A+ Computer Science

QUEUES

What is a queue?

What is a queue?

A queue is a group of items all of the same type where items are added to the back of the queue and removed from the front.

The first item added would be the first item removed. Queues work in a FIFO manner.

What is a queue?

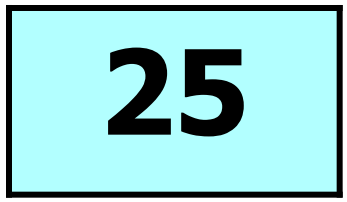
An empty integer queue.

```
Queue<Integer> queue;
queue = new LinkedList<Integer>();
```

queue will only store integer values.

What is a queue?

```
queue.add(25);
```



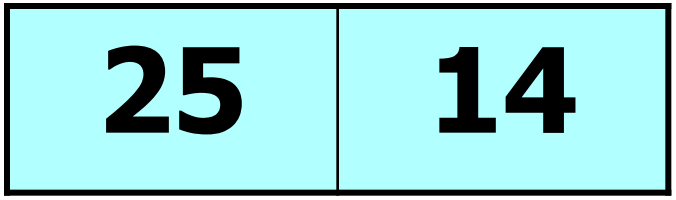
add adds an item to the queue.

enqueue is a very common name given to the operation of adding items to a queue.

What is a queue?

queue.add(14);

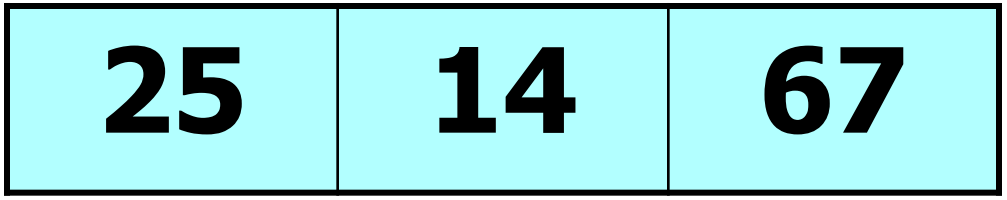
add adds an item to the queue.



What is a queue?

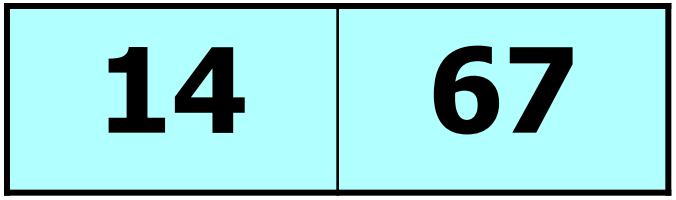
queue.add(67);

add adds an item to the queue.



What is a queue?

queue.remove();



remove removes an item from the queue.

dequeue is a very common name given to the operation of removing items from a queue.

What is a queue?

remove removes an item from the queue.

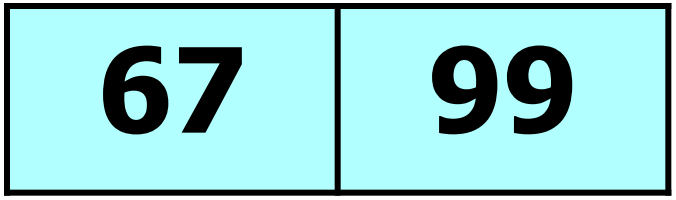
```
queue.remove();
```

67

What is a queue?

queue.add(99);

add adds an item to the queue.



Queue Interface

The Queue interface was designed to allow the use of a queue in java.

The LinkedList class implements the Queue interface.

If you need a queue, just make a Queue reference to a LinkedList.

Queue methods

Linked List as a Queue
frequently used methods

Name	Use
add(x)	adds item x to the queue
remove()	removes and returns front item
peek()	returns the front item with no remove
size()	returns the # of items in the queue
isEmpty()	checks to see if the queue is empty

```
import  java.util.Queue;
```

add() method

```
Queue<Integer> queue;
queue = new LinkedList<Integer>();

queue.add(11);
queue.add(10);
queue.add(7);
out.println(queue);
```

OUTPUT

[11, 10, 7]

remove() method

```
Queue<Integer> queue;
queue = new LinkedList<Integer>();

queue.add(11);
queue.add(10);
queue.add(7);
out.println(queue.remove());
out.println(queue);
```

OUTPUT

```
11
[10, 7]
```

queueadd.java
queueremove.add

peek() method

```
Queue<Integer> queue;
queue = new LinkedList<Integer>();
queue.add(11);
queue.add(7);
out.println(queue);
out.println(queue.peek());
queue.remove();
out.println(queue.peek());
queue.remove();
out.println(queue);
```

OUTPUT

```
[11, 7]
11
7
[]
```

queuepeek.java

isEmpty() method

```
Queue<Integer> queue;
queue = new LinkedList<Integer>();
queue.add(11);
queue.add(10);
queue.add(7);

while(!queue.isEmpty())
{
  out.println(queue.remove());
}
```

OUTPUT

```
11
10
7
```

queueisempty.java

PriorityQueue

PriorityQueue

A PriorityQueue is a queue structure that organizes the data inside by the natural ordering or by some specified criteria.

The Java PriorityQueue is a min heap as it removes the smallest items first.

The Java PriorityQueue stores Comparables.

PriorityQueue methods

PriorityQueue

frequently used methods

Name	Use
add(x)	adds item x to the pQueue
remove()	removes and returns min priority item
peek()	returns the min item with no remove
size()	returns the # of items in the pQueue
isEmpty()	checks to see if the pQueue is empty

add() method

```
PriorityQueue<Integer> pQueue;
pQueue = new PriorityQueue<Integer>();

pQueue.add(11);
pQueue.add(10);
pQueue.add(7);
out.println(pQueue);
```

OUTPUT

[7, 11, 10]

remove() method

```
PriorityQueue<Integer> pQueue;
pQueue = new PriorityQueue<Integer>();

pQueue.add(11);
pQueue.add(10);
pQueue.add(7);
out.println(pQueue);
out.println(pQueue.remove());
out.println(pQueue);
```

OUTPUT

```
[7, 11, 10]
7
[10, 11]
```

pqadd.java
pqremove.java

isEmpty() method

```
PriorityQueue<Integer> pQueue;
pQueue = new PriorityQueue<Integer>();

pQueue.add(11);
pQueue.add(10);
pQueue.add(7);

while(!pQueue.isEmpty())
{
  out.println(pQueue.remove());
}
```

OUTPUT

7
10
11

pqueueisempty.java

Work on Programs!

Crank

Some Code!

A+ Computer Science

QUEUES