

12th Assignment on Collection Framework and Maps
Java and DSA with System Design 2.0

Q1. What is the Collection framework in Java?

Ans:

The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes ([ArrayList](#), Vector, [LinkedList](#), [PriorityQueue](#), HashSet, LinkedHashSet, TreeSet).

Q2. What is the difference between ArrayList and LinkedList?

Ans:

ArrayList	LinkedList
1) ArrayList internally uses a dynamic array to store the elements.	LinkedList internally uses a doubly linked list to store the elements.
2) Manipulation with ArrayList is slow because it internally uses an array. If any element is removed from the array, all the other elements are shifted in memory.	Manipulation with LinkedList is faster than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory.
3) An ArrayList class can act as a list only because it implements List only.	LinkedList class can act as a list and queue both because it implements List and Deque interfaces.
4) ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.
5) The memory location for the elements of an ArrayList is contiguous.	The location for the elements of a linked list is not contiguous.
6) Generally, when an ArrayList is initialized, a default capacity of 10 is assigned to the ArrayList.	There is no case of default capacity in a LinkedList. In LinkedList, an empty list is created when a LinkedList is initialized.

12th Assignment on Collection Framework and Maps

Java and DSA with System Design 2.0

7) To be precise, an ArrayList is a resizable array.

LinkedList implements the doubly linked list of the list interface.

Q3. What is the difference between Iteration and ListIteration?

Ans:

Iterator	ListIterator
Can traverse elements present in Collection only in the forward direction.	Can traverse elements present in Collection both in forward and backward directions.
Helps to traverse Map, List and Set.	Can only traverse List and not the other two.
Indexes cannot be obtained by using Iterator.	It has methods like nextIndex() and previousIndex() to obtain indexes of elements at any time while traversing List.
Cannot modify or replace elements present in Collection	We can modify or replace elements with the help of set(E e)
Cannot add elements and it throws ConcurrentModificationException.	Can easily add elements to a collection at any time.
Certain methods of Iterator are next(), remove() and hasNext().	Certain methods of ListIterator are next(), previous(), hasNext(), hasPrevious(), add(E e).

12th Assignment on Collection Framework and Maps
Java and DSA with System Design 2.0

Q4. What is the difference between Iteration and Enumeration?

Ans:

Iterator	Enumeration
Iterator is a universal cursor as it is applicable for all the collection classes.	Enumeration is not a universal cursor as it applies only to legacy classes.
Iterator has the remove() method.	Enumeration does not have the remove() method.
Iterator can do modifications (e.g using remove() method it removes the element from the Collection during traversal).	Enumeration interface acts as a read only interface, one can not do any modifications to Collection while traversing the elements of the Collection.
Iterator is not a legacy interface. Iterator can be used for the traversal of HashMap, LinkedList, ArrayList, HashSet, TreeMap, TreeSet .	Enumeration is a legacy interface which is used for traversing Vector, Hashtable.

Q5. What is the difference between List and Set?

Ans:

List	Set
1. The List is an indexed sequence.	1. The Set is an non-indexed sequence.
2. List allows duplicate elements	2. Set doesn't allow duplicate elements.
3. Elements by their position can be accessed.	3. Position access to elements is not allowed.

12th Assignment on Collection Framework and Maps

Java and DSA with System Design 2.0

List	Set
4. Multiple null elements can be stored.	4. Null element can store only once.
5. List implementations are ArrayList, LinkedList, Vector, Stack	5. Set implementations are HashSet, LinkedHashSet.

Q6. What is the difference between HashSet and TreeSet?

Ans:

Parameters	HashSet	TreeSet
Ordering or Sorting	It does not provide a guarantee to sort the data.	It provides a guarantee to sort the data. The sorting depends on the supplied Comparator.
Null Objects	In HashSet, only an element can be null.	It does not allow null elements.
Comparison	It uses hashCode() or equals() method for comparison.	It uses compare() or compareTo() method for comparison.
Performance	It is faster than TreeSet.	It is slower in comparison to HashSet.
Implementation	Internally it uses HashMap to store its elements.	Internally it uses TreeMap to store its elements.
Data Structure	HashSet is backed up by a hash table.	TreeSet is backed up by a Red-black Tree.
Values Stored	It allows only heterogeneous value.	It allows only homogeneous value.

12th Assignment on Collection Framework and Maps
Java and DSA with System Design 2.0

Q7. What is the difference between Array and ArrayList?

