



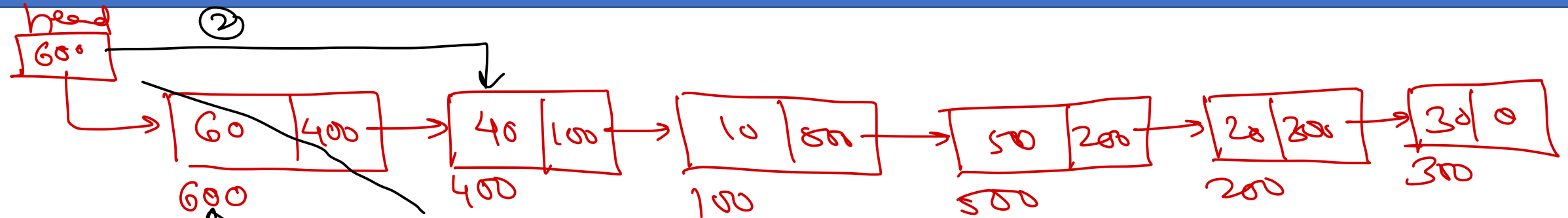
# Data Structure & Algorithms

Sunbeam Infotech



# Linked List

del\_first()



①  $\boxed{600}$   
temp

if (head != NULL)

{ temp = head;

head = head->next;

delete temp;

}

① take first node address into temp.

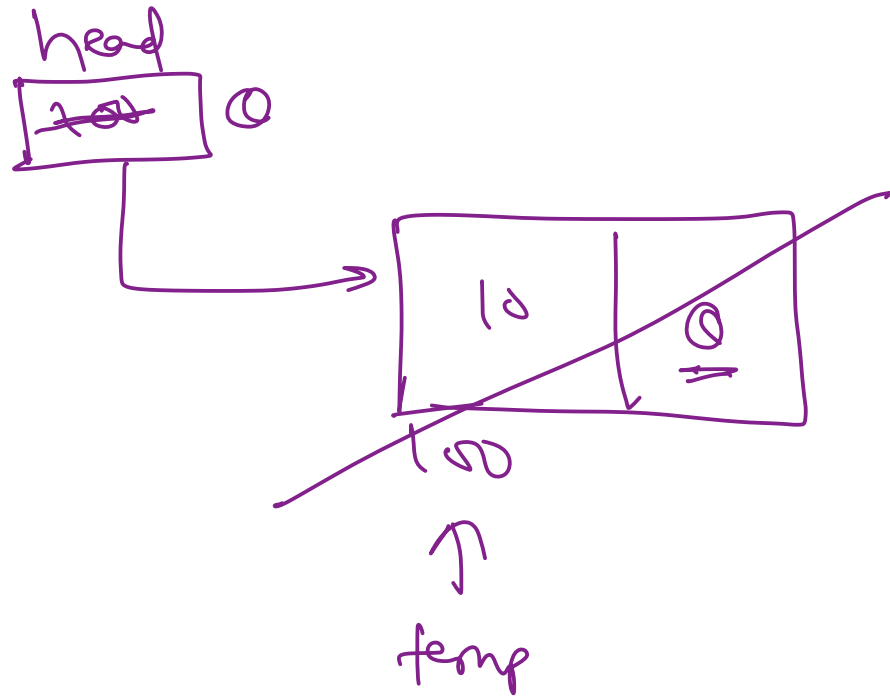
② take head to the next node.

③ delete temp.

head  
 $\boxed{0}$

# Linked List

if list has single node?

