**Q) What are key points about SynchronousQueue?**

**Things to remember about SynchronousQueue in Java**

Here are some of the important properties of this special blocking queue in Java. It's very useful to transfer data from one thread to another thread synchronously. It doesn't have any capacity and blocks until there is a thread on the other end.  
  
1) SynchronousQueue blocks until another thread is ready to take the element, one thread is trying to put.  
  
2) SynchronousQueue has zero capacity.  
  
3) SynchronousQueue is used to implement queuing strategy of  direct hand-off, where thread hands-off to waiting thread, else creates new one if allowed, else task rejected.  
  
4) This queue does not permit null elements, adding null elements will result in [NullPointerException](http://javarevisited.blogspot.sg/2012/06/common-cause-of-javalangnullpointerexce.html).  
  
5) For purposes of other Collection methods (for example contains), a SynchronousQueue acts as an empty collection.  
  
6) You cannot peek at a synchronous queue because an element is only present when you try to remove it; similarly you cannot insert an element (using any method) unless another thread is trying to remove it.  
  
7) You cannot iterate over SynchronousQueue, as there is nothing to iterate

8) A SynchronousQueue constructed with fairness policy set to true grants threads access in FIFO order.

**Q) Which Implementation of Queue follows FIFO principle?**

**LinkedList** and **LinkeBlockingQueue** will read the elements from the FIFO order?

**Q). Difference between Normal Queue and BlockingQueue implementations?**

Noramal Queue classes will have **peek**(), **poll**(), **remove**(), **element**() methods.

BlockingQueues have additional methods **take**() and **put**() which blocks execution

**Q). When to use ArrayBlockingQueue**

It must have the capacity to be passed while creating this object. If the capacity is full it block the element to be added. And if is empty blocks until element is present.

Q).