**ConcurrentModificationException is a RuntimeException**   
**Why Concurrent ModifictionException occurred?**  
=>when we try to add or remove elements from a collection, like a list, during iteration.

**1. Modifying a collection directly while you’re iterating over it using an Iterator or a for-each loop**  
// Removing an element from the list

List<String> names = new ArrayList<>(Arrays.asList("Global", "Trend", "Raj"));

for (String name : names) {

if (name.equals("Trend")) {

names.remove(name); // Throws ConcurrentModificationException

}

}

// Adding an element to the list

for (String name : names) {

if (name.equals("Trend")) {

names.add("Singh"); // unsafe modification

}

}  
  
**2. Modifying a collection in a multi-threaded environment, where one thread iterates over the collection while another thread modifies it.**

List<String> sharedList = new ArrayList<>(Arrays.asList("One", "Two", "Three"));

// Thread 1

new Thread(() -> {

for (String item : sharedList) {

System.out.println(item);

try {

Thread.sleep(100);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}).start();

// Thread 2

new Thread(() -> {

try {

Thread.sleep(200);

} catch (InterruptedException e) {

e.printStackTrace();

}

sharedList.add("Four"); // Throws ConcurrentModificationException in Thread 1

}).start();  
  
**Fixation**

1. **By Using the Iterator’s add() and/or remove() method**

add() method

List<String> names = new ArrayList<>(Arrays.asList("Alice", "Bob", "Charlie"));

Iterator<String> iterator = names.iterator();

while (iterator.hasNext()) {

String name = iterator.next();

if (name.equals("Bob")) {

iterator.add("David"); // Safe addition

}

}

Using remove()  
List<String> names = new ArrayList<>(Arrays.asList("Alice", "Bob", "Charlie"));

Iterator<String> iterator = names.iterator();

while (iterator.hasNext()) {

String name = iterator.next();

if (name.equals("Bob")) {

iterator.remove(); // Safe removal

}

}

1. **By Using Java 8 Streams and filter()**

List<String> names = new ArrayList<>(Arrays.asList("Global", "Trend", "Raj"));

names = names.stream()

.filter(name -> !name.equals("Trend"))

.collect(Collectors.toList());