**understand how data is stored and retrieved from HashMap:**

How HashMap works internally is much more important to understand with get and put functions. Lets understand it with step by step.

***Put Operation in HashMap*:**

Let us add below keys and its values into HashMap using put function.

scores.put (“Rohit”, 140);

scores.put (“Dinesh”, 70);

scores.put (“Dhoni”, 90);

scores.put (“Kholi”, 100);

scores.put (“Sachin”, 150);

scores.put (“Dravid”, 130);

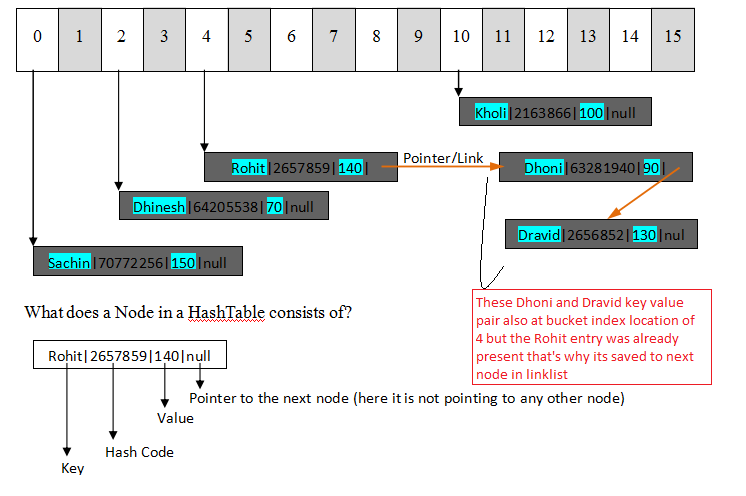
***Let us understand at which location below key value pair will be saved into HashMap.***

**scores.put (“Rohit”, 140);**

* When you call the put function then it computes the hash code of the Key.

using **hash(k) function**, Lets suppose the Hash code of (“Rohit”) is 2657860.

* ***[Most Important]*** Once the hash code is generated then HashMap find the bucket index value using a modular operation to find out where the key value is going to fit in the HashMap.
  + Index = Reminder of (hash code / size of HashMap which is 16 by default)
  + Index = 2657860%16 => 4
  + Index = 4, So 4 is the computed bucket index value where the entry will go sit as a node in the HashMap.
* Now bucket index value computed, so Key and Value pair will save at the bucket index 4 of the hashMap as a node as shown in the below picture, Please note: if the entry is already present then the value will save at next node in the linklist at 4 bucket location.
* Similarly all other entries will also enter into HashMao with same process of generating a hashcode and computing its corresponding index number.
* Java HashMap allows Null Keys (Key= Null) for which the hash code will be zero and it