

## Q1 What is Collection ? What is a Collections Framework ? What are the benefits of Java Collections Framework ?

Collection : A collection (also called as container) is an object that groups multiple elements into a single unit.

Collections Framework : Collections framework provides unified architecture for manipulating and representing collections.

Benefits of Collections Framework :

1. Improves program quality and speed
2. Increases the chances of reusability of software
3. Decreases programming effort.

## Q2 What is the root interface in collection hierarchy ?

Root interface in collection hierarchy is Collection interface . Few interviewer may argue that

Collection interface extends Iterable interface. So iterable should be the root interface. But you should reply iterable interface present in java.lang package not in java.util package .It is clearly mentioned in [Oracle Collection docs](#) , that Collection interface is a member of the Java Collections framework. For [Iterable interface Oracle doc](#) , iterable interface is not mentioned as a part of the Java Collections framework .So if the question includes collection hierarchy , then you should answer the question as Collection interface (which is found in java.util package).

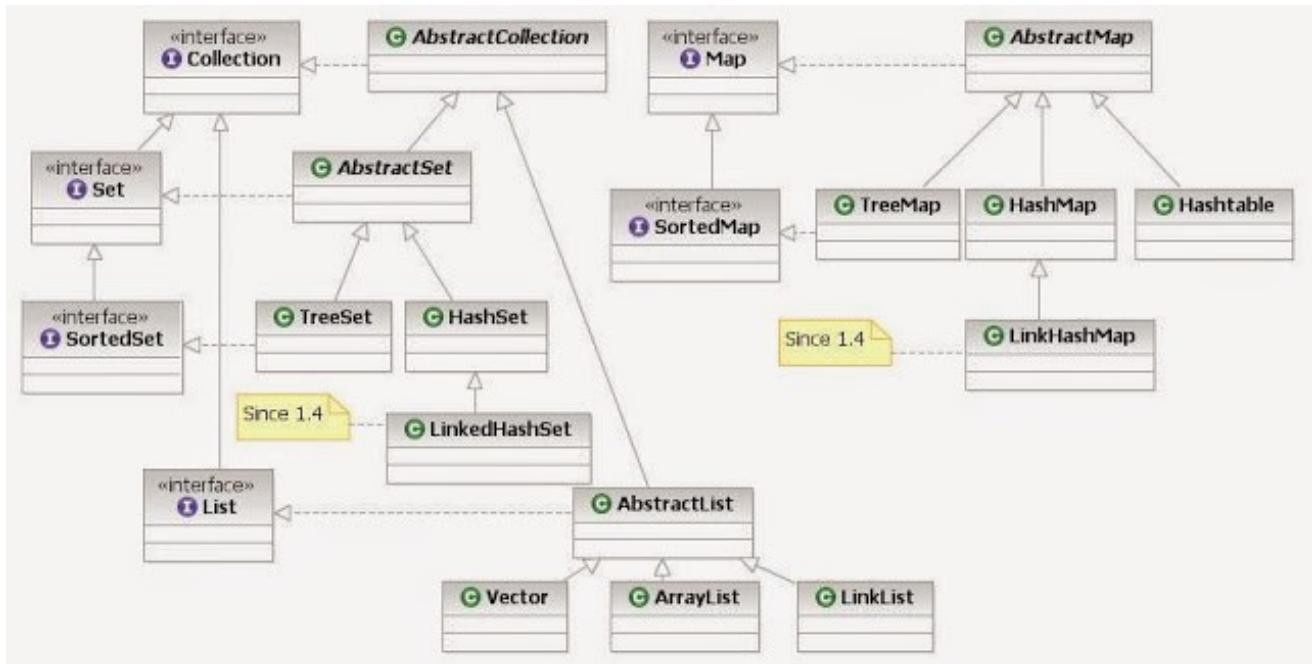
## Q3 What is the difference between Collection and Collections ?

Collection is an interface while Collections is a java class , both are present in java.util package and part of java collections framework.

## Q4 Which collection classes are synchronized or thread-safe ?

Stack, Properties , Vector and Hashtable can be used in multi threaded environment because they are synchronized classes (or thread-safe).

## Q5 Name the core Collection interfaces ?



The list of core collection interfaces are : just mention the important ones

Important : Collection , Set , Queue , List , Map

Other interface also in the list : SortedSet, SortedMap , Deque, ListIterator etc.

Q6 What is the difference between List and Set ?

Set contain only unique elements while List can contain duplicate elements.

Set is unordered while List is ordered . List maintains the order in which the objects are added .

Q7 What is the difference between Map and Set ?

