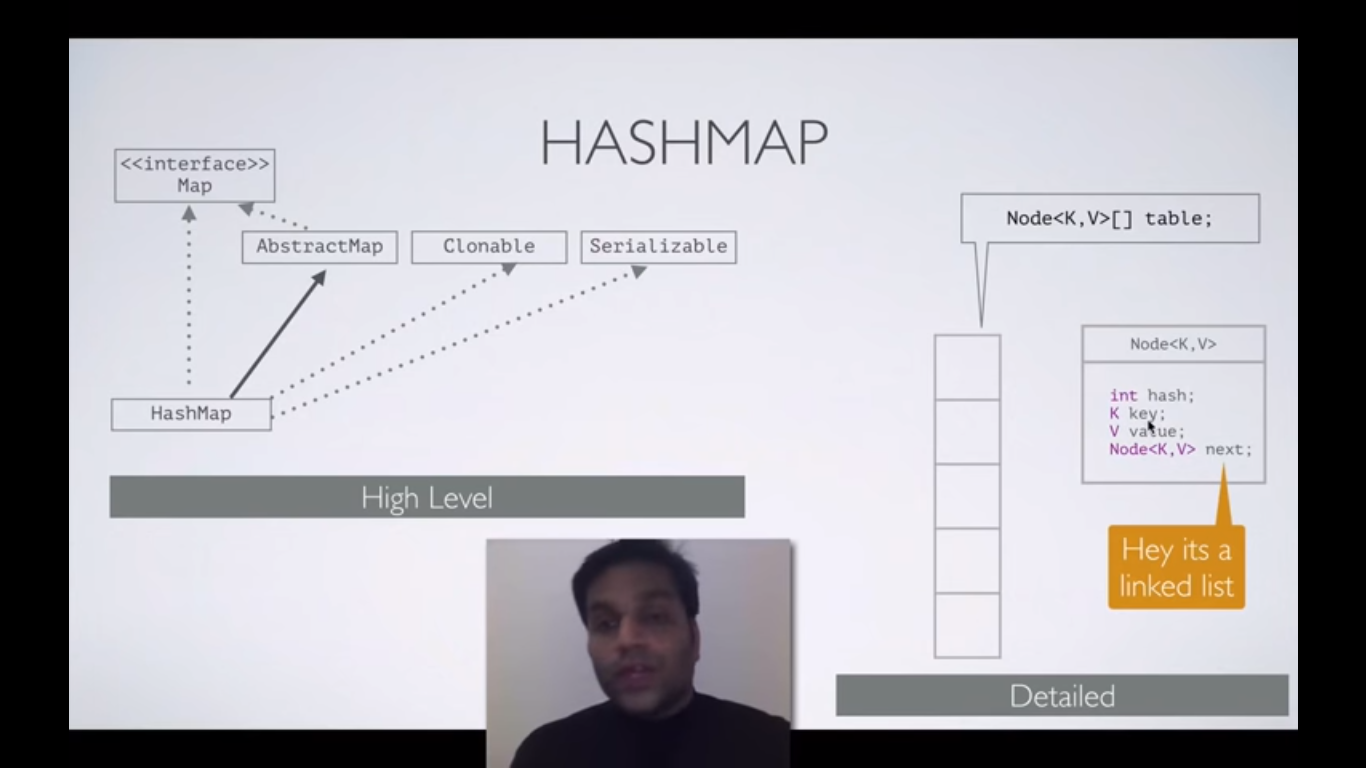
**How HashMap Works in java Internally.**

Hash map:

1. is provided in java.util package.
2. Mainly uses key and values to store and retrieve the data.
3. It’s implementation of map interface.
4. It’s follows technic called hashing (transforms or convert large string into a small/fixed length object).
5. In java every object has hash value, which is generated from **public int hashCode()** method.
6. If 2 objects are equal then there hashcode will also equal.



Each node has above mentioned data.

Each index is known as bucket.

Each bucket in term known as node which can be a linked list of nodes.

Here we are storing person score and marks into hashmap using put method.

Hashmap has table size 16.

