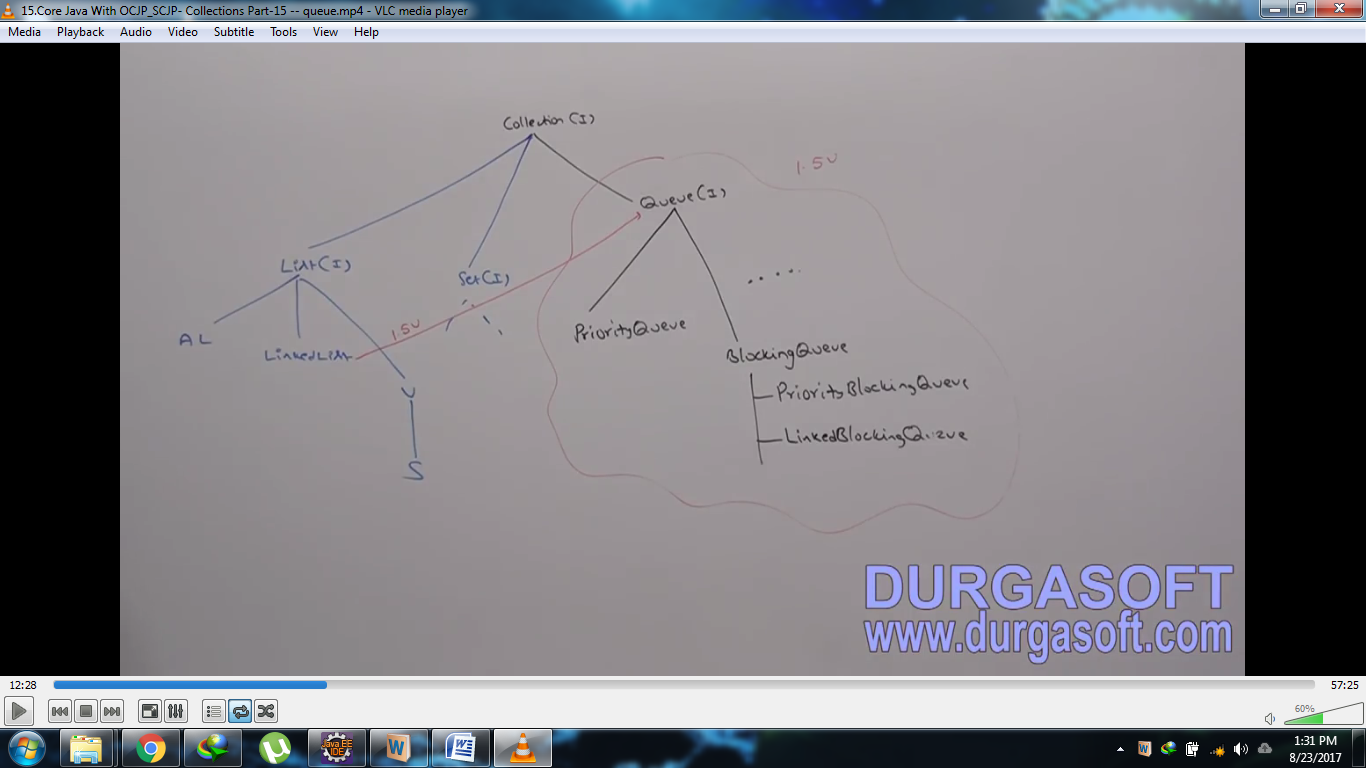
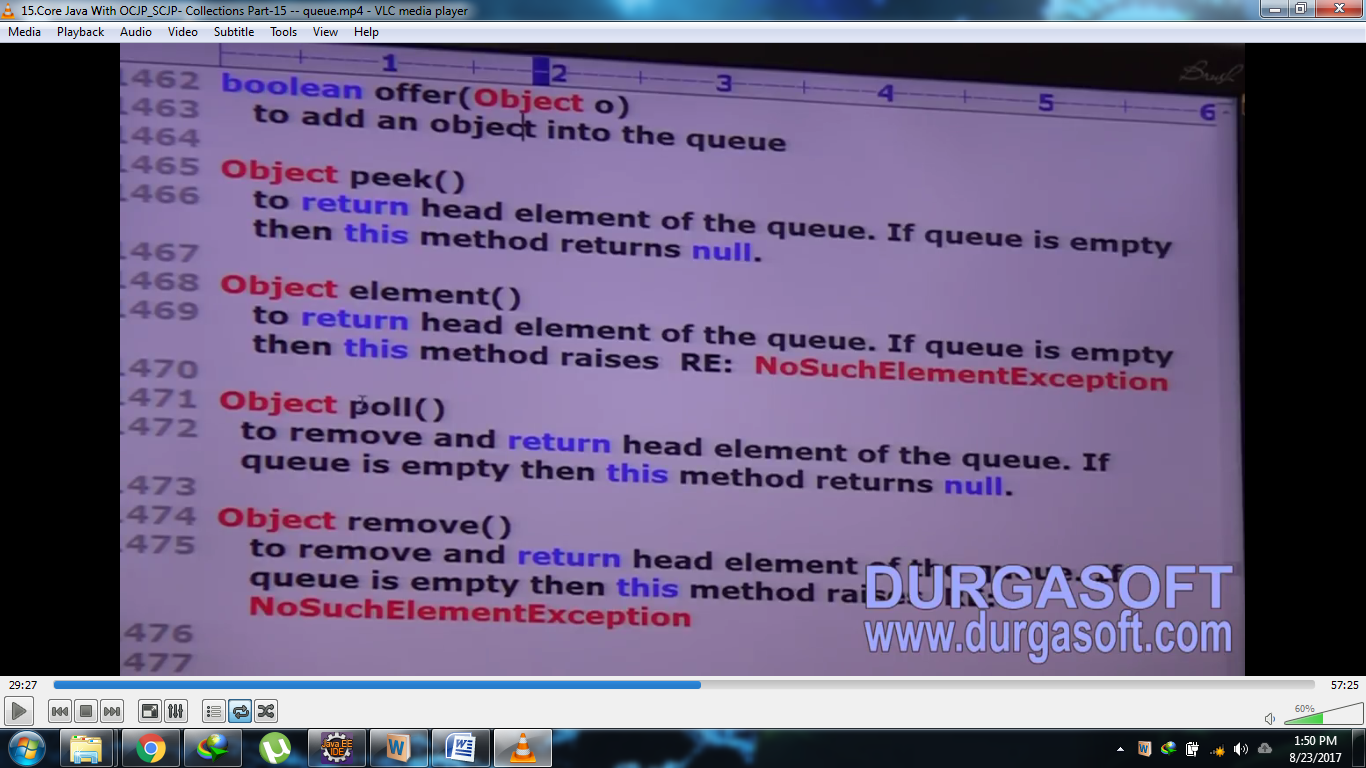
Queue