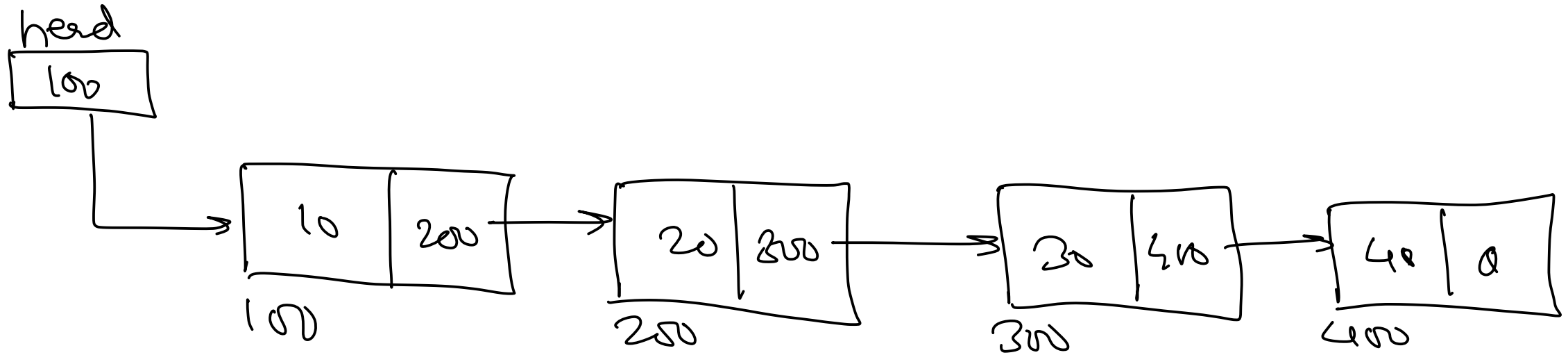


```
temp = head;  
head = head->next;  
delete temp;
```



# Linked List

del\_all()

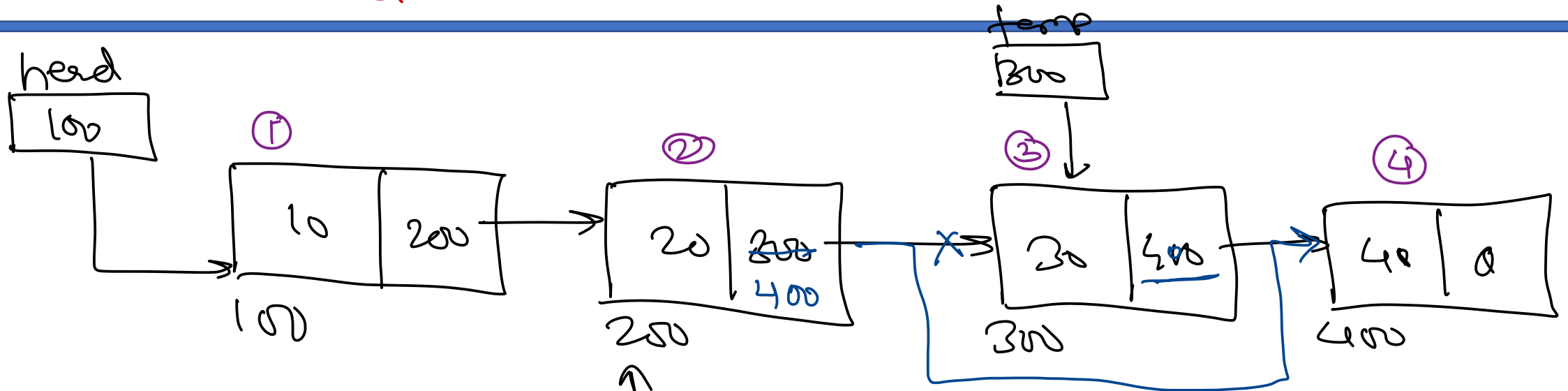


```
while (head != NULL)
    del_first();
```



# Linked List

del\_at\_pos()



```
temp = head;  
for (i = 1; i < pos - 1; i++)  
    temp = temp->next;
```

```
temp = temp->next;
```

```
temp->next = temp->next->next;
```

```
delete temp;
```

- ① traverse till pos - 1  
(temp)
- ② get ptr to the next node  
(node to be deleted).
- ③ temp next should point to  
temp next;
- ④ delete temp node



# Linked List

head



10 → 20 → 30 → 40



6



# Linked List

① Implement stack using linkedlist.

Answer:- Stack is LIFO, add/del done from same end.

```
class Stack {  
    list l;  
public:  
    push(v) {  
        add_at(v);  
    }  
    pop() {  
        del_at(v);  
    }  
    empty() {  
        head == NULL;  
    }  
}
```

```
peek() {  
    return head->data;  
}  
};
```



# Linked List

Q Implement Queue using Linked list.

Answer: Queue is FIFO. Add/Del done from diff ends.

push  $\rightarrow$  add\_last()

pop  $\rightarrow$  del\_front()







Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

