**Q1>What is collection in java?**

**Ans ==** In java a collection is a framework that provides an architecture to store and manipulate a group of objects.it is part of the java Collections Framework,which is a set of classes and interfaces that offer various data structures and algorithms to work with collections of objects.

The main purpose of using collections in java is to provide a more efficient and convenient way to handle groups of elements,such as lists,sets,maps,etc.

**Q2>Differentiate between Collection and collections in the context of java.**

**Ans == 1.Collection** (with an uppercase”C”) is an interface in the java Collections Framework.It is the root interface of the framework and defines the basic methods and behaviours that all collection classes must implement.Collection provides a common set of methods for working with collections,such as adding,removing and querying elements.It represents a group of objects, and its main subintefaces are List,Set,and Queue.Collection (uppercase) is part of the ‘java.util’ package.

**2.collections (lowercase):** The term “collections”(with a lowercase “c”) is a more generic term and does not refer to a specific java class or interface.It is used informally to refer to various data structures or collections of objects in general.When talking about “collections”(lowercase), it can be any grouping of elements,such as arrays,lists,sets,maps,etc. regardless of whether they are part of the Colections Framework.

**Q3>What are the advantages of the collection Framework?**

**Ans ==** The java Collections Framework provides several advantages that make it a powerful tool for handling groups of objects.Some of the key advantages are:

**1.Reusable Data Structures:** The framework offers a wide range of data structures (lists,sets,maps,queues,etc) implemented through various classes.These pre-implemented data structure are readily available for use,saving developers from reinventing the wheel and promoting code reuse.

**2.Dynamic Resizing:** Unlike arrays, which have a fixed size, collections can dynamically resize themselves as elements are added or removed. This dynamic resizing simplifies the management of collections and avoids the need for manual resizing operations.

**3.Type Safety:** The java Collections Framework is designed to be type-safe.Generics were introduced with java 5 to ensure that collections hold a specific type of elements, making the code less error-prone and providing compile-time safety against type-related errors.

**Q4>Explain the various interfaces used in the collection framework.**

**Ans ==** In the java Collections Framework,interfaces are used to define the contract and common behavior that different collection classes should adhere to.The interfaces provide a set of methods that collection classes must implement,ensuring consistency and uniformity in how collections are used.Below are the main interfaces used in the collection framework:

1>Collection<E>

2>List<E>

3>Set<E>

4>SortedSet<E>

**Q5>Differentiate between List and Set in java.**

**Ans ==** In java, both List and Set are interfaces in the java Collections Framework,and they represent different types of collections.They have distinct characteristics that set them set them apart in terms of their behavior and use cases:

**List:**

**1.Ordering:**

\*List is an ordered collection,which means the elements are stored in a specific sequence based on their insertion order.

\*The order of elements in a List is predictable and consistent.When you iterate over a List,you will get elements in the same order as they were added.

**2.Duplicates:**

\*List allow duplicate elements.You can have multiple elements with the same value in a List.

**Set:**

**1.Ordering:**

\*Sets is an unordered collection,which means the elements in a Set have no guaranted order.

\*The elements are not stored in the order they are added,and the order of elements in a Set may change over time or depending on the specific set implementation.

**2.Duplicates:**

\*Sets do not allow duplicate elements.Each element in a Set must be unique.

\*if you attempt to add a duplicate element to a Set,it will be rejected,and the Set will remain unchanged.

**Q6>What is the difference between iterator and ListIterator on java?**

**Ans == Iterator:**

1.’Iterator’ is a more general-purpose interface that allows you to traverse elements in a collection in a forward-only direction.

2.It is available for most standard collections like ‘ArrayList’, ‘LinkedList’, ‘HashSet’, ‘TreeSet’,etc.

**ListIterator:**

1. ‘ListIterator’ is a more specialized interface that extends ‘Iterator’ and is specific to lists (‘List’ implementation).
2. It allow you to traverse element in both forward and backward directions,and it provides additional methods for more fine-grained list manipulation.

**Q7>What is the difference between Comparable and Comparator?**

**Ans == Comparable:**

1.’Comparable’ is a natural ordering interface that allows objects of a class to be compared each other for the purpose of sorting.

2.When a class implements the ‘Comparable’ interface,it defines a single method called ‘compareTo()’, which is used to determine the natural ordering of objects of that class.

3.The natural ordering establish by ‘Comparable’ is used by sorting methods in java collections, like ‘Array.sort()’ and ‘Collections.sort()’.

**Comparator:**

1.’Comparator’ is a comparision interface that provides a way to define custom comparision logic for sorting objects that do not implement the ‘Comparable’ interface,or when you wants to sort objects in a different way than their natural ordering.

2.It allows you to create multiple comparision strategies for the same class.

3.Unlike ‘Comparable’,which defines the natural ordering within the class, the ‘Comparator’ is used as an external sorting mechanism and can be passed to sorting methods.

**Q8>What is collision in HashMap?**

**Ans ==** In a ‘HashMap’, a collision occurs when two or more different keys hash to the same index in the underlying array that is used to store the key-value pairs.This situation can happen because the number of possible hash codes is typically smaller than the number of elements in the collection.Since multiple keys can potentially map to the same index, a collision occurs, and the keys need to be handled appropriately to maintain the intergrity of the data structure.

**Q9>Distinguish between a hashmap and a Treemap.**

**Ans == HashMap:**

**1.**’HashMap’: Uses an array of linked lists (or sometimes a combination of linked lists and red-black trees in later java version) to store the key-value pairs.It uses the hash code of the keys to determine the index in the array,allowing for quick retrieval of values based on their keys.Hashing provides average-case constant-time (O(1)) complexity for basic operations like put and get, assuming a good hash function and minimal collisions.

**2.’**HashMap’:Offers better average-case performance for basic operations like ‘put()’ and ‘get()’ due to its constant-time complexity (O(1)), assuming a good hash function.

**TreeeMap:**

**1.**’treeMap’: Uses a Red-Black tree data structure to store the key-value pairs. The elements in a ‘TreeMap’ are sorted based on the natural ordering of the keys or a custom comparator provided during the creation of the ‘TreeMap’.As a result, operations like put and get have logarithmic time complexity (O(log n)),where n is the number of elements in the map.The tree structure ensure that the elements are always sorted, allowing for efficient range-based operations.

**2.**’TreeMap’:Maintains the keys in sorted order, either based on their natural ordering or a custom comparator.This means that iterating through a ‘TreeMap’ will return the elements in their sorted order,which can be useful in scenarios where you need the keys to be processed in a specific sequence.

**Q10>Define LinkedHashMap in java?**

**Ans ==** ‘LinkedHashMap’ is a class in java that extends the ‘HashMap’ class and implements the ‘Map’ interface.It is a variation of the ‘HashMap’ that maintains the order of insertion of its elements. In addition to the key-value pairs, ‘LinkedHashMap’ also maintains a doubly-linked list to keep track of the order in which elements were added.