

# Why we Need Collection classes? Page-1

→ Today we will learn why collection classes are necessary in java language.

→ Collection Classes are known as Collection Framework.

→ It is very important for Java language interviews.

Like an array - which can hold only homogeneous data elements, that means, if we declare an array of Students type then it can store only Students type.

eg. `Students s = new Students[10];`  
`s[0] = new Students();`  
`s[1] = new Students();`

## Limitations with arrays.

- ① Fixed Size declaration
- ② Only Homogeneous data storage
- ③ There is no underlying data structure. (which is ready-made defined)

To overcome these problems - we should go for Collections

- ✓ ① Collections are growable in nature. that is, we can increase or decrease the size as needed.
- ✓ ② Collections can hold both homogeneous and the heterogeneous objects (elements).
- ✓ ③ Every Collection class is implemented by using some standard data structure. Hence, ready-made method support is available for every requirement. Being a programmer, we have to use this method and we are not responsible to provide implementation.

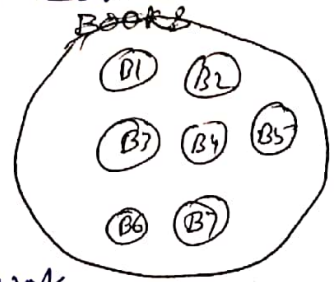
Arrays

- ① Fixed in Size
- ② With respect to memory, arrays are not recommended to use.
- ③ With respect to performance, arrays are recommended to use.
- ④ Only homogeneous elements can be stored.
- ⑤ Underlying data structure is not available i.e; readymade method support is not available.
- ⑥ In array, we can hold primitives and objects both.

Collections

- ① Growable in nature
- ② With respect to memory, Collections are recommended to use.
- ③ With respect to performance, Collections are not recommended to use.
- ④ Both-Homogeneous and Heterogeneous elements are stored.
- ⑤ For every Collections class, there are standard readymade methods supports defined in its Standard data structure.
- ⑥ By Using collection class, we can hold only objects, but not primitives.

Collection :- A group of individual objects.  
 If we want to represent a group of individual objects as a single entity then we should go for Collection.



Collection Framework :-

Several classes and Interfaces which can be used a group of objects as single entity is called Collection Framework.

Java	C++
Collection Framework	Container
	STL (Standard Template Library)

# 9 Key Interfaces of Collection

① Collection(I) - Root interface of Collection framework. It defines most basic <sup>common</sup> methods that can be used by implementing classes (applicable for any Collection object.)

→ There is no concrete class which implements Collection interface directly.

Difference between Collection and Collections :-

Interview Question

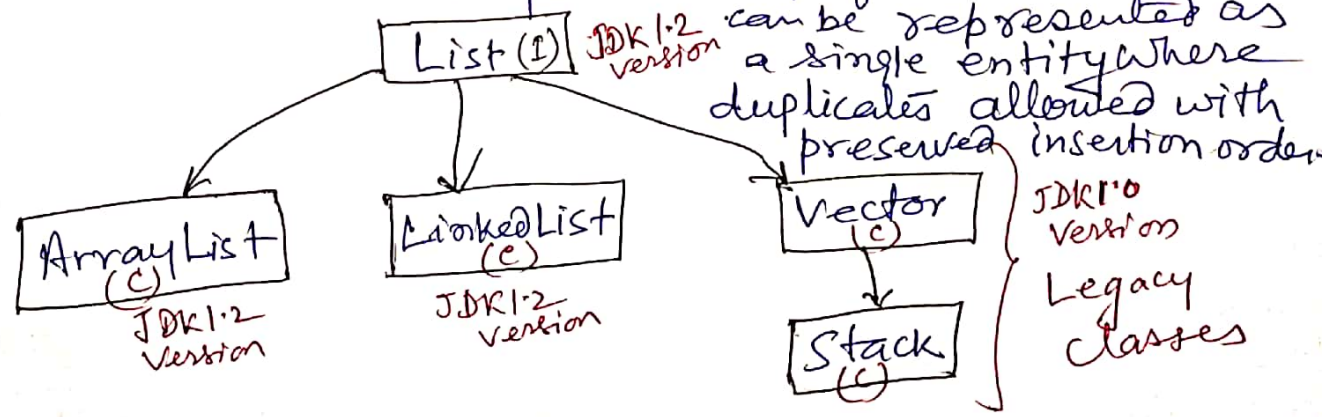
## Collection (I)

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

## Collections (C)

- It is class present in java.util package to define several utility methods. (like Sorting, Searching....) for Collection Objects.

