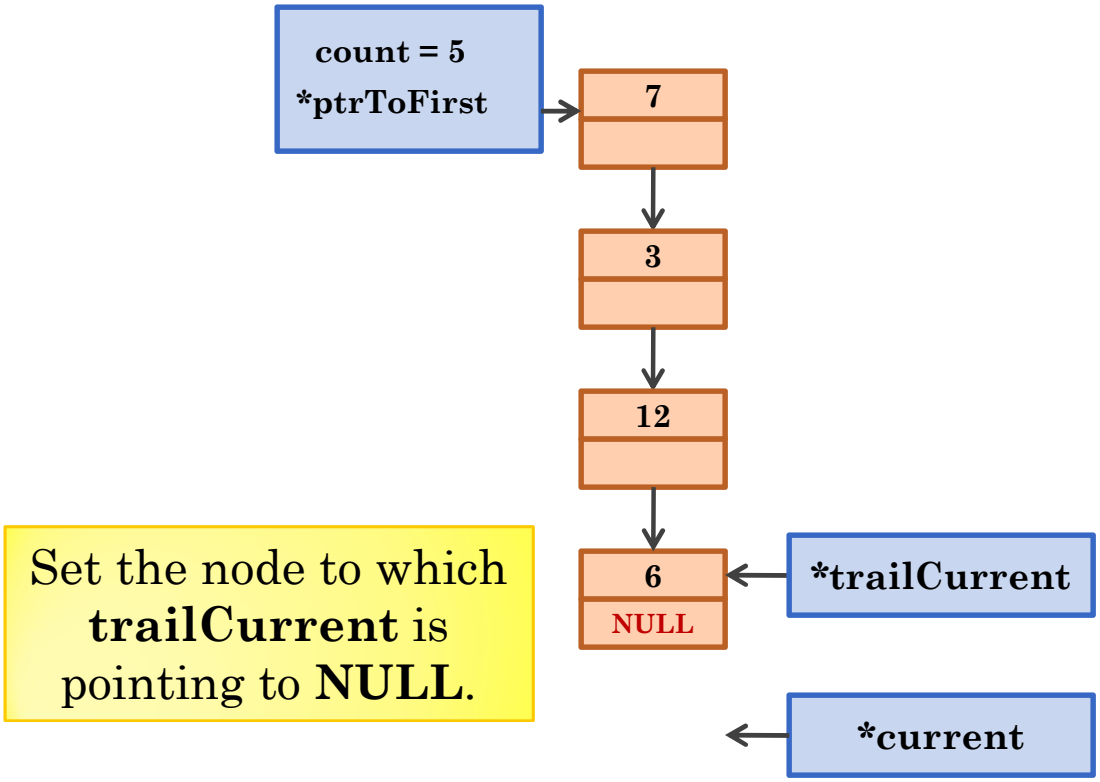
Deleting a Node (cont.)



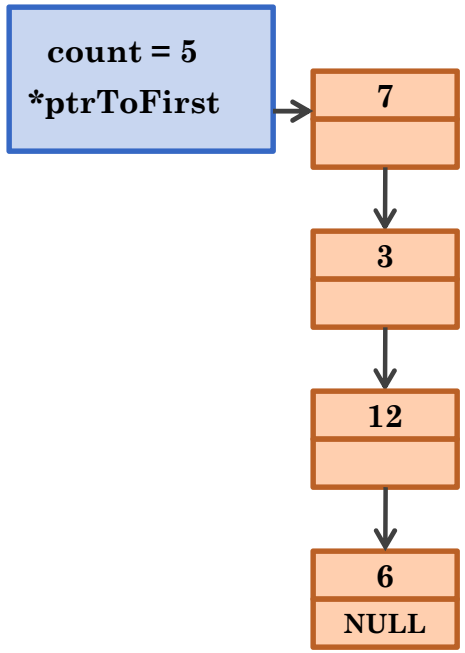
Looking for (8)

Delete the node **current** is pointing to.

Deleting a Node (cont.)



Deleting a Node (cont.)



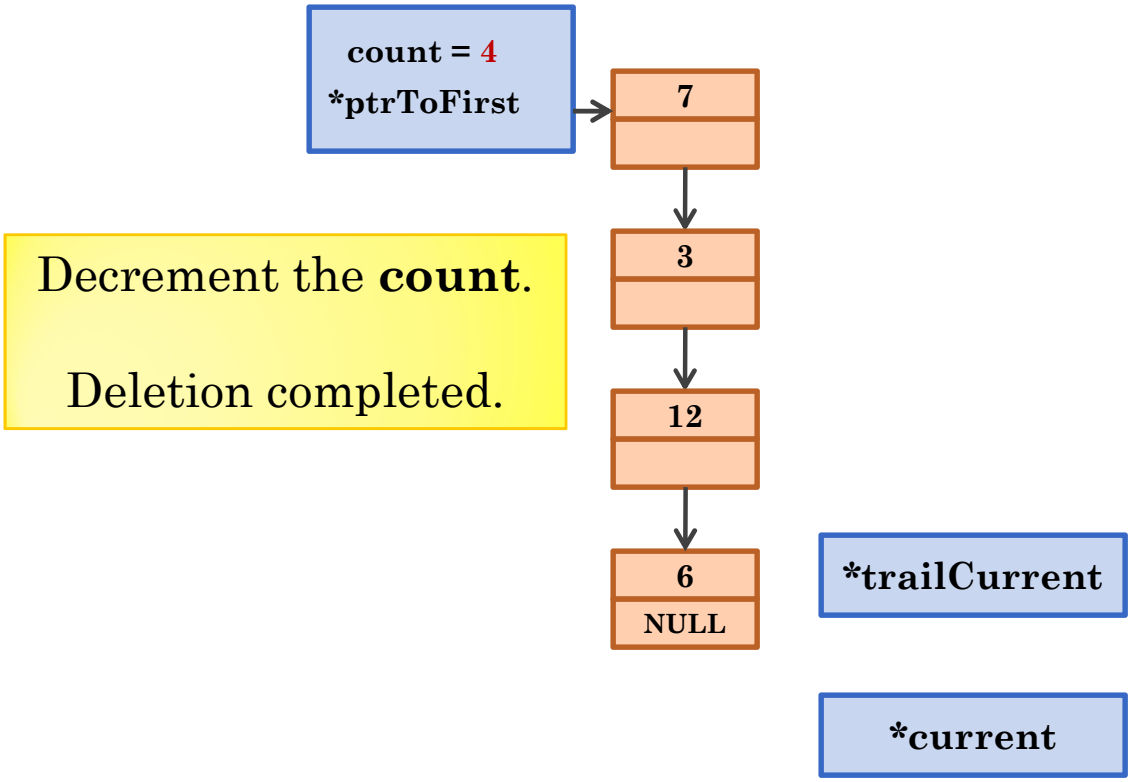
Looking for (8)

*trailCurrent

*current

Re-set to **NULL** both **current** and **trailCurrent**.

Deleting a Node (cont.)



End Singly-linked Lists (Part B)