Map object has unique keys each containing some value, while Set contain only unique values.

Q8 What are the classes implementing List and Set interface ?

*Class implementing List interface :* ArrayList , Vector , LinkedList ,

*Class implementing Set interface :* HashSet , TreeSet

Q9 What is an iterator ?

Iterator is an interface . It is found in java.util package. It provides methods to iterate over any Collection.

Q10 What is the difference between Iterator and Enumeration ?

The main difference between Iterator and Enumeration is that Iterator has remove() method while Enumeration doesn't.

Hence , using Iterator we can manipulate objects by adding and removing the objects from the collections. Enumeration behaves like a read only interface as it can only traverse the objects and fetch it .

Q11 Which design pattern followed by Iterator ?

It follows iterator design pattern. Iterator design pattern provides us to navigate through the collection of objects by using a common interface without letting us know about the underlying implementation.

Enumeration is an example of Iterator design pattern.

Q12 Which methods you need to override to use any object as key in HashMap ?

To use any object as key in HashMap , it needs to implement equals() and hashCode() method .

Q13 What is the difference between Queue and Stack ?

Queue is a data structure which is based on FIFO ( first in first out ) property . An example of Queue in real world is buying movie tickets in the multiplex or cinema theaters.

Stack is a data structure which is based on LIFO (last in first out) property . An example of Stack in real world is insertion or removal of CD from the CD case.

Q14 How to reverse the List in Collections ?

There is a built in reverse method in Collections class . reverse(List list) accepts list as parameter.

Collections.reverse(listobject);

Q15 How to convert the array of strings into the list ?

Arrays class of java.util package contains the method asList() which accepts the array as parameter.

So,

```
String[] wordArray = {"Love Yourself" , "Alive is Awesome" , "Be in present"};
```

```
List wordList = Arrays.asList(wordArray);
```

*Intermediate Level (1-3 yrs): Java Collections Interview Questions and Answers*

Q16 What is the difference between ArrayList and Vector ?

It is one of the frequently asked collection interview question , the main differences are Vector is synchronized while ArrayList is not . Vector is slow while ArrayList is fast . Every time when needed, Vector increases the capacity twice of its initial size while ArrayList increases its ArraySize by 50%. find detailed explanation [ArrayList vs Vector](#) .

Q17 What is the difference between HashMap and Hashtable ?

It is one of the most popular collections interview question for java developer . Make sure you go through this once before appearing for the interview .

Main differences between HashMap and Hashtable are :

- a. HashMap allows one null key and any number of null values while Hashtable does not allow null keys and null values.
- b. HashMap is not synchronized or thread-safe while Hashtable is synchronized or thread-safe .

find detailed explanation here [Hashtable vs HashMap in Java](#)

Q18 What is the difference between peek(),poll() and remove() method of the Queue interface ?

Both poll() and remove() method is used to remove head object of the Queue. The main difference lies when the Queue is empty().

If Queue is empty then poll() method will return null . While in similar case , remove() method will throw NoSuchElementException .

peek() method retrieves but does not remove the head of the Queue. If queue is empty then peek() method also returns null.

Q19 What is the difference between Iterator and ListIterator.

Using Iterator we can traverse the list of objects in forward direction . But ListIterator can traverse the collection in both directions that is forward as well as backward.

Q20 What is the difference between Array and ArrayList in Java ?

This question checks whether student understand the concept of static and dynamic array.

Some main differences between Array and ArrayList are :

- a. Array is static in size while ArrayList is dynamic in size.
- b. Array can contain primitive data types while ArrayList can not contain primitive data types.

find detailed explanation [ArrayList vs Array in Java](#)

Q21 What is the difference between HashSet and TreeSet ?

Main differences between HashSet and TreeSet are :

- a. HashSet maintains the inserted elements in random order while TreeSet maintains elements in the sorted order
- b. HashSet can store null object while TreeSet can not store null object.

find detailed explanation here [TreeSet vs HashSet in Java](#)

Q22 Write java code showing insertion,deletion and retrieval of HashMap object ?

Do it yourself (DIY) , if found any difficulty or doubts then please mention in the comments.

Q23 What is the difference between HashMap and ConcurrentHashMap ?

This is also one of the most popular java collections interview question . Make sure this question is in your to do list before appearing for the interview .