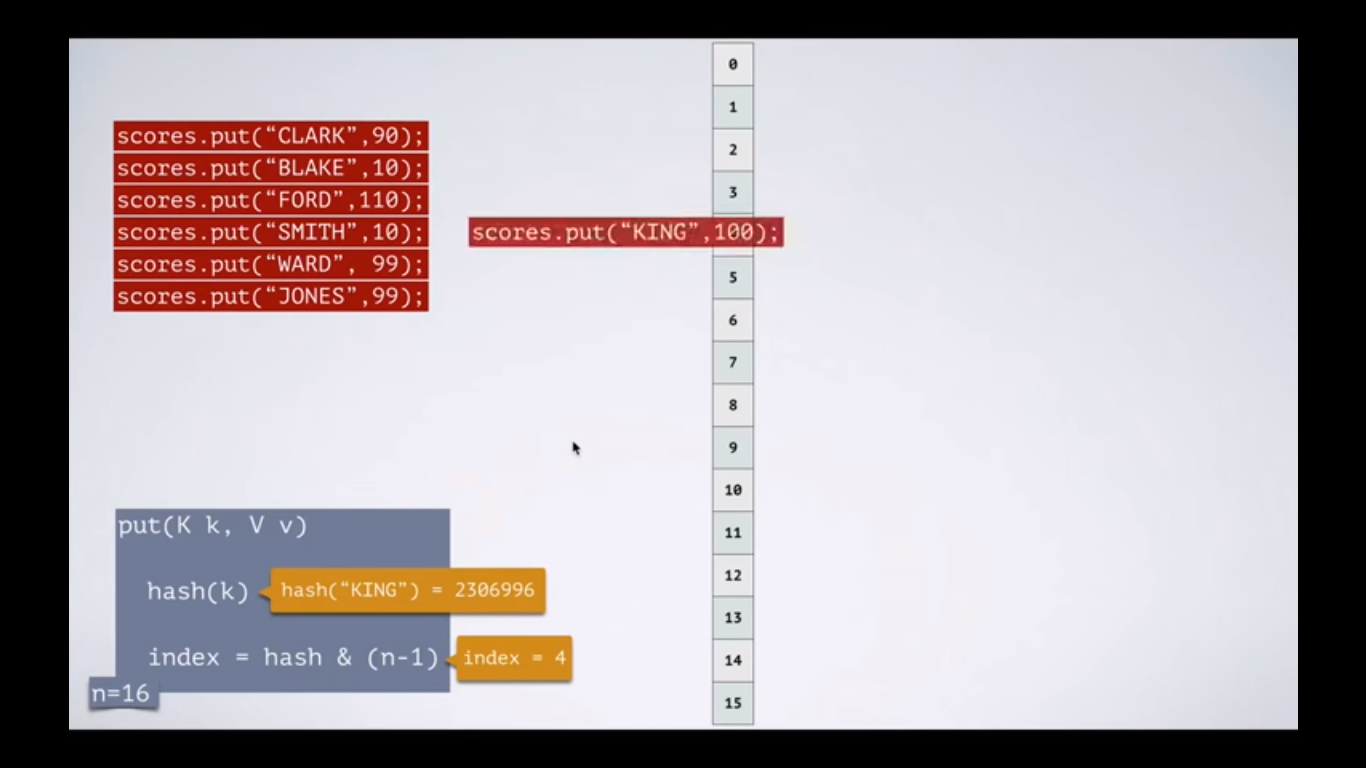
Index is computed using modulo operation.

**HOW PUT OPERATIONS WORKS ?**

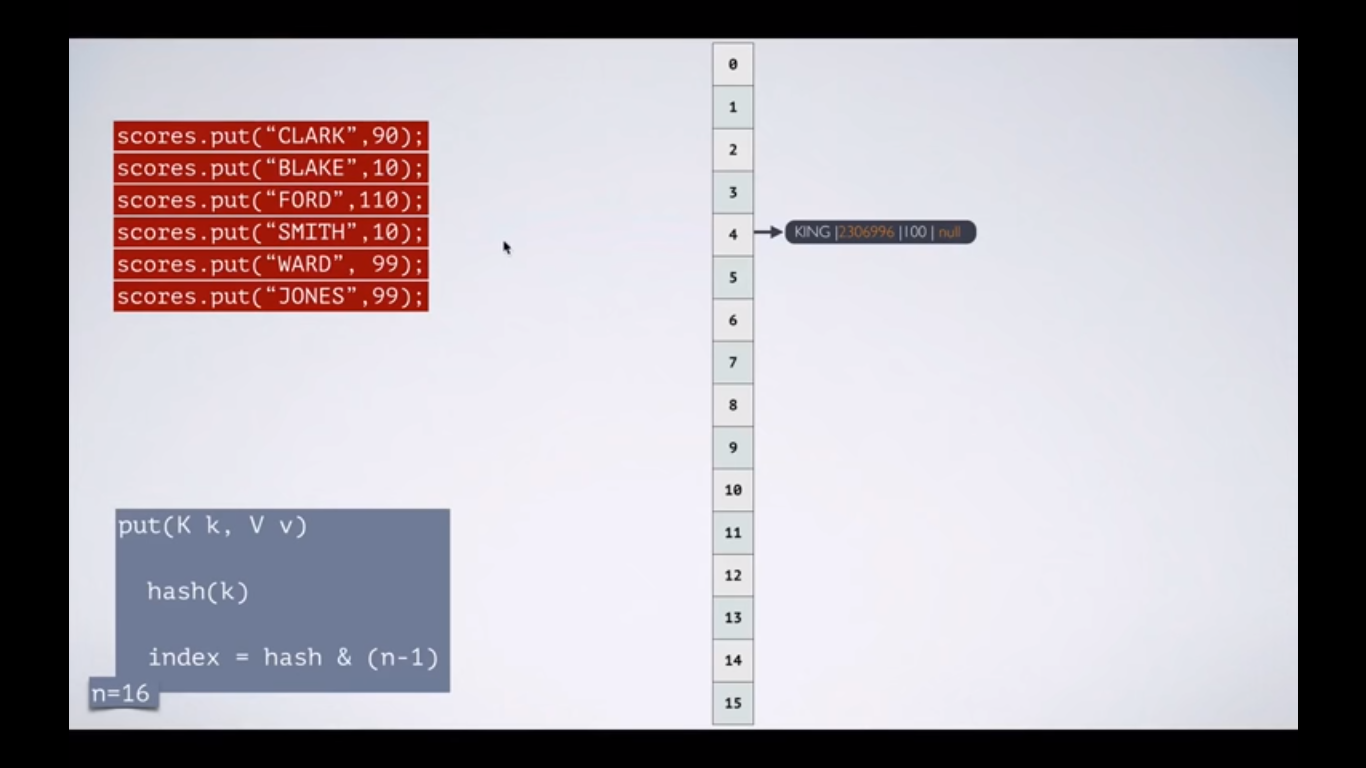
In below pic, we are storing the 1st data in to hashtable.

