HashMap

1. It is used to hold **key value pairs**.
2. **Data Structure** is **HashTable.**
3. Multiple threads can operate simultaneously on **HashMap Object** and Hence it is **not thread safe.**
4. Can update the **value** using **key** again
5. **Values** can be duplicated.
6. Null is allowed for both **key(Only Once)** and **value(No of Times)**
7. It is non legacy and introduced in 1.2v.
8. **Insertion Order** is preserved.

values()

values() returns a collection view of the **values** contained in the map.

keyset()

returns the set of **keys** containe in the map.

```
//It will order all the elements in the form of keys
Map<Integer, String> hashMap = new HashMap<Integer, String>();
hashMap.put(5, "NameFive");
hashMap.put(3, "NameThree");
hashMap.put(1, "NameOne");
hashMap.put(4, "NameFour");
hashMap.put(2, "NameTwo");
hashMap.put(2, "NameTwoDuplicate"); //Can update the value using key again
hashMap.put(6, "NameFive"); // Values are duplicated
hashMap.put(8, null); // null allowed multiple times
hashMap.put(7, null); // // null allowed multiple times
hashMap.put(null, "null");
hashMap.put(null, "null"); //Duplicate null key and null value
System.out.println(hashMap); // {null=null, 1=NameOne, 2=NameTwoDuplicate, 3=NameThree, 4=NameFour, 5=NameFive,
6=NameFive, 7=null, 8=null}
```

```
HashMap<Integer, String> hm = new HashMap<Integer, String>();
hm.put(01, "NameOne");
hm.put(02, "NameTwo");
hm.put(03, "NameThree");
hm.put(04, "NameFour");
hm.put(05, null);
// returns the number of entries in the map.
System.out.println(hm.size()); // 5

System.out.println(hm); // {1=NameOne, 2=NameTwo, 3=NameThree, 4=NameFour, 5=null}
hm.remove(3);
hm.remove(2, "Adv Java"); // {1=NameOne, 2=NameTwo, 4=NameFour, 5=null}
System.out.println(hm);
hm.remove(4, "NameFour");
System.out.println(hm); // {1=NameOne, 2=NameTwo, 4=NameFour, 5=null}
hm.clear();
System.out.println(hm); // {}
```

```
HashMap<Integer, String> hm = new HashMap<Integer, String>();
hm.put(1, "Core Java");
hm.put(2, "Adv Java");
hm.put(3, "Hibernate");
hm.put(5, "Rest");
hm.put(4, "Spring");

// values(): Returns a Collection view of the values contained in this map.
Collection<String> values = hm.values();
System.out.println("Set of Values: " + values); //Set of Values: [Core Java, Adv Java, Hibernate, Spring, Rest]

// keySet(): Returns a Set view of the keys contained in this map.
Set<Integer> keySet = hm.keySet();
System.out.println("Set of Keys: " + keySet); // Set of Keys: [1, 2, 3, 4, 5]
```