COLLECTION QUESTIONS

1) Difference Between Collections & Collection.

Collection	**Collections**
It is an Interface.	It is an Utility class.
It's a root interface in java collection framework.	It's a helper class static method working with collections.
Java.util.collection	Java.util.collections
Collection is a blue print for all collection type in java.	Collections is a toolbox, usage of static methods.

2) Collection Hierarchy

*) It is part of the **Java Collections Framework**. It provides Interface and classes for storing and manipulating the group of data.

3) Difference between List, set and map.

List is a Interface extends from the collection interface.	Set is a Interface extends from the collection interface.	Map is a class implement from the map interface.
It follows the insertion order format.	It doesn't follow the insertion order format.	It doesn't follow the insertion order format, because it follows the hashing Technique.
It allows the duplicate values.	Set doesn't allow the duplicate value.	Key should be unique and values can have duplicates.

4) Difference between Array and Arraylist.

It is used to store Primitive and Non primitive value.	It is Stored only non-primitive value.
There is no default size in Array.	Arraylist default size is 10. It follows the Load Factor mechanism to extend the size.
Array is fixed in size.	ArrayList is not fixed in size.

5) Difference between ArrayList and LinkedList.

It is a Dynamic Array.	It is a Doubly LinkedList.
It presented a java.util.ArrayList	It presented a java.util.LinkedList
It is Inefficient memory	It is good memory utilization
Insertion operation is slow	Insertion operation is fast
Delection operation is not very efficiency	Delection operation is very efficient
It is used to store only similar types of data.	It is used to store any type of data

COLLECTION QUESTIONS

6) Convert Array Element to ArrayList

```
Import java.util.*;

public class Number {
public static void main(String [] args) {
int [] nums = {1,2,3,4,5};

ArrayList<Integer> numberList = new ArrayList<>();
For(int num : nums) {
numberList.add(num);
      }
System.out.println(numberList);
   }
}
```

7) Covert ArrayList to Array by Using with and without in-build function

```

Public class words {
Public static void main(String[] args) {
ArrayList<String> List = new ArrayList<>();
List.add("Apple");
List.add("Mango");
List.add("Banana");

String [] arr = List.toArray(new String[0]);
For(String i : arr) {
System.out.println(i);
   }
  }
}
```

8) Difference between ArrayList and HashMap

ArrayList	**HashMap**
The java ArrayList implements from the List Interface.	The HashMap implements from the map Interface.
ArrayList always maintain the Insertion order Format.	HashMap doesn't maintain the Insertion order Format
ArrayList stored only elements.	HashMap stores key and value pairs.
ArrayList can contains duplicate elements.	HashMap doesn't contain duplicate keys but values contain duplicate values.
We can have any number of null values in ArrayList.	We can have only one null key and any number of null values in HashMap.

COLLECTION QUESTIONS

9) How ArrayList stores data internally and explains load factor and Memory Shifting.

*) Array of object to store the elements. The Default array is created the list is Utilized. When the array becomes full and new element is added, the array is resized.

Load Factor Mechanism

*) ArrayList default size is 10. If fill the capacity of 75% and it extend size of 50% of the default size.

Memory Shifting

*) Memory shifting is moving the data, one memory location to another memory Location. Usually fill the gap or make a room for new data.

10) Difference Between Array and Collection

Array	**Collection**
It is a Fixed Size.	It is a Dynamic Size
Arrays can hold Only homogeneous data type elements.	Collection can hold both homogeneous and heterogeneous elements
It stores primitive and non-primitive values	It stores only Objects

11) What is Iterable?

*) Iterable is a interface in java that represents a collection of elements which can be iterated one by one, usually in for each loop.

12) What is Hashing Technique?

*) Hashing is a Technique used to convert data to a fixed size value, known as a hash code and hash value. It is commonly used in data Structures like HashMap and HashSet to enable fast Retrieval.

13) Explain Hash bucket, Hash collision, hash Code, equals(), and Hash table.

Hash Collision

*) It happens When the index values are same. Object to avoid hash collision if checks with equals(), before storing in bucket. If both the key are same it will override. Otherwise it will create a new code and store the values.

Hash Code

*) Java returns the integer hash code values of the object, and used to find out the right bucket for the object.

Equals

*) It is used to compare the keys and values of the object to avoid the collision.

Hash Table

*) Collection of has buckets is called hash table.

14) What is Iterator?

*) It is an interface that is used to remove and elements while iterating.

*) It is used to avoid conCurrentModification() Exception.

*) It is represented from java.util.package.

*) It has some method likes, hashNext(), next(), remove().

15) Purpose of hashNext(), next(), remove() methods.

*) These methods are fundamental to using an Iterator in java.

Hash Next

*) To check if there are more elements left to iterator.