For the key King first it’ll compute the Hashcode,

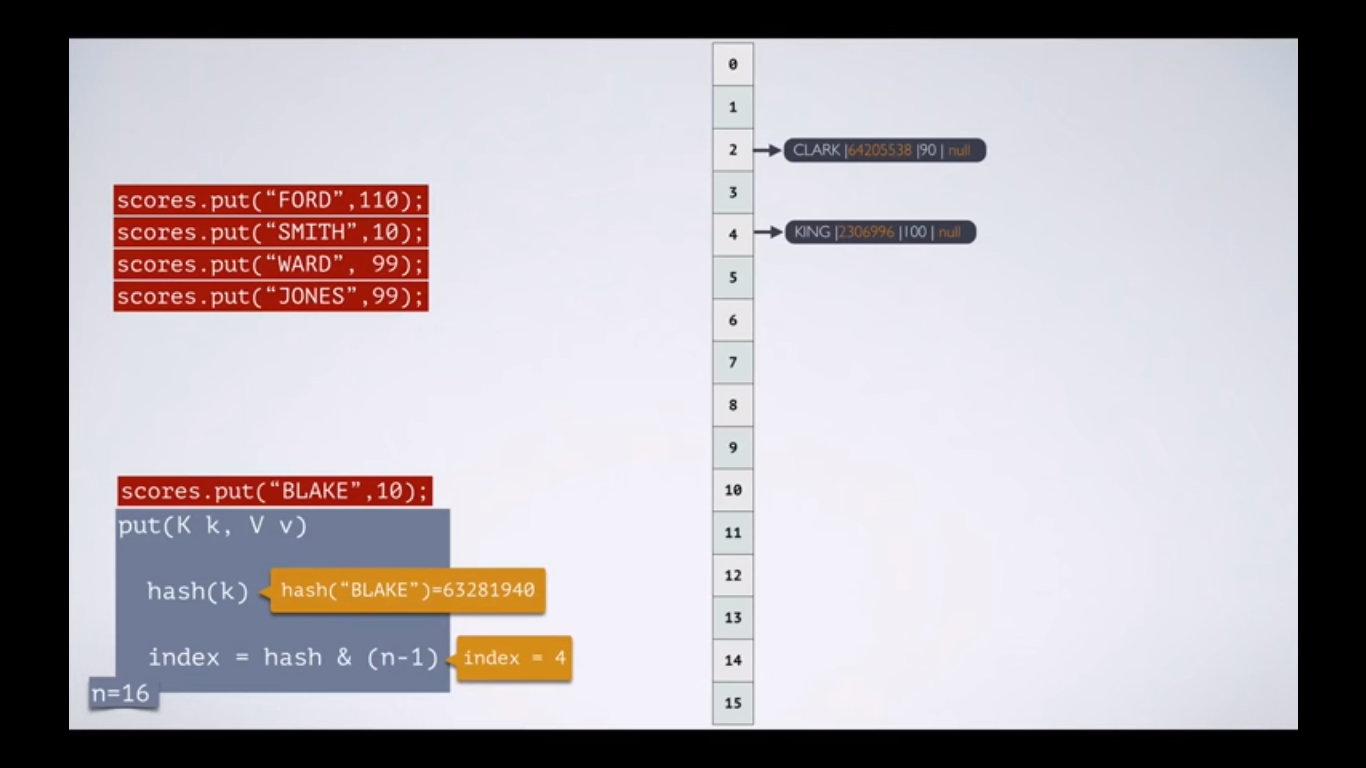
And it’ll compute the index (bucket location)(index is calculated using modulo operation and reminder will be the index key, bitwise operator is used to find the same)



