Q: What is fail-fast vs fail-safe ?
Answer:

i. **Fail-fast:**

   A Fail-Fast iterator is an iterator that immediately throws an exception when The collection is structurally modified while iterating without using the iterator's own modification methods.

   Throws -> **ConcurrentModificationException**

   Eg:

```
ArrayList<Integer> numbers = new ArrayList<>(List.of(10, 20, 30, 40, 50));

Iterator<Integer> numIter = numbers.iterator();

numbers.add(110);
```

ii. **Fail-safe:**
   Fail-safe iterators in Java are collections that do not throw ConcurrentModificationException when the collection is modified while iterating, by iterating over a copy of the collection, rather than the actual collection itself.

   Eg:

```
Iterator<String> fruitIter = fruits.iterator();

fruits.add("Watermelon");
fruits.add("Pineapple");

while (fruitIter.hasNext())
   { System.out.print("\"" + fruitIter.next() + "\"  ");
}
```

Q: Why does ConcurrentModificationException occur ?
Answer:
A **ConcurrentModificationException** is a runtime exception in Java that occurs when a collection is structurally modified while it is being iterated in a way that the iterator cannot safely handle.

Q: Real-world example of fail-safe usage
Answer:

Real Time Notification Systems

Problem: Users subscribe or unsubscribe to notifications while a notification service iterates over subscribers to send alerts.

Fail-safe solution: CopyOnWriteArrayList or ConcurrentHashMap ensures notifications are sent without ConcurrentModificationException.

Q: Difference between Iterator and ListIterator
Answer:

i. **Iterator**
   Works with any collection data structure (Map, Set, List, etc)

   Traverses in forward direction only with method such as **hasNext()** and **next()**

   Only **remove()** method modification can be done.

   No collection index info

ii. **ListIterator:**
   Works with List implemented data structures only (ArrayList, LinkedList, etc)

   Traverses in forward, backward with method such as **hasNext()** and **next()**, **hasPrevious()** and **previous()**

   Modifications such as inserting, updating and deleting can be done using methods **add()**, **set()** and **remove()** respectively

   List index info can be retrieved using methods **nextIndex()** and **previousIndex()**