Main differences between HashMap and ConcurrentHashMap are :

- a. HashMap is not synchronized while ConcurrentHashMap is synchronized.
- b. HashMap can have one null key and any number of null values while ConcurrentHashMap does not allow null keys and null values .

find detailed explanation here [ConcurrentHashMap vs HashMap in Java](#)

Q24 Arrange the following in the ascending order (performance):

HashMap , Hashtable , ConcurrentHashMap and Collections.SynchronizedMap

Hashtable < Collections.SynchronizedMap < ConcurrentHashMap < HashMap

Q25 How HashMap works in Java ?

This is one of the most important question for java developers. HashMap works on the principle of Hashing . Find detailed information here to understand [what is hashing and how hashmap works in java](#) .

Q26 What is the difference between LinkedList and ArrayList in Java ?

Main differences between LinkedList and ArrayList are :

- LinkedList is the doubly linked list implementation of list interface , while , ArrayList is the resizable array implementation of list interface.
- LinkedList can be traversed in the reverse direction using descendingIterator() method provided by the Java Api developers , while , we need to implement our own method to traverse ArrayList in the reverse direction . find the detailed explanation here [ArrayList vs LinkedList in java](#).

Q27 What are Comparable and Comparator interfaces ? List the difference between them ? We already explained what is comparable and comparator interface in detail along with examples here, [Comparable vs Comparator in Java](#)

Q28 Why Map interface does not extend the Collection interface in Java Collections Framework ?

One liner answer : Map interface is not compatible with the Collection interface.

Explanation : Since Map requires key as well as value , for example , if we want to add key-value pair then we will use put(Object key , Object value) . So there are two parameters required to add element to the HashMap object . In Collection interface add(Object o) has only one parameter.

The other reasons are Map supports valueSet , keySet as well as other appropriate methods which have just different views from the Collection interface.

Q29 When to use ArrayList and when to use LinkedList in application?

ArrayList has constant time search operation  $O(1)$  .Hence, ArrayList is preferred when there are more get() or search operation .

Insertion , Deletion operations take constant time  $O(1)$  for LinkedList. Hence, LinkedList is preferred when there are more insertions or deletions involved in the application.

Q30 Write the code for iterating the list in different ways in java ?

There are two ways to iterate over the list in java :

- using Iterator
- using for-each loop

Coding part : Do it yourself (DIY) , in case of any doubts or difficulty please mention in the comments .

*Advance Level (3+ yrs): Java Collections Interview Questions and Answers*

Q31 How HashSet works internally in java ?

This is one of the popular interview question . HashSet internally uses HashMap to maintain the uniqueness of elements. We have already discussed in detail [hashset internal working in java](#).

Q32 What is CopyOnWriteArrayList ? How it is different from ArrayList in Java?

[CopyOnWriteArrayList](#) is a thread safe variant of ArrayList in which all mutative operations like add , set are implemented by creating a fresh copy of the underlying array.

It guaranteed not to throw ConcurrentModificationException.

It permits all elements including null. It is introduced in jdk 1.5 .

Q33 How HashMap works in Java ?

We are repeating this question , as it is one of the most important question for java developer. HashMap works on the principle of Hashing . please find the detailed answer here [hashmap internal working in java](#) .

Q34 How remove(key) method works in HashMap ?

This is the new question which is getting popular among java interviewers . We have shared the detailed explanation here [how remove method works internally in java](#).

Q35 What is BlockingQueue in Java Collections Framework?

[BlockingQueue](#) implements the `java.util.Queue` interface . BlockingQueue supports operations that wait for the queue to become non-empty when retrieving an element , and wait for space to become available in the queue when storing an element .

It does not accept null elements.

Blocking queues are primarily designed for the producer-consumer problems.

BlockingQueue implementations are thread-safe and can also be used in inter-thread communications.

This concurrent Collection class was added in jdk 1.5

Q36 How TreeMap works in Java ?

TreeMap internally uses Red-Black tree to sort the elements in natural order. Please find the detailed answers here [internal implementation of TreeMap in java](#) .

Q37 All the questions related to HashSet class can be found here , [frequently asked HashSet interview questions](#)

Q38 What is the difference between Fail- fast iterator and Fail-safe iterator ?

This is one of the most popular interview question for the higher experienced java developers .

Main differences between Fail-fast and Fail-safe iterators are :

- Fail-fast throw `ConcurrentModificationException` while Fail-safe does not.
- Fail-fast does not clone the original collection list of objects while Fail-safe creates a copy of the original collection list of objects.

The difference is explained in detail here [fail-safe vs fail-fast iterator in java](#).

Q39 How ConcurrentHashMap works internally in Java?

The detailed answer is already explained here [internal implementation of concurrenthashmap](#)

Q40 How do you use a custom object as key in Collection classes like HashMap ?