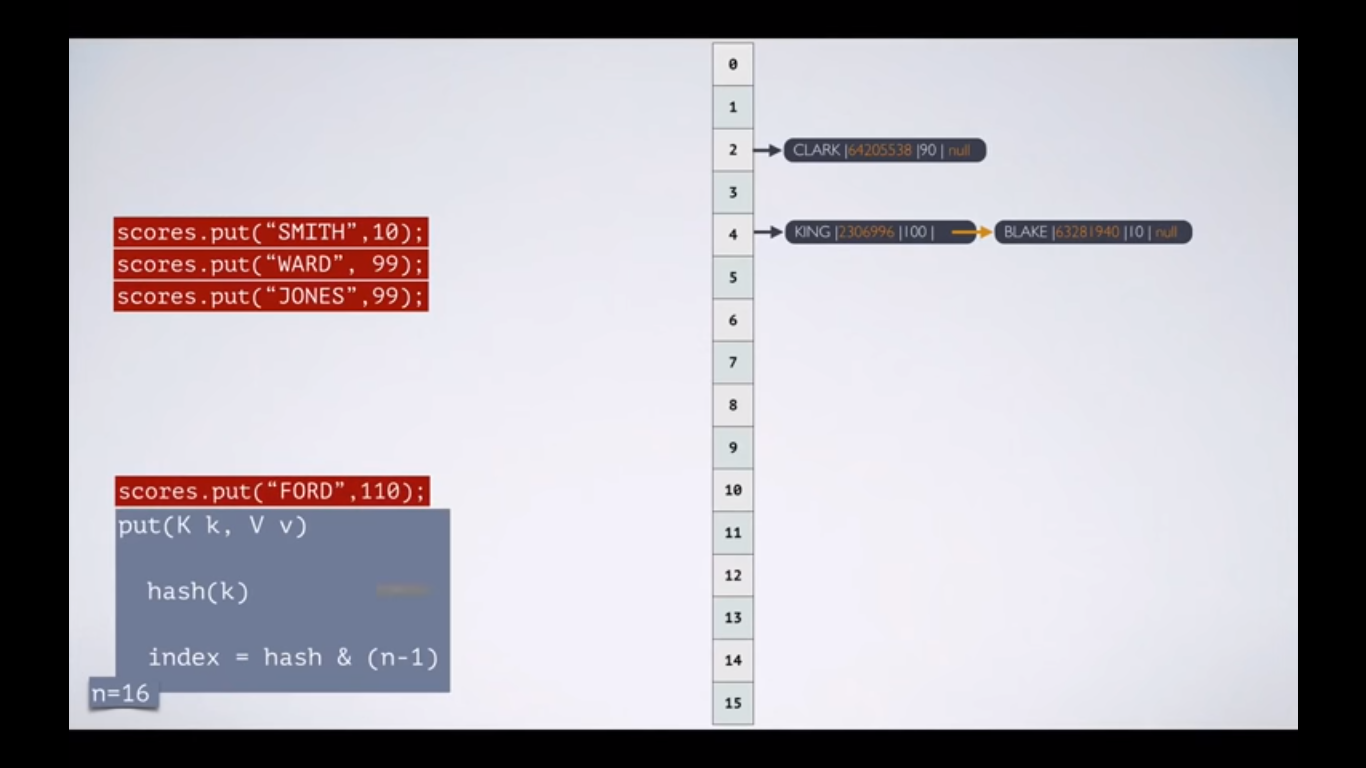
Below is how data stored in hashtable.



