**IMP Q/A:**

Fail-fast and Fail-safe are the concepts of concurrent modification. Concurrent modification is a process in which an object is modified concurrently when a different task is running over it. Fail-fast and Fail-safe are the iterators to iterate over the Collection objects.

**Fail-fast Iterator:**

When we use the Fail-fast iterator, it immediately throws **ConcurrentModificationException** when an element is added or removed from the collection while the thread is iterating over the collection.

Examples: Iterator in HashMap, Iterator on ArrayList, etc.

The Fail-fast iterators use **modCount**, i.e., an internal flag to check whether the collection is structurally modified or not. When the Fail-fast iterator gets the next value using the next() method, it checks the modCount flag's value. When the iterator finds the modified modCount value, it throws the ConcurrentModificationException.

Ex: Iterator on HashMap, ArrayList, and other similar data structures.

**Fail-safe Iterator:**

The Fail-safe iterator doesn't throw the **ConcurrentModificationException,** and it tries to avoid raising the exception. The Fail-safe iterator creates a copy of the original collection or object array and iterates over that copied collection. Any structural modification made in the iterator affects the copied collection, not the original collection. Therefore, the original collection remains structurally unchanged.

Ex : Iterator on ConcurrentHashMap, Iterator on CopyOnWriteArrayList and etc.