Collections

1. Differences between Arrays and Collections?

Arrays	**Collections**
It is a collection of homogeneous data elements.	It is a collection of homogeneous and heterogeneous data elements.
Arrays are fixed in size.	Collections are growable in nature.
Performance point of view arrays are recommended to use.	Memory point of view collections are recommended to use.
Arrays are type safe.	Collections are not type safe.
Arrays are not implemented based on data structure concept so we can't expect any readymade (utility) methods.	Collections are implemented based on data structure concept so we can expect readymade (utility) methods.
It holds primitive and object types.	It holds only object types but not primitive types.

2. What is Collection framework?

   It defines several classes and interfaces to represent a group of objects as a single entity.

3. What is Collection?

   - Collection is an interface which is present in java.util package.
   - It is a root interface for entire Collection framework.
   - If we want to represent group of individual objects in a single entity then we should go for Collection.
   - Collection interface defines the most common methods which can be applicable for any collection object.

4. Differences between Collection and Collections?

   - Collection is an interface which can be used to represent a group of objects as a single entity.

   - Collections is an utility class present in java.util package to define several utility methods for Collection objects.

5. What are the methods present in Collection interface?

The following is the list of methods present in Collection interface.

- boolean add(Object o);
- boolean addAll(Collection c);
- boolean remove(Object o);
- boolean removeAll(Object o);
- boolean retainAll(Collection c);
- void clear();
- boolean contains(Object o);
- boolean containsAll(Collection c);
- boolean isEmpty();
- int size();
- Object[] toArray();
- Iterator iterator();

6. Differences between ArrayList vs Vector?

ArrayList	**Vector**
No method is synchronized	Every method is synchronized
At a time multiple Threads are allow to operate on ArrayList object and hence ArrayList object is not Thread safe.	At a time only one Thread is allow to operate on Vector object and hence Vector object is Thread safe.
Relatively performance is high because Threads are not required to wait.	Relatively performance is low because Threads are required to wait.
It is non legacy and introduced in 1.2v.	It is legacy and introduced in 1.0v.

7. Differences between ArrayList and LinkedList?

ArrayList	**LinkedList**
The underlying data structure is resizable array or growable array.	The underlying data structure is doubly linked list.
ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.
The memory location for the elements of an ArrayList is contiguous.	The location for the elements of a linked list is not contagious.
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.

8. Differences between List and Set?

List	**Set**
If we want to represent group of individual objects in a single entity where duplicates are allowed and order is preserved then we need to use List.	If we want to represent group of individual objects in a single entity where duplicates are not allowed and order is not preserved then we need to use Set.
List allows us to add any number of null values.	Set allows us to add at least one null value in it.
List implementation classes are ArrayList, LinkedList and Vector.	Set implementation classes are HashSet, LinkedHashSet and TreeSet.
ListIterator cursor is used to iterate the List elements.	Iterator cursor is used to iterate the set elements.

9. Differences between HashSet and LinkedHashSet?

HashSet	LinkedHashSet
The underlying data structure is Hashtable.	The underlying data structure is Hastable and LinkedList.
Insertion order is not preserved.	Insertion order is preserved.
Introduced in 1.2 version.	Introduced in 1.4 version.

10. Differences between HashSet and TreeSet?

HashSet	**TreeSet**
The underlying data structure is Hashtable.	The underlying data structure is Balanced Tree.
Null insertion is possible.	Null insertion is not possible.
Heterogeneous objects are allowed.	Heterogeneous objects are not allowed.
Insertion order is not preserved.	Insertion order is sorting order of an object.

11. Differences between Comparable and Comparator interface?

Comparable	**Comparator**
It is present in java.lang package	It is present in java.util package
It contains only one method i.e compareTo()	It contains two methods i.e compare() and equals()
If we depend upon natural sorting order then we need to use Comparable interface.	If we depend upon customized sorting order then we need to use Comparator interface.

12. What is Map interface?

- Map is an interface which is present in java.util package.
- It is a not a child interface of Collection interface.
- If we want to represent group of objects in key, value pair then we need to use Map.
- Each key, value pair is called single entry.
- Both key and value must be objects.
- Key can’t be duplicate but value can be duplicate.

13. Differences between HashMap and LinkedHashMap?

HashMap	**LinkedHashMap**
The underlying data structure is Hashtable.	The underlying data structure is Hastable and LinkedList.
Insertion order is not preserved.	Insertion order is preserved.
Introduced in 1.2 version.	Introduced in 1.4 version.

