Day 6 - Interview Questions

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:

- 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.
- Arrays can only store homogeneous or similar type objects, but in Collection, heterogeneous objects can be stored.
- Arrays cannot provide the ?ready-made? methods for user requirements as sorting, searching, etc. but Collection includes readymade methods to use.

3. Explain various interfaces used in the Collection framework?

Collection framework implements various interfaces, Collection interface and Map interface (java.util.Map) are the mainly used interfaces of Java Collection Framework. List of interfaces of Collection Framework is given below:

A. Collection interface: Collection (java.util.Collection) is the primary interface, and every collection must implement this interface.
B. List interface: List interface extends the Collection interface, and it is an ordered collection of objects. It contains duplicate elements. It also allows random access of elements.
C. Set interface: Set (java.util.Set) interface is a collection which cannot contain duplicate elements. It can only include inherited methods of Collection interface
D. Queue interface: Queue (java.util.Queue) interface defines queue data structure, which stores the elements in the form FIFO (first in first out).
E. Deque interface: it is a double-ended-queue. It allows the insertion and removal of elements from both ends. It implants the properties of both Stack and queue so it can perform LIFO (Last in first out) stack and FIFO (first in first out) queue operations.
F. Map interface: A Map (java.util.Map) represents a key, value pair storage of elements. Map interface does not implement the Collection interface. It can only

contain a unique key but can have duplicate elements. There are two interfaces which implement Map in java that are Map interface and Sorted Map.

4. What is the difference between ArrayList and Vector?

No.	ArrayList	Vector
1)	ArrayList is not synchronized.	Vector is synchronized.
2)	ArrayList is not a legacy class.	Vector is a legacy class.
3)	ArrayList increases its size by 50% of the array size.	Vector increases its size by doubling the array size.
4)	ArrayList is not thread-safe as it is not synchronized.	Vector list is thread-safe as every method is synchronized.

5. What is the difference between ArrayList and LinkedList?

No	ArrayList	LinkedList
1)	ArrayList uses a dynamic array.	LinkedList uses a doubly linked list.
2)	ArrayList is not efficient for manipulation because too much is required.	LinkedList is efficient for manipulation.
3)	ArrayList is better to store and fetch data.	LinkedList is better to manipulate data.
4)	ArrayList provides random access.	LinkedList does not provide random access.
5)	ArrayList takes less memory overhead as it stores only object	LinkedList takes more memory overhead, as it stores the object as well as the address of that object.

6. What is the difference between Iterator and ListIterator?

Iterator traverses the elements in the forward direction only whereas ListIterator traverses the elements into forward and backward direction.

No	Iterator	ListIterator

1)	The Iterator traverses the elements in the forward direction only.	ListIterator traverses the elements in backward and forward directions both.
2)	The Iterator can be used in List, Set, and Queue.	ListIterator can be used in List only.
3)	The Iterator can only perform a remove operation while traversing the collection.	ListIterator can perform ?add,? ?remove,? and ?set? operation while traversing the collection.

7. What is the difference between Iterator and Enumeration?

No	Iterator	Enumeration
1)	The Iterator can traverse legacy and non-legacy elements.	Enumeration can traverse only legacy elements.
2)	The Iterator is fail-fast.	Enumeration is not fail-fast.
3)	The Iterator is slower than Enumeration.	Enumeration is faster than Iterator.
4)	The Iterator can perform a remove operation while traversing the collection.	The Enumeration can perform only traverse operations on the collection.

8. What is the difference between List and Set?

The List and Set both extend the collection interface. However, there are some differences between the two which are listed below.

- The List can contain duplicate elements whereas Set includes unique items.
- The List is an ordered collection which maintains the insertion order whereas Set is an unordered collection which does not preserve the insertion order.
- The List interface contains a single legacy class which is the Vector class whereas Set interface does not have any legacy class.
- The List interface can allow a number of null values whereas Set interface only allows a single null value.

9. What is the difference between Collection and Collections?

The differences between the Collection and Collections are given below.

- The Collection is an interface whereas Collections is a class.
- The Collection interface provides the standard functionality of data structure to List, Set, and Queue. However, the Collections class is to sort and synchronize the collection elements.
- The Collection interface provides the methods that can be used for data structure whereas the Collections class provides the static methods which can be used for various operations on a collection.

10. What is the difference between Comparable and Comparator?

No	Comparable	Comparator
1)	Comparable provides only one sort of sequence.	The Comparator provides multiple sorts of sequences.
2)	It provides one method named compareTo().	It provides one method named compare().
3)	It is found in the java.lang package.	It is located in the java.util package.
4)	If we implement the Comparable interface, The actual class is modified.	The actual class is not changed.

11. What are the best practices for Java Collections Framework?

Following are some of the best practices while using Java Collections:

- Programs should be written as interfaces, not implementations, so we can modify the implementation later.
- Whenever possible, use Generics to ensure type safety and avoid ClassCastExceptions.
- Choosing the appropriate type of collection based on the need. For example, if the size is fixed, we might want to use an Array over an ArrayList. When iterating over the Map, we should use LinkedHashMap. Set is the best way to avoid duplicates.
- Use immutable classes provided by JDK as keys in Map to avoid implementation of hashCode() and equals().

- In order to increase the readability of the code, we should use isEmpty() instead of finding the size of the collection and comparing it to zero.
- Rather than writing your own implementation, use the Collections utility class to get read-only, Synchronized, or empty collections instead. It enhances code reuse while resulting in greater stability.

12. What is the advantage of the generic collection?

There are three main advantages of using the generic collection.

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