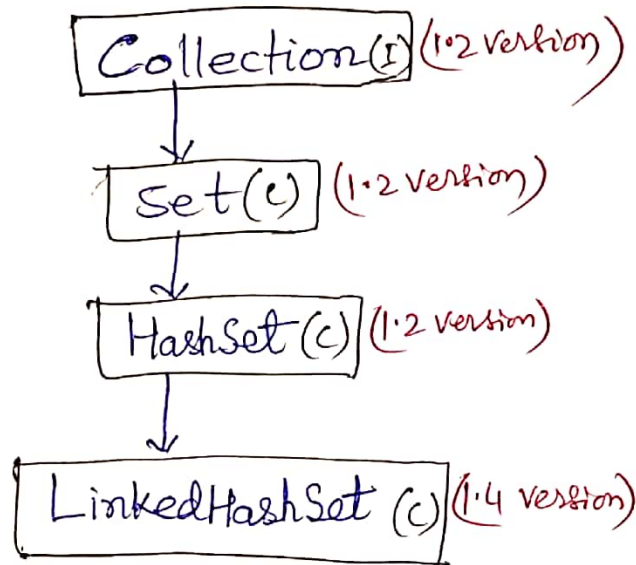
② List (I) - It is child interface of Collection interface. Group of individual objects



Legacy classes - The classes which are coming from older version are known as legacy classes

② List(I) :- In other words, List is used to represent group of individual objects as a single entity where duplicates are allowed and insertion order preserved.  
Then we should go for List.

③ Set(I) :- It is a child interface of Collection interface where duplicates are not allowed and insertion order not preserved.

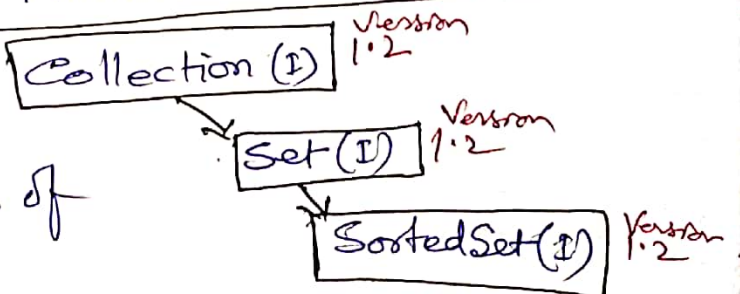


Interview Question

Difference between List and Set :-

List	Set
① Duplicates are allowed.	① Duplicates are not allowed.
② Insertion order Preserved	② Insertion order NOT Preserved.

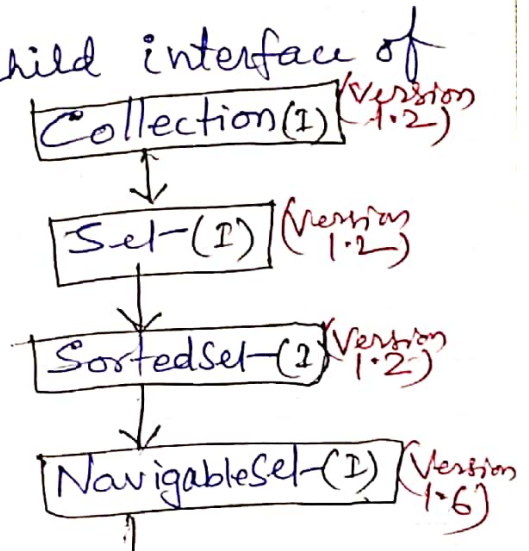
④ SortedSet(I) :-  
It is a child interface of Set.



If we want to represent a group of individual objects as a single entity where duplicates are not allowed but all objects should be inserted according to some sorting order then we should go for SortedSet.

⑤ NavigableSet(I) :- It is child interface of SortedSet. It defines several methods for navigation purposes.

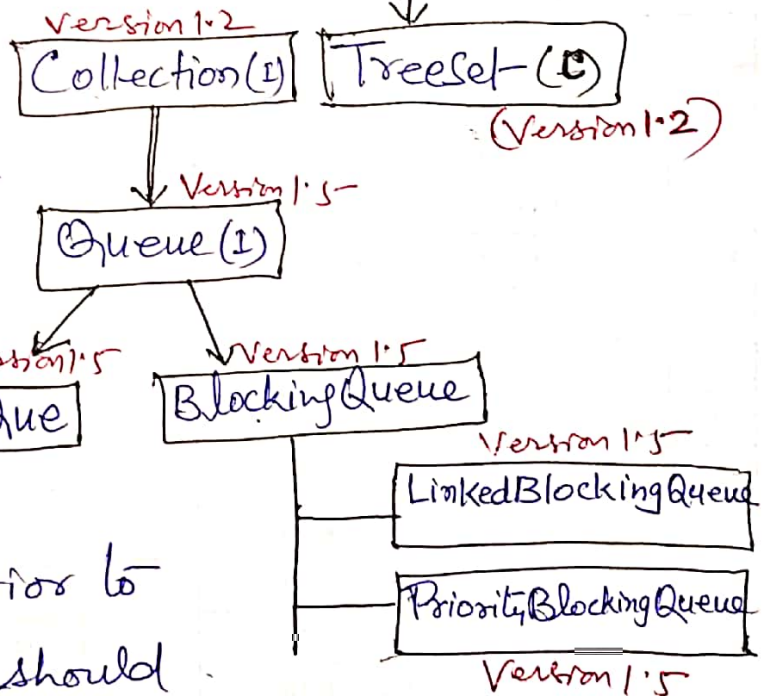
→ TreeSet is a class implementing this interface.