14. Differences between HashMap and TreeMap?

HashMap	**TreeMap**
The underlying data structure is Hashtable.	The underlying data structure is Red Black Tree.
Insertion order is not preserved.	Insertion order is sorting order of an object.
Both key and value can be null.	Key can’t be null but value can be null.

15. Differences between TreeMap and Hashtable?

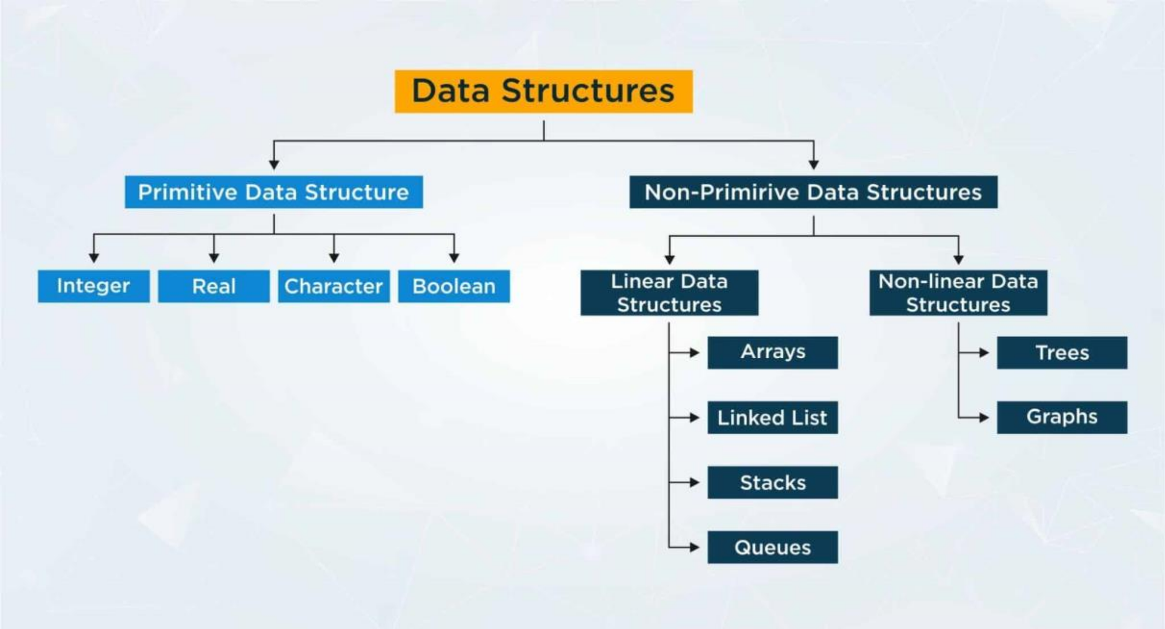
TreeMap	**Hashtable**
The underlying data structure is Red Black Tree.	The underlying data structure is Hashtable.
Key can’t be null but value can be null.	Both key and value can’t be null.
Both key and value can be null.	Key can’t be null but value can be null.
It is a non-legacy class.	It is a legacy class
It is introduced in 1.2 version.	It is introduced in 1.0 version
Methods are not synchronized.	All methods are synchronized.

16. Types of Cursors in java?

We have three types of cursors in java.

Enumeration	Iterator	ListIterator
It is used to read objects one by one from legacy Collection objects.	It is used to read objects one by one from any Collection objects.	It is used to read objects one by one from List Collection objects.
It contains 2 methods i.e hasMoreElements() and nextElement().	It contains 3 methods i.e hasNext(), next() and remove()	It contains 9 methods i.e hasNext(), next(), hasPrevious(), previous(), remove(), set(), add(), previousIndex() and nextIndex().
It performs read operation.	It performs read and remove operation.	It perform read, remove, adding and replacement of new objects.
We can create object by using elements() method.	We can create object by using iterator() method.	We can create object by using listIterator() method.
It is not a universal cursor.	It is a universal cursor.	It is a bi-directional cursor.

17. Types of Data Structure in java?


Data Structures
Primitive Data Structure
Non-Primirive Data Structures
Integer
Real
Character
Boolean
Linear Data Structures
Non-linear Data Structures
Arrays
Linked List
Stacks
Queues
Trees
Graphs