

Collections

Data – collection of values

Collections deals with arranging the data inside the main memory. That data will be used by java program.

Array is a basic data structure or basic collection available in Java but it has its limitation.

The limitation of an array is always size because the size is always a fixed size. We cannot increase the same size array.

Create a bigger size array

```
Int A[] = {2,4,10,12,16}
```

```
Int B[]=new int[10];
```

```
For (int i=0; i<A; i++) {
```

```
    B[i]=A[i];
```

```
}
```

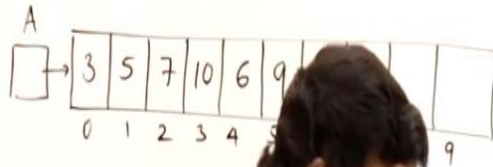
```
A= B;
```

```
Int A[]={2,4,10,12,16}
```

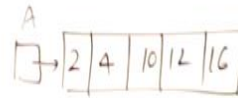
```
Int B[]={2,4,10,12,16,24,32,48,54} // Bigger size of array
```

Collection Framework

`int A[] = int A[10];`



`int A[] = {2, 4, 10, 12, 16};`

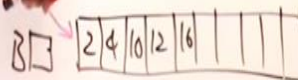
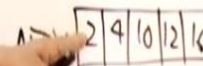


`→ int B[] = new int[10];`

`for (int i = 0; i < A.length; i++)`
`{`

`B[i] = A[i];`

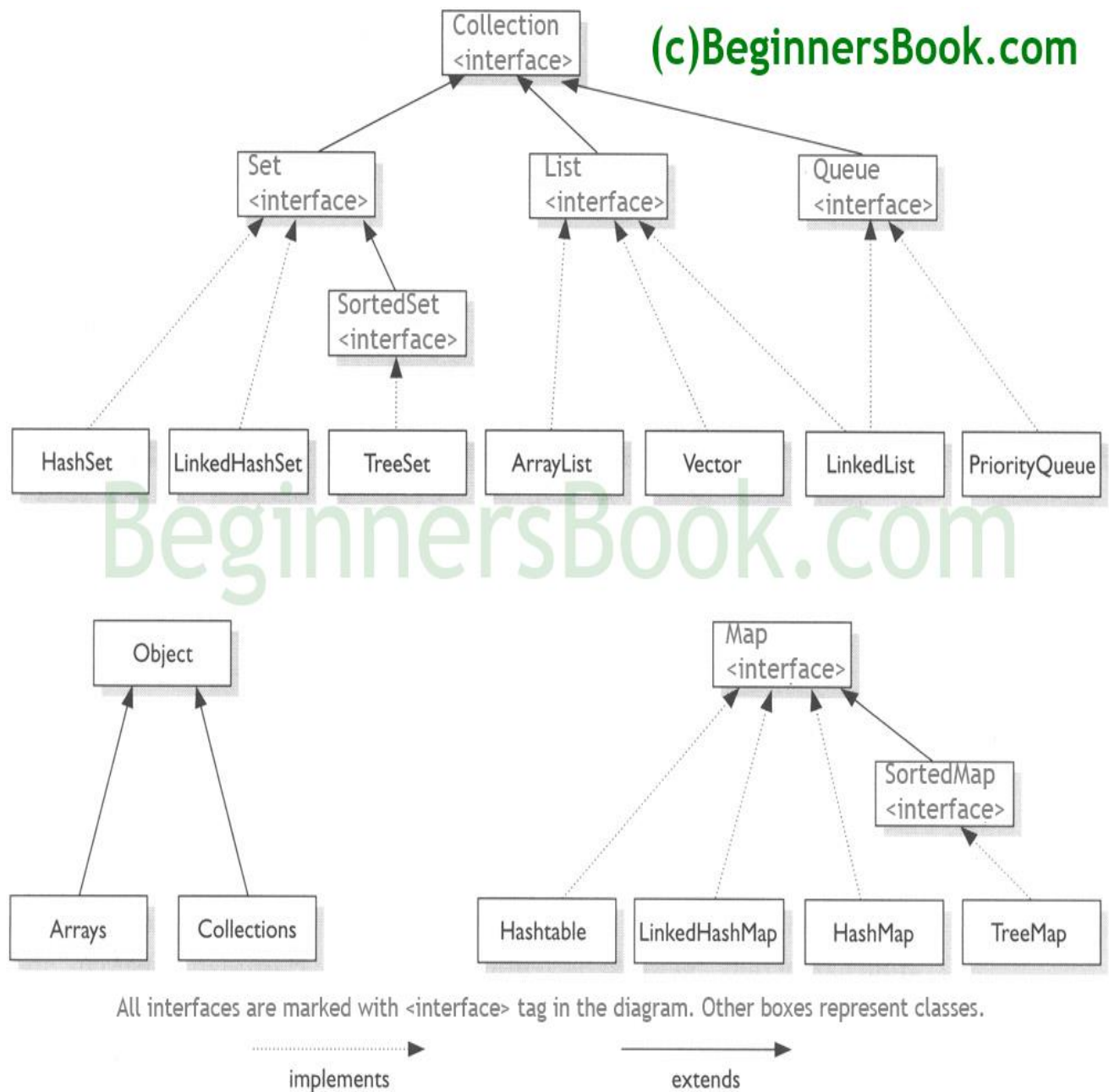
`A = B; B = null;`



`A=B; /*or we can say*/ B=null; // B will be not pointing on bigger array`

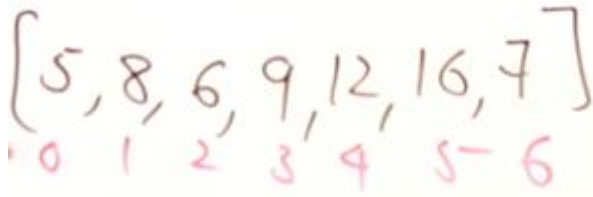
Now array A is pointing on bigger array.

This is the way we can increase the size of array. So, the same array we cannot be increased or decreased. So, if we want to increase or decrease, we have to create another array and shift all the objects in that new array.



- Collections- It's a group of elements or items or group of objects.

- **List** is a collection but it's the ordered list of elements means the elements will have a number.



Handwritten text showing a list of numbers in brackets: [5, 8, 6, 9, 12, 16, 7]. Below each number is its corresponding index: 0, 1, 2, 3, 4, 5, 6.

Index	Element
0	5
1	8
2	6
3	9
4	12
5	16
6	7

0, 1, 2, 3, 4, 5, 6. So, they have index. They are ordered list of elements.

What it's benefit if they have ordered?

If they are ordered then we can insert or add an element or delete an element based on the indices. We can't do it in collection and List can have duplicate elements.