⑥ Queue(I) :-

It is a child interface of Collection.

If we want to represent a group of individual objects prior to processing then we should go for Queue - First In First Out sequence.



These 6 interfaces are used for group of objects - (individual objects).

If we want to represent a group of objects as key-value pairs then we should go for Map Interface.

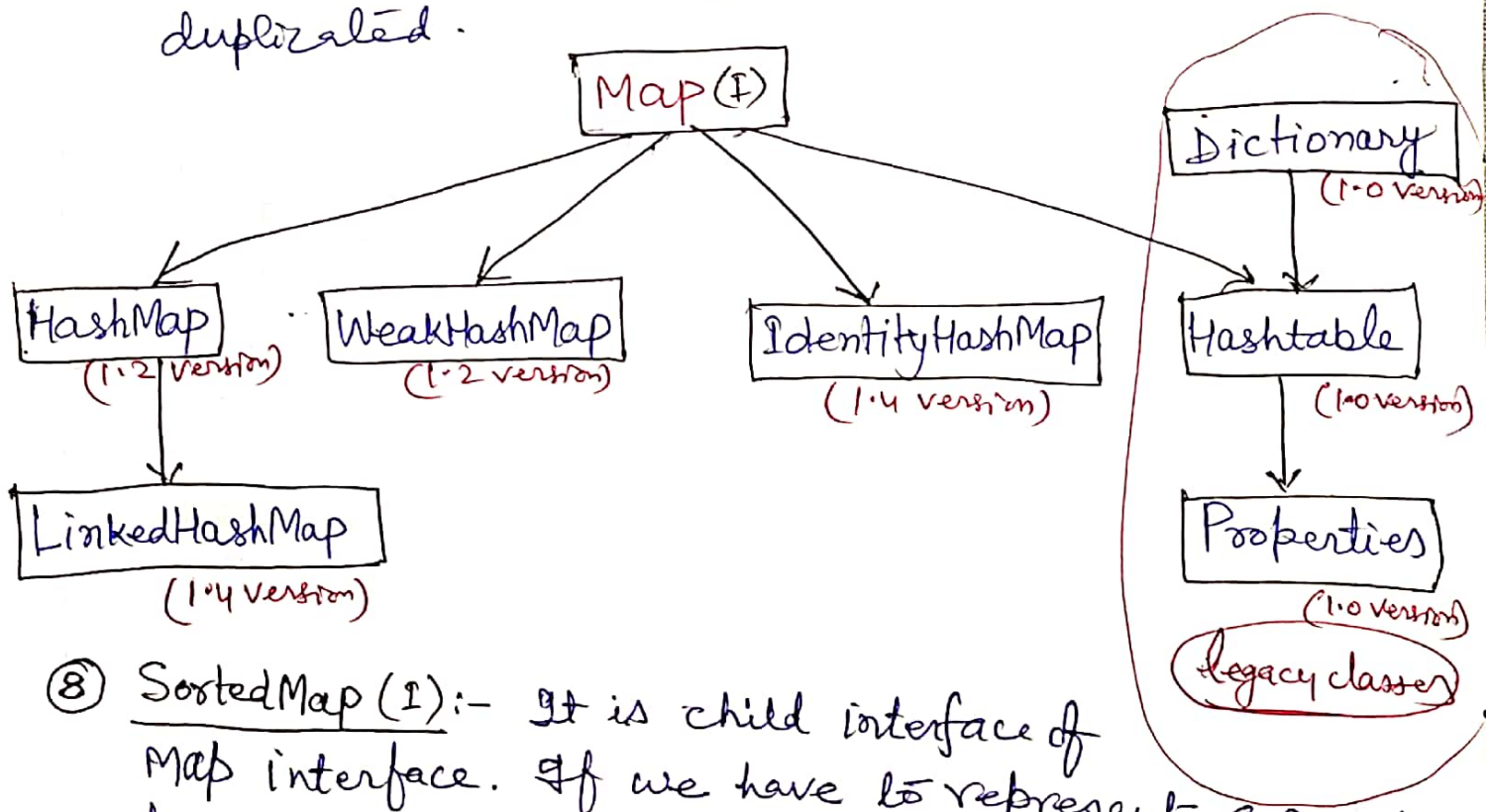
7. Map(I) - Map is not a child-interface of Collection.

~~Map~~

If we want to represent a group of individual objects as key value then we have to use Map

~~HashMap~~ ~~WeakHashMap~~

ex: Roll - Name  
Both key and value are objects, duplicates are not allowed for keys but values can be duplicated.



8. SortedMap(I):- It is child interface of Map interface. If we have to represent a group of key value pairs in some sorting order of keys then we have to use SortedMap.

Map (I) (1.2 version)

SortedMap (I) (1.2 version)

⑨ NavigableMap (I) :-

It is a child interface of SortedMap. It provides several methods for navigation purpose.

