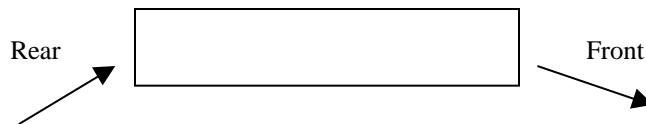


# Queues (FIFO)

Feb 11

A FIFO Queue is a linear data structure in which new elements are added in one end, called the rear of the queue, and removed from the other end, the front of the queue.

It is a First In First Out structure (FIFO).



Uses of queues:

- Line ups
- Simulation
- Modeling
- Program execution on a server.
- Inside an O/S

## Queue Specifications

queue() - constructor, initializes queue to empty state.

empty() - returns boolean  
- checks to see if the queue is empty

insert(Object x) - inserts x onto the end of the queue

remove() - returns object  
- if queue is empty, returns null  
- otherwise, takes an object from the front of the queue and returns it.

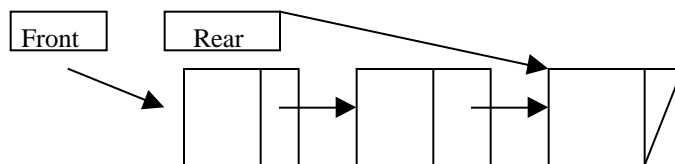
See using queue example ☺

## Queue Implementation

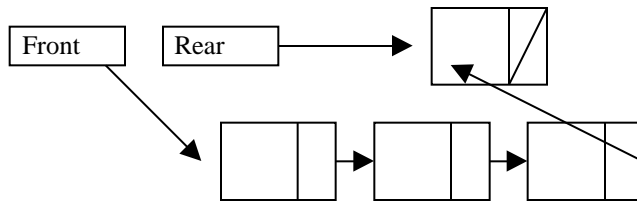
In this course we will talk about 2 implementations of queues:

- Linked list implementation
- Array implementation

### Linked List Implementation:



Adding a node:



Removing a node is similar, and is straightforward.

NOTE: Using the rear reference is handy because it reduces the order of complexity of adding a node to  $O(1)$ .

Special Cases:

When the queue is empty, adding an item to the list results in having to update BOTH front and rear references, setting them to the new item.

If there is only one entry in the list, and we remove it, we again have to update both the front and rear references, setting them to null.

[See the Queue implementation example in the text 😊](#)