**Difference between Hashtable and HashMap or Hashtable vs HashMap**

1. In terms of Null keys and Null values:
   1. Hashtable does not allows null keys or values. If null keys or values tried to be inserted into the table then it throws java.lang.NullPointerException.
   2. HashMap allows multiple null values but single null key. Why single null keys? The reason is that if the key is same then the old value gets replaced with the new value.
2. Iterating the Map:
   1. Hashtable uses Enumerator to iterate through the table.
   2. HashMap uses iterator to iterate through the table.
3. Thread safety:
   1. Hashtable is Synchronized and hence it is thread safe. It can be used in multithreading environment. Hashtable locks entire table to achieve synchronization. Better option for concurrency in map is ConcurrentHashMap as it does not lock entire table.
   2. HashMap is not synchronized so it is not thread safe and hence cannot be used in multithreaded environment. As it is not synchronized it is faster than Hashtable or ConcurrentHashMap. Also HashMap should be used in such environment which guarantees that the HashMap object will be not shared among different threads.
4. Performance:
   1. Hastable is slow as it is synchronized.
   2. HashMap is faster than Hashtable as it is not synchronized and no overhead of been shared by different thread.
5. Iterator in HashMap vs Enumerator in Hashtable:
   1. Enumerator in Hashtable are not fail fast.
   2. Iterator in HashMap are fail fast. Means if the structure of Map is modified except the iterator’s own remove() method then it will throw ConcurrentModificationException.

Hashtable should not be used as it locks entire table. We can use ConcurrentHashMap as it does not lock entire table. It locks only the specific index in table which allows other thread to work with other index entries on map simultaneously.