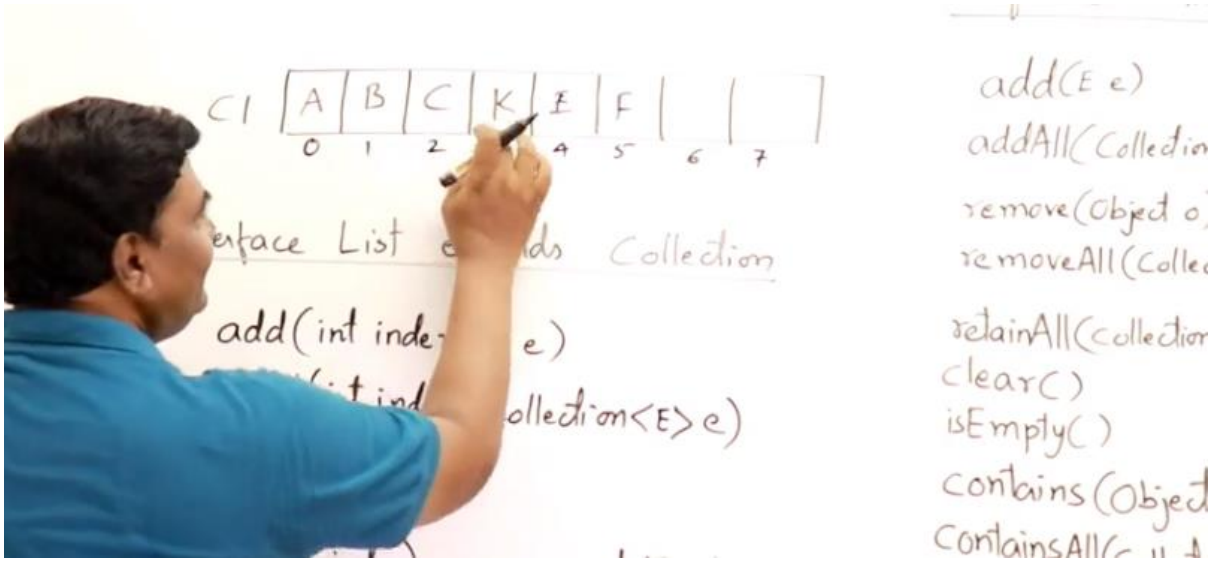
Interface List extends Collection

- add(int index, E e) – it will add an element at a given index.
- addAll(int index, Collection<E>e) – If I have to insert more than one values from another collection, starting at some given index.
- remove(int index) – remove an object from given index
- get(int index) – we can know the object of index
- set(int index, E e) – replace the object of a given index
- subList(int from, int to) – we can get the part of a list, i.e a collection of a few elements we can take.



Ex- I want the elements from 1-4 indices – C1 [B, C, K, E]

- indexOf(Object o)- index of a given object and it will search from first indices to last indices.
  Ex- If I say indexOf(C)- I will get 2
- lastIndexOf(Object o)- it will search from last to first.
- listIterator()- access the element either direction, we can move forward direction also we can move from backward direction. This is bidirectional.

- listIterator(int index)- we can start moving from some starting point.
  Ex- we want to start from index 3 onwards.

Interface Queue extends Collection

- add(E e)- It will not ask to where to add, it will simply add at the end because it support FIFO mechanism. A new object will be inserted at the end of the queue.
- poll()- this will remove the first object. As it's in FIFO ill will not ask which element to remove. It will remove first element only. [remember if queue is empty poll will return null]
- remove() throws NoSuchElement Exception- it does the same thing like poll() but if the queue is empty it will return exception.
- Peek() – If I just want to know the object or the element present in the beginning, before removing it from the queue. Just I want to see what's there inside the queue. If the queue is empty, it returns null.
- element() throws – It will give us the first element but here if queue is empty it will throw exception.