- List of the classes: -
- ArrayList
- LinkedList
- Vector
- Stack

Vector and Stack are called as legacy classes. These are not much use.

- **Set**- List and Set both are collections but Set will not have duplicates. It will not allow us to insert duplicate objects. And second thing is the elements of the collection of elements are not ordered. So, it's an unordered collection, but it will have unique objects.

- Benefits?

When we want to perform search operation frequently. So, for searching, we prefer that there are no duplicates in the list.

- From Set there is one more interface that is Sorted set.

It's a Set but the elements are sorted out.

- The class which is coming out form Set is HashSet and LinkedHashSet.

Hash code is using for storing the elements in this class.

- So, from SortedSet there is a class that is implementing this interfaces TreeSet.

- Benefit of SortedSet is that we can search for the element faster and tree data structure is used for faster searching.

- **Queue**- works on discipline FIFO. It means we can insert the elements in the collection from one end and delete from another end. Suppose insertion done in these [5, 6, 9, 12, 16, 7]
If I have to add one more element, I may be adding it here.

[5, 6, 9, 12, 16, 7, 15]

Now, if I am deleting from here then I have to delete from front (5).

[6, 9, 12, 16, 7, 15]

[6, 9, 12, 16, 7, 15]

So, first 5 will be deleted then 6 will be deleted.

- There is a class that's implementing from there Priority Queue.
- Deque is another interface – This insertion and deletion is allowed from both the ends and there is one class that's implementing the interface that's ArrayDeque.

- **Map**- This is the implementing of Hashing Techniques.
So there the objects are inserted or stored in the form of a key and value pair. So, whenever we want to search, we can give a key. And we can get the value that corresponding to that key.

Classes- HashMap, LinkedHashMap, HashTable.

SortedMap[interface] faster -> TreeMap.