-> True – if another elements exists.

-> False – If you have reached the end.

Next

*) To get the next element in the iteration.

*) The next element in the collision.

*) You should only call hashNext() to avoid NoSuchElementException.

Remove

*) To remove the last element returned by next() from the collision.

*) You must call next() before remove() otherwise it throws Illegal State Exception.

16) What is concurrent Modification Exception?

*) It is a runtime exception in java. It is an Error when you try to change a collection while Iterating.

*) While Iterating, the system watching the list and any changes to the list makes system confuse and throws an exception.

COLLECTION QUESTIONS

17) Difference Between Fail fast and Fail safe?

Fail Fast	**Fail Safe**
Throws concurrent modification exception in the collection modified using Iteration.	Doesn't exception is modified during iteration.
Directly works on the original collection.	Works on the copy of the collection.
It performance in faster	It performance in slower.

18) FIFO & LIFO

First In First Out

*) The first element is added, first to be removed. Example: Ticket counter.

Last In First Out

*) The Last element added is the first to be removed. Example: Stack of plates.

19) What are the Subclasses are queue interfaces?

*) The Queue Interface is a part of the collection Framewor k.

Subclasses (or) Implementing Classes

- ➔ Linked List - Doubly Linked List
- ➔ Priority Queue – Based on priority
- ➔ Array Dequeue – fast no capacity restriction
- ➔ Concurrent Linked Queue – non blocking thread safe
- ➔ Linked Blocking Queue – Blocking thread safe

20) Difference Between Comparable and Comparator?

Comparable	**Comparator**
It is used to Nature sorting order object	It is used for custom sorting order separate from the class
Compareto() method inside the class	Override the Compare() method

21) Difference Between HashMap() and Hash Table()?

Hash map	**Hash Table**
It is Not thread Safe and unsynchronized	It is Thread safe and synchronized.
It is Faster when compared to Hash Table	It is Slower when compare to hash Map.
Allows one null key and have any null values.	No null key or values

COLLECTION QUESTIONS

It’s fail fast	It’s not fail fast

22) Difference Between hash Set() and hash map()?

Hash Set	**Hash Map**
Store unique elements	Stores data as key & values pair
No key only values	Key must be unique but values can repeat
Allows one null elements	One null key and multiple null values

23) Join two ArrayList(single Array without Stream)?

```
import java.util.*;
import java.util.stream.Collectors;
import java.util.stream.Stream;

public class JoinArrayListExample {
   public static void main(String[] args) {
      List<Integer> list1 = Arrays.asList(10, 20, 30);
      List<Integer> list2 = Arrays.asList(40, 50, 60);

      // 🔸 Method 1: Without Stream
      List<Integer> num = new ArrayList<>(list1);
      num.addAll(list2);
      System.out.println("Without Stream: " + num);

      // 🔹 Method 2: With Stream
      List<Integer> nums = Stream.concat(list1.stream(), list2.stream())
                                  .collect(Collectors.toList());
      System.out.println("With Stream: " + nums);
   }
}
```

24) What will happen it two different keys of hashmap returns the same hashcode()

*) This is called hash collision. Both index values are same for the object. It happens hash collision. If check with equals method before storing it in bucket of both keys and same it will override otherwise it will create a new code and store the values.

25) What is Stream API?

*) Stream API introduced in java 8 features. It used access to large no of data collection in pipelined structure and it has various method to get desired result.

*) It has intermediate and terminal operation method. Its like collect(), count() intermediate and skip(), limit() is terminal.

26) Difference Between Intermediate and Terminal

Intermediate	**Terminal**
Returns a new Stream result	Returns a non-stream result
Do not Trigger the exception	Trigger the exception of the pipelined
It is used to build a processing pipelined	Can be final value a collection

27) List out the Stream API Methods

*) Filter(), Map(), Collect(), Limit(), Skip(), Flatmap(), findfirst(), min(), max(), all match(), Sorted(), Distinct(), toArray(), reduce(), Stream(), count().

28) What are the collectors to explain some of its methods?

*) Collectors is Utility class in java.utill.stream. that provides ready made implementations of the collector interfaces.

*) They are mainly used with the collect() terminal operation to gather elements from a stream into a result like list, set map.

29) What are the predefine packages are there

*) java.land -> cored Default

*) java.util -> Utility classes

*) java.io -> Input and Outpot

*) java.net -> Networking

*) java.sql -> Used for database connectivity (JDBC).

30) How to remove the duplicates in Array list, what are the way available?

```
List<Integer> list = Arrays.asList(1, 2, 2, 3, 4, 4, 5);

List<Integer> unique = list.stream().distinct().sorted().collect(Collectors.toList());

System.out.println(unique);
```

COLLECTION QUESTIONS