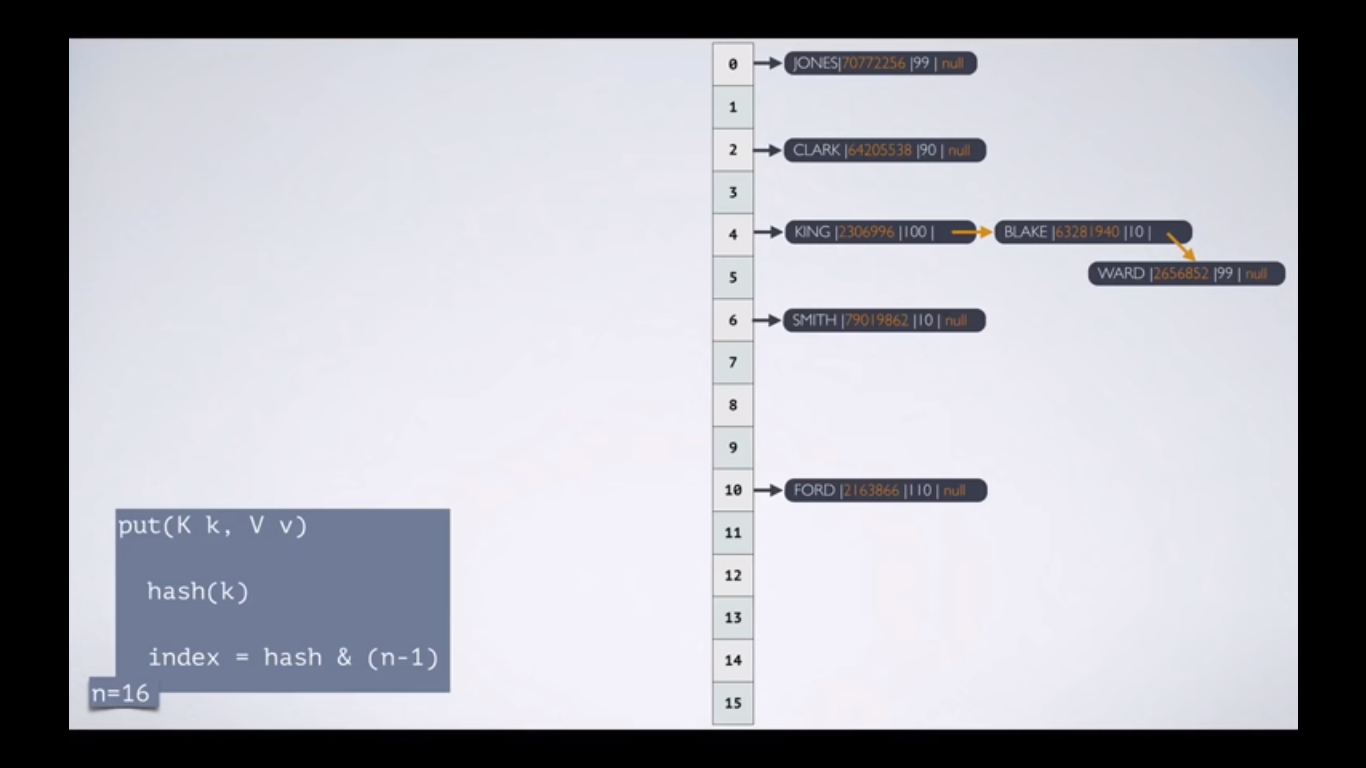
Here is the collision, there is already record present in index 4, one more record has same index.



So pointer is pointed to next node as both records has same index (bucket locations)



At the End:

