

**What is Hashmap in java**

* Hashmap is the implementation of Map interface which stores the data in key value pair
* Duplicate key is not allowed but values are allowed to be duplicate
* It has 4 values key, value, hash and next value
* Put() calls hashcode() which gives you integer value on which hashing will be performed with module technique by taking the remainder between 0 – 15 (index).
* Once we get the index equal() will check whether we have the same key on that index if yes then it will simply replace the value or it will add the new node in linked list format

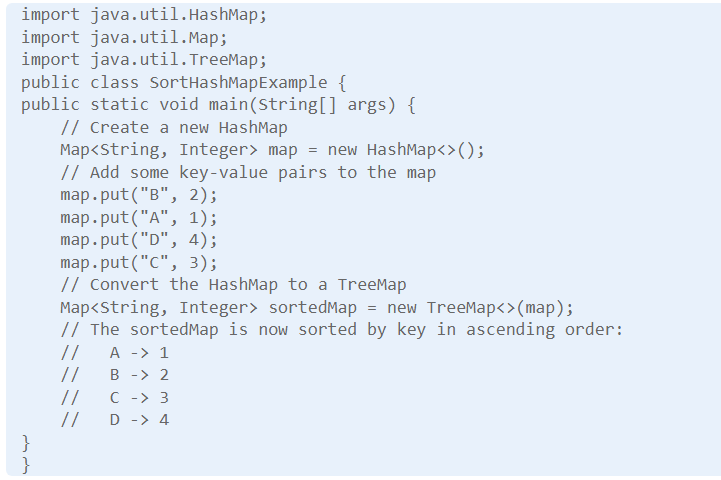
**What is collision in HashMap?**

* A collision occurs when more than one key generates the same hashCode() value.
* An inefficient hashCode() algorithm causes considerable collisions in a hashmap.
* The collision is resolved by adding the element in linked list format.
* An increased number of collisions can affect the performance of a hashmap.

**Can a map be sorted in java?**

* There is no particular order for storing keys in a hashmap. All the keys are stored in an unsorted order.
* One way to sort a HashMap is to first convert it to a TreeMap, which is a sorted map data structure. I can do this by using the putAll method to add the key-value pairs from the HashMap to a new TreeMap.

e.g



**Is Hashmap thread-safe in java?**

* The HashMap class in Java is not a thread-safe method.
* If one or more threads modify the HashMap (e.g., inserting or removing a map) we will face concurrent modification exception.
* But ConcurrentHashMap is thread-safe as it has locking mechanism on segment level.

**What is the load factor in HashMap**

A load factor is a number that controls the resizing of HashMap when a number of elements in the HashMap cross the load factor as if the load factor is 0.75 and when becoming more than 75% full then resizing trigger which involves array copy.

**How does resize happens in HashMap?**

The resizing happens when the map becomes full or when the size of the map crosses the load factor. For example, if the load factor is 0.75 and then becomes more than 75% full, then resizing trigger, which involves an array copy. First, the size of the bucket is doubled, and then old entries are copied into a new bucket.

**Changes in java 8 for HashMap.**

* Even though HashMap represents a hash table, it is internally implemented by using an array and linked list data structure in JDK.  The array data structure is used as a bucket, while a [linked list](http://javarevisited.blogspot.sg/2017/07/top-10-linked-list-coding-questions-and.html#axzz4qw7RoNvw) is used to store all mappings which land in the same bucket. From Java 8 onwards, the linked list is dynamically replaced by [binary search tree](https://javarevisited.blogspot.com/2015/10/how-to-implement-binary-search-tree-in-java-example.html), once a number of elements in the linked list cross a certain threshold to improve performance.

### Can you store a duplicate key in HashMap?

* No, you cannot insert duplicate keys in HashMap, it doesn't allow duplicate keys. If you try to insert an existing key with the new or same value, then it will override the old value, but the size of HashMap will not change i.e., it will remain the same. This is one of the reasons when you get all keys from the HashMap by calling [keySet()](http://www.java67.com/2016/05/keyset-vs-entryset-vs-values-in-java-map-example.html" \t "_blank). It returns a Set, not a Collection because Set doesn't allow duplicates.

**What are the different ways to iterate over HashMap in Java?**

Here are some of the ways to iterate over HashMap in Java:

* by using keySet and iterator
* by using entrySet and iterator
* by using entrySet and enhanced for loop
* by using keySet and get() method