

## Queue Data Structure:

38

→ Unlike stacks, where we followed LIFO (Last in First Out) discipline, here in the queue, we have FIFO (First in First Out).

→ In stack, we had to maintain just one end, head, where both insertion and deletion used to take place, and the other end was closed. But here, in queues, we have to maintain both the ends because we have insertion at one end and deletion from the other end.

## → Queue ADT

### DATA:

In order to create a queue, we need two pointers, one pointing to the insertion end, to gain knowledge about the address where the new element will be inserted to. And the other pointer pointing to the deletion end, which holds the address of the element which will be deleted first. Along with that, we need the storage to hold the element itself.

### METHODS:

1. ENQUEUE() : to insert an element in a queue.
2. DEQUEUE() : to remove an element from the queue.
3. FIRSTVAL() : to return the value which is at the first position.
4. LASTVAL() : to return the value which is at the last position.
5. PEEK(POSITION) : to return the element at some specific position.
6. ISEmpty() / ISFULL() : to determine



Page No.	
Date	

Page No.			
Date			

whether the queue is empty or full, which helps us carry out efficient enqueue and dequeue operations.