

Iterator:

- ❖ It can be applied to any collection interface.
- ❖ Single direction, i.e we can traverse elements present in the collection only in the forward direction.
- ❖ We can perform a read and remove operation.
- ❖ By calling the iterator() method present in any collection interface.

13. Difference between comparable and comparator

Ans:

Comparable:

- ❖ Comparable provides a single sorting sequence. In other words, we can sort the collection on the basis of a single element such as id, name, and price.
- ❖ Comparable affects the original class, i.e., the actual class is modified. ❖ Comparable provides compareTo() method to sort elements.
- ❖ Comparable is present in the java.lang package.
- ❖ We can sort the list elements of Comparable type by Collections.sort(List) method.

Comparator:

- ❖ The Comparator provides multiple sorting sequences. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc. ❖ Comparator doesn't affect the original class, i.e., the actual class is not modified. ❖ Comparator provides compare() method to sort elements.
- ❖ A Comparator is present in the java.util package.
 - ❖ We can sort the list elements of Comparator type by Collections.sort(List, Comparator) method.

14. List down the security breaches that can happen in the frontend &

backend. Ans: Injection

Broken Authentication
Sensitive Data Exposure
XML External Entities (XXE)
Broken Access Control
Security Misconfiguration
Cross-Site Scripting (XSS)
Insecure Deserialization.