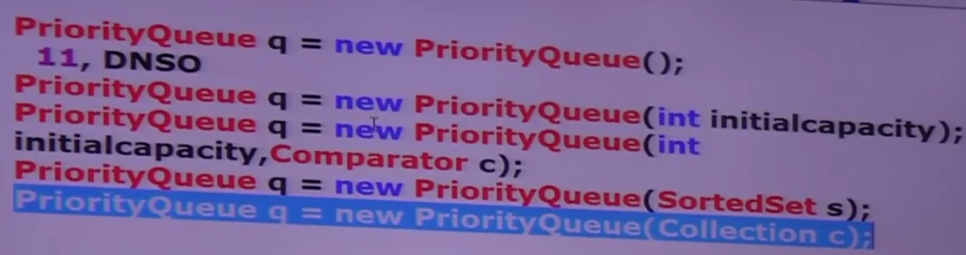
1. 1.5 version enchancement 🡪 Queue(I)
   1. Child interface of Collection.
   2. **Usage**: If we want to represent a group of individual objects **prior to processing.**

from 1.5, the LinkedList implements Queue(I) too.

* 1. **Example:** Before sending message to all mobile numbers, we have to store those numbers in some data structure. In which order, we add mobile numbers, in the same order only, message should be sent. For this, 1st in 1st requirement, Queue is the best choice.  
     

2. Usually Queue follows 1st in 1st out order but based on our requirement, we can implement our own priority order (priority queue).
3. From 1.5v onwards, LinkedList class also implements Queue(I)
4. **Queue interface specific methods**:
   1. **offer(Object):** as we get into the queue to get some service.
   2. **Object peek(): To return head element of the queue. If queue is empty then it returns null**  
      Object element(): same as peek() but if queue is empty throws **NoSuchElementException**
   3. **Object poll(): To remove head element and returns it but if queue is empty, then it returns null  
      Object remove(): same as poll() but if queue is empty, it throws NoSuchElementException**
5. d

PriorityQueue(c)

1. If we want to represent a group of objects prior to processing according to some priority, then we should go for priority queue.
2. The priority can be default natural sorting order or customized sorting order defined by comparator.
3. Insertion order is not preserved and it’s based on some priority.
4. Duplicate objects are not allowed.
5. If we are depending on default natural sorting order, then the objects should be homogeneous and comparable otherwise we will get Runtime exception 🡪 ClassCastException.
6. If we are defining our own sorting by comparator then objects need not be homogeneous and comparable.
7. null insertion is not allowed even as 1st element also.
8. **Constructors**
   1. 
9. Some platforms (Operating Systems) don’t provide proper support for thread priority and priority queue due to which the elements sorted in PriorityQueue may not be shown on sop(priorityQueueObj).
10. 