Ans:

Basis	Array	ArrayList
Definition	An array is a dynamically-created object. It serves as a container that holds the constant number of values of the same type. It has a contiguous memory location.	The ArrayList is a class of Java Collections framework. It contains popular classes like Vector , HashTable , and HashMap .
Static/ Dynamic	Array is static in size.	ArrayList is dynamic in size.
Resizable	An array is a fixed-length data structure.	ArrayList is a variable-length data structure. It can be resized itself when needed.
Initialization	It is mandatory to provide the size of an array while initializing it directly or indirectly.	We can create an instance of ArrayList without specifying its size. Java creates ArrayList of default size.
Performance	It performs fast in comparison to ArrayList because of fixed size.	ArrayList is internally backed by the array in Java. The resize operation in ArrayList slows down the performance.
Primitive/ Generic type	An array can store both objects and primitives type.	We cannot store primitive type in ArrayList. It automatically converts primitive type to object.
Iterating Values	We use for loop or for each loop to iterate over an array.	We use an iterator to iterate over ArrayList.
Type-Safety	We cannot use generics along with array because it is not a convertible type of array.	ArrayList allows us to store only generic/ type , that's why it is type-safe .
Length	Array provides a length variable which denotes the length of an array.	ArrayList provides the size() method to determine the size of ArrayList.

12th Assignment on Collection Framework and Maps

Java and DSA with System Design 2.0

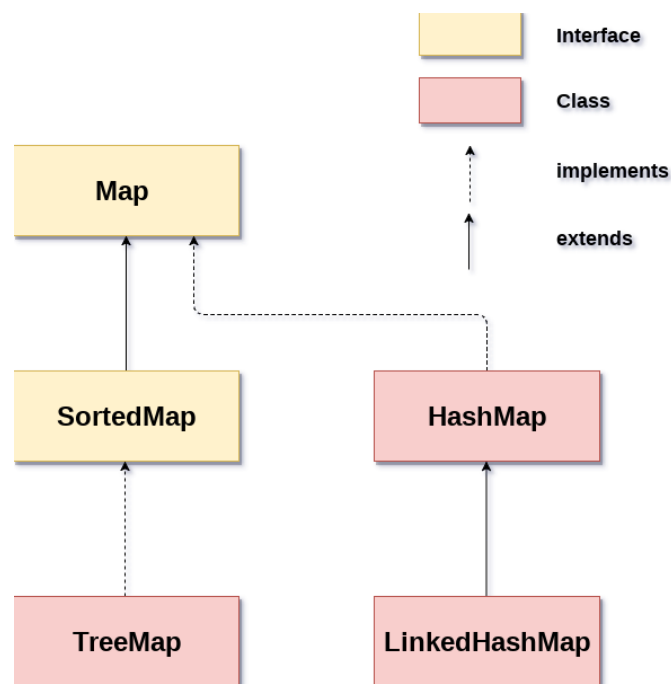
Adding Elements	We can add elements in an array by using the assignment operator.	Java provides the add() method to add elements in the ArrayList.
Single/ Multi- Dimensional	Array can be multi-dimensional .	ArrayList is always single-dimensional .

Q8. What is a Map in Java?

Ans:

A Map is an object that maps keys to values. A map cannot contain duplicate keys: Each key can map to at most one value. It models the mathematical function abstraction.

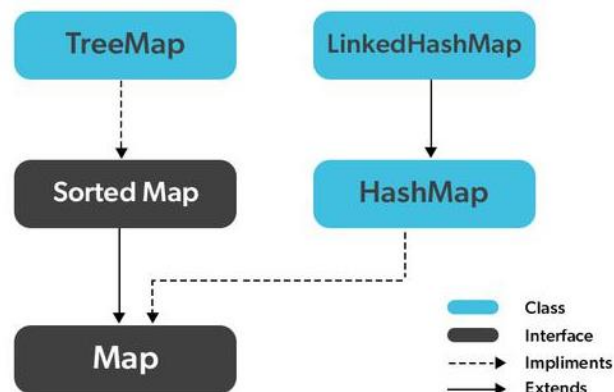
There are two interfaces for implementing Map in java: Map and SortedMap, and three classes: HashMap, LinkedHashMap, and TreeMap. The hierarchy of Java Map is given below:



Q9. What are the commonly used implementations of Map in Java?

Ans:

The three general-purpose Map implementations are HashMap , TreeMap and LinkedHashMap .



Q10. What is the difference between HashMap and TreeMap?

Ans:

HashMap	TreeMap
It does not provide any order for elements.	It provides orders for elements.
It's speed is fast.	It's speed is slow.
It allows one key as null and also allows multiple values.	It does not allow key as null but it allows multiple null values.
It consumes more memory space.	It consumes less memory space.
It has only basic features.	It has advanced features.
For comparing keys, equals() is used.	For comparing keys, compare or compareTo() is used.
It's complexity is O(1).	It's complexity is O(log n).

12th Assignment on Collection Framework and Maps
Java and DSA with System Design 2.0

Q11. How do you check if a key exists in a Map in Java?

Ans: `containsKey()` method is used to check whether a particular key is being mapped into the HashMap or not. It takes the key element as a parameter and returns True if that element is mapped in the map.