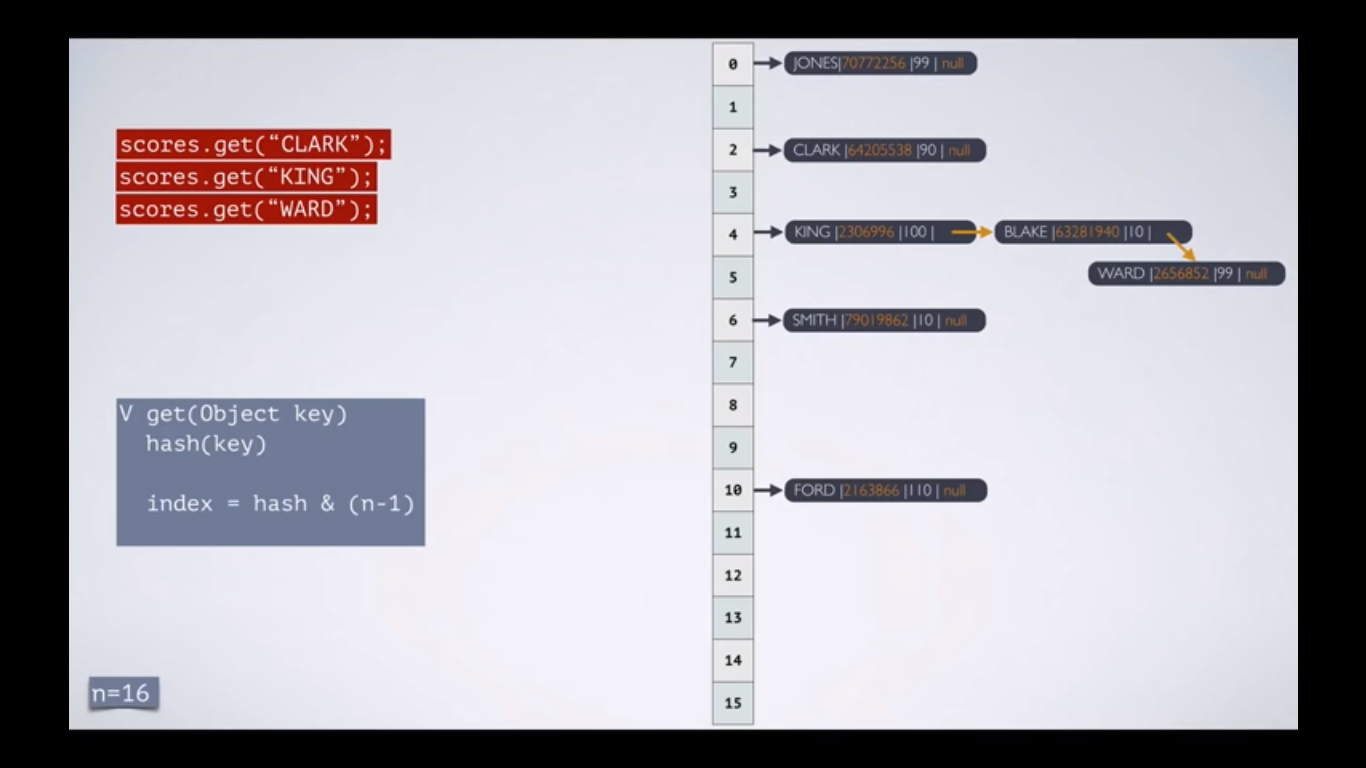


**HashMap stored null as key. So for null key index will always be 0 (zero).**

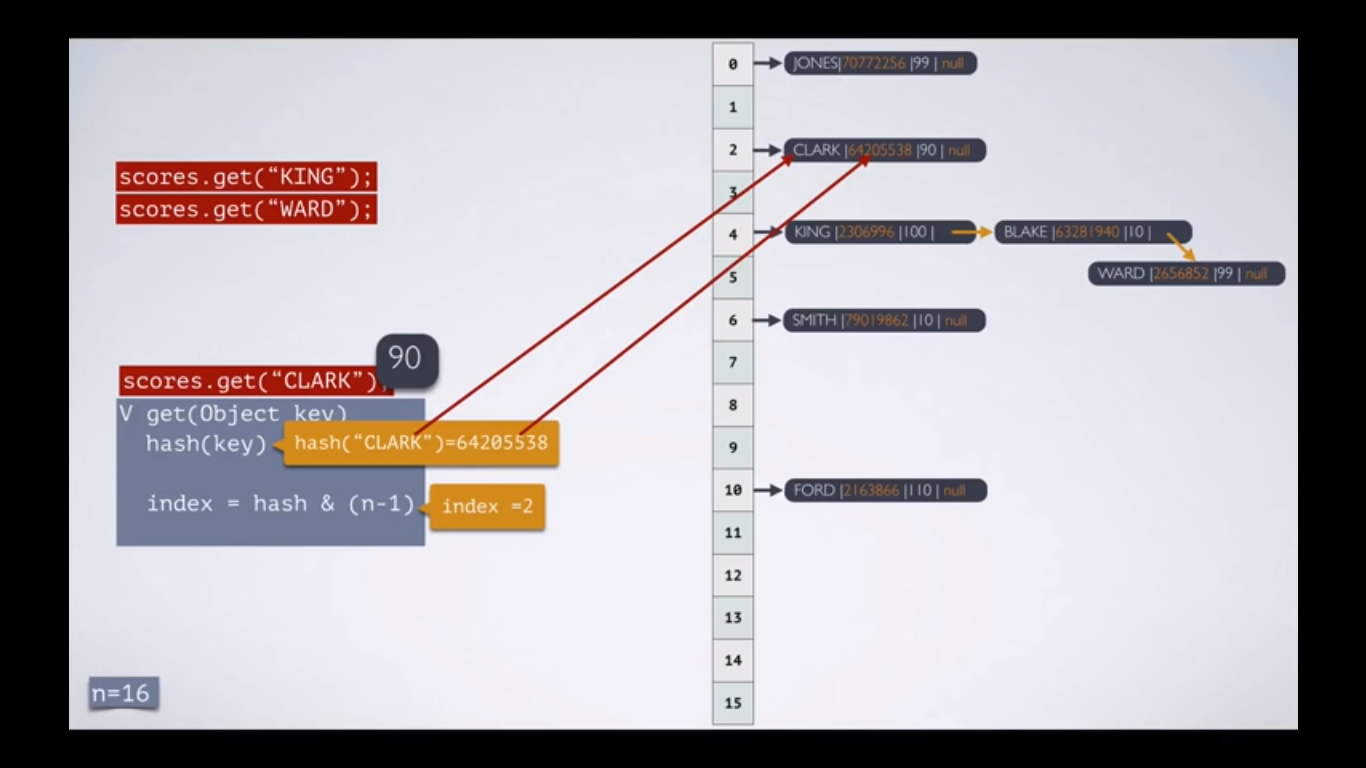
**If we override the hashCode() method and if it returns 1 then it’ll replace the old records with new and stored only latest records.**

HOW GET OPERATIONS WORKS ?