If one is using the custom object as key then one needs to override `equals()` and `hashCode()` method and one also need to fulfill the contract. If you want to store the custom object in the SortedCollections like `SortedMap` then one needs to make sure that `equals()` method is consistent to the `compareTo()` method. If inconsistent , then collection will not follow their contracts ,that is , Sets may allow duplicate elements.

Q41 What is hash-collision in Hashtable ? How it was handled in Java?

In Hashtable , if two different keys have the same hash value then it lead to hash -collision. A bucket of type linkedlist used to hold the different keys of same hash value.

Q42 Explain the importance of `hashCode()` and `equals()` method ? Explain the contract also ?

HashMap object uses Key object `hashCode()` method and `equals()` method to find out the index to put the key-value pair. If we want to get value from the HashMap same both methods are used . Somehow, if both methods are not implemented correctly , it will result in two keys producing the same `hashCode()` and `equals()` output. The problem will arise that HashMap will treat both output same instead of different and overwrite the most recent key-value pair with the previous key-value pair.

Similarly all the collection classes that does not allow the duplicate values use `hashCode()` and `equals()` method to find the duplicate elements. So it is very important to implement them correctly.

Contract of `hashCode()` and `equals()` method

- If `object1.equals(object2)` , then `object1.hashCode() == object2.hashCode()` should always be true.
- If `object1.hashCode() == object2.hashCode()` is true does not guarantee `object1.equals(object2)`



Q43 What is EnumSet in Java ?

[EnumSet](#) is a specialized Set implementation for use with enum types. All of the elements in an enum set must come from a single enum type that is specified explicitly or implicitly, when the set is created.

The iterator never throws ConcurrentModificationException and is weakly consistent.

Advantage over HashSet:

All basic operations of EnumSet execute in constant time. It is most likely to be much faster than HashSet counterparts.

It is a part of Java Collections Framework since jdk 1.5.

Q44 What are concurrentCollectionClasses?

In jdk1.5, Java Api developers had introduced new package called java.util.concurrent that have thread-safe collection classes as they allow collections to be modified while iterating. The iterator is fail-fast that is it will throw ConcurrentModificationException.

Some examples of concurrentCollectionClasses are :

a. CopyOnWriteArrayList

b. ConcurrentHashMap

Q45 How do you convert a given Collection to SynchronizedCollection ?

One line code : Collections.synchronizedCollection(Collection collectionObj) will convert a given collection to synchronized collection.

Q46 What is IdentityHashMap ?

IdentityHashMap

[IdentityHashMap](#) is a class present in java.util package. It implements the Map interface with a hash table, using [reference equality instead of object equality](#) when comparing keys and values. In other words, in IdentityHashMap two keys k1 and k2 are considered equal if only if (k1==k2).

IdentityHashMap is not synchronized.

Iterators returned by the iterator() method are fail-fast, hence, will throw ConcurrentModificationException.

Q47 What is WeakHashMap ?

WeakHashMap :

[WeakHashMap](#) is a class present in java.util package similar to IdentityHashMap. It is a Hashtable based implementation of Map interface with weak keys. An entry in WeakHashMap will automatically be removed when its key is no longer in ordinary use. More precisely the presence of a mapping for a given key will not prevent the key from being discarded by the garbage collector.

It permits null keys and null values.

Like most collection classes this class is not synchronized. A synchronized WeakHashMap may be constructed using the Collections.synchronizedMap() method.

Iterators returned by the iterator() method are fail-fast, hence, will throw ConcurrentModificationException.

Q48 How will you make Collections readOnly

We can make the Collection readOnly by using the following lines code:

General : Collections.unmodifiableCollection(Collection c)

Collections.unmodifiableMap(Map m)

Collections.unmodifiableList(List l)

Collections.unmodifiableSet(Set s)

#### Q49 What is UnsupportedOperationException?

This exception is thrown to indicate that the requested operation is not supported.

Example of UnsupportedOperationException:

In other words, if you call add() or remove() method on the readOnly collection . We know readOnly collection can not be modified . Hence , UnsupportedOperationException will be thrown.

Q50 Suppose there is an Employee class. We add Employee class objects to the ArrayList. Mention the steps need to be taken , if I want to sort the objects in ArrayList using the employeeId attribute present in Employee class.

- a. Implement the Comparable interface for the Employee class and now to compare the objects by employeeId we will override the emp1.compareTo(emp2)
- b. We will now call Collections class sort method and pass the list as argument , that is , Collections.sort(empList)

If you want to add more java collections interview questions and answers or in case you have any doubts related to the Java Collections framework , then please mention in the comments.