**Collection Interview Questions and Answers**

1. What is the Collection framework in Java?

Collection Framework is a combination of classes and interface, which is used to store and manipulate the data in the form of objects. It provides various classes such as ArrayList, Vector, Stack, and HashSet, etc. and interfaces such as List, Queue, Set, etc. for this purpose.

2.What are the main differences between array and collection?

Array and Collection are somewhat similar regarding storing the references of objects and manipulating the data, but they differ in many ways. The main differences between the array and Collection are defined below:

1. Arrays are always of fixed size, i.e., a user can not increase or decrease the length of the array according to their requirement or at runtime, but In Collection, size can be changed dynamically as per need.
2. Arrays can only store homogeneous or similar type objects, but in Collection, heterogeneous objects can be stored.
3. Arrays cannot provide the ?ready-made? methods for user requirements as sorting, searching, etc. but Collection includes readymade methods to use.

3. Differentiate between Collection and collections in the context of Java.

Collection : In the java.util.package, there is an interface called a collection. It's used to represent a collection of separate objects as a single entity. It's equivalent to the container in the C++ programming language. The collection framework's root interface is referred to as the collection. It has a number of classes and interfaces for representing a collection of individual objects as a single unit. The key sub-interfaces of the collection interface are List, Set, and Queue. Although the map interface is part of the Java collection framework, it does not inherit the interface's collection. The Collection interface's most significant functions are add(), remove(), clear(), size(), and contains().

Collections: The java.util.package has a utility class called Collections. It defines various utility methods for working with collections, such as sorting and searching. All of the methods are static. These techniques give developers much-needed convenience, allowing them to interact with Collection Framework more successfully. It provides methods like sort() to sort the collection elements in the normal sorting order, and min() and max() to get the minimum and maximum value in the collection elements, respectively.

4. Can you add a null element into a TreeSet or HashSet?

We can add null elements in a HashSet but we cannot add null elements in a TreeSet. The reason is that TreeSet uses the compareTo() method for comparing and it throws a NullPointerException when it encounters a null element.

5.What is the difference between HashSet and TreeSet?

The HashSet and TreeSet, both classes, implement Set interface. The differences between the both are listed below.

HashSet maintains no order whereas TreeSet maintains ascending order.

HashSet impended by hash table whereas TreeSet implemented by a Tree structure.

HashSet performs faster than TreeSet.

HashSet is backed by HashMap whereas TreeSet is backed by TreeMap.

6.What is the difference between Set and Map?

The differences between the Set and Map are given below.

1. Set contains values only whereas Map contains key and values both.
2. Set contains unique values whereas Map can contain unique Keys with duplicate values.
3. Set holds a single number of null value whereas Map can include a single null key with n number of null values.

7. What is the difference between HashSet and HashMap?

The differences between the HashSet and HashMap are listed below.

1. HashSet contains only values whereas HashMap includes the entry (key, value). HashSet can be iterated, but HashMap needs to convert into Set to be iterated.
2. HashSet implements Set interface whereas HashMap implements the Map interface
3. HashSet cannot have any duplicate value whereas HashMap can contain duplicate values with unique keys.
4. HashSet contains the only single number of null value whereas HashMap can hold a single null key with n number of null values.

8. What is the difference between HashMap and TreeMap?

The differences between the HashMap and TreeMap are given below.

1. HashMap maintains no order, but TreeMap maintains ascending order.
2. HashMap is implemented by hash table whereas TreeMap is implemented by a Tree structure.
3. HashMap can be sorted by Key or value whereas TreeMap can be sorted by Key.
4. HashMap may contain a null key with multiple null values whereas TreeMap cannot hold a null key but can have multiple null values.

9. Why we override equals() method?

The equals method is used to check whether two objects are the same or not. It needs to be overridden if we want to check the objects based on the property.

For example, Employee is a class that has 3 data members: id, name, and salary. However, we want to check the equality of employee object by the salary. Then, we need to override the equals() method.

10. What is the advantage of the generic collection?

There are three main advantages of using the generic collection.

1. If we use the generic class, we don't need typecasting.
2. It is type-safe and checked at compile time.
3. Generic confirms the stability of the code by making it bug detectable at compile time.

11. What is hash-collision in Hashtable and how it is handled in Java?

Two different keys with the same hash value are known as hash-collision. Two separate entries will be kept in a single hash bucket to avoid the collision. There are two ways to avoid hash-collision.

1. Separate Chaining
2. Open Addressing

12. What do you understand by fail-fast?

The Iterator in java which immediately throws ConcurrentmodificationException, if any structural modification occurs in, is called as a Fail-fast iterator. Fail-fats iterator does not require any extra space in memory.

13. How to make Java ArrayList Read-Only?

We can obtain java ArrayList Read-only by calling the Collections.unmodifiableCollection() method. When we define an ArrayList as Read-only then we cannot perform any modification in the collection through add(), remove() or set() method.

14. How to synchronize ArrayList?

We can synchronize ArrayList in two ways.

1. Using Collections.synchronizedList() method
2. Using CopyOnWriteArrayList<T>

15. What is CopyOnWriteArrayList?

CopyOnWriteArrayList is a variant of ArrayList in which operations like add and set are implemented by creating a copy of the array. It is a thread-safe, and thereby it does not throw ConcurrentModificationException. This ArrayLists permits all the elements, including null.