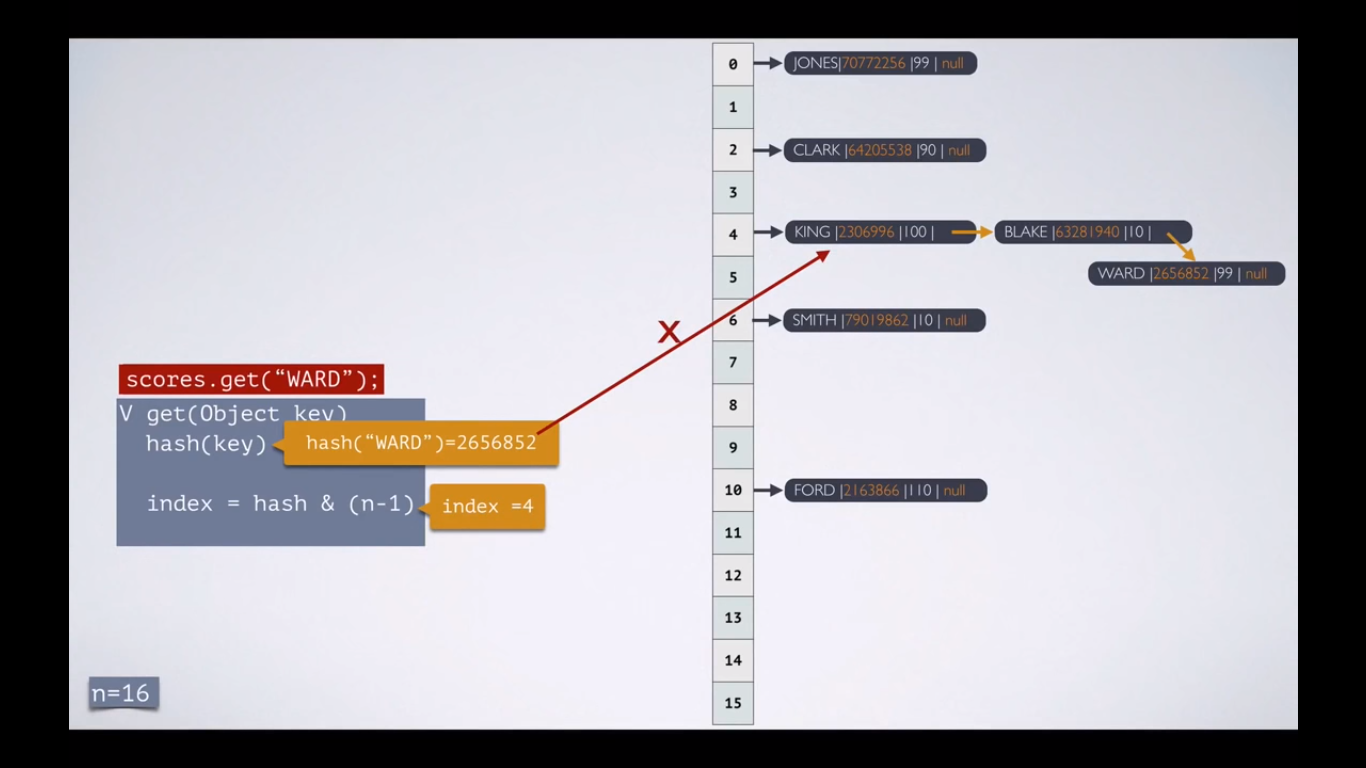
Get method uses **key**  to find the records in hashtable.

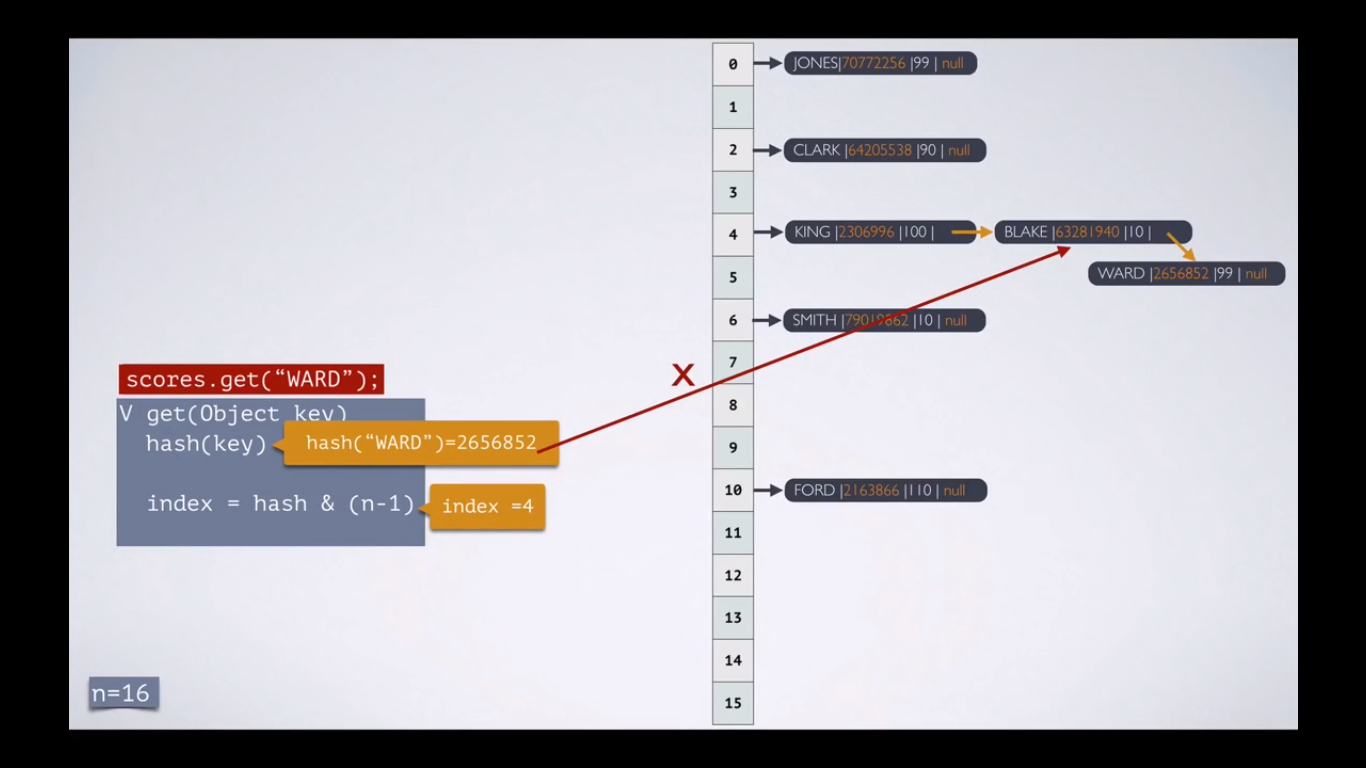


Using key it’ll find the hashcode. It matches the hashcode with records present in bucket location. Then it matches key with key in hashtable using **equal()** method.

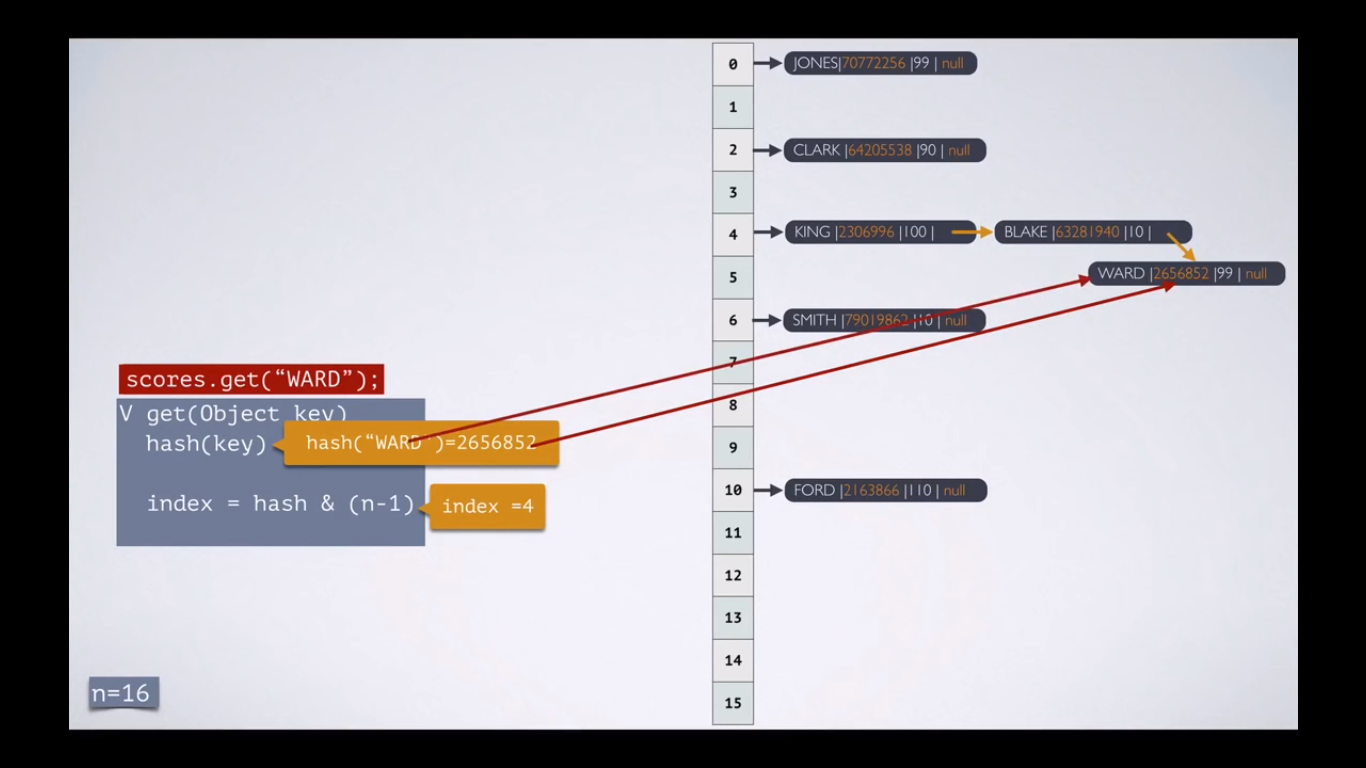


Here it’ll compute the hashcode for the key and matches with first record, if it is doesn’t match then it’ll look into the next node.





In the Final it’ll find the matching hashcode in the table and compare the key. If both are matches then it’ll return the data.



Changes in Java 8:

Linked hashmap replaced with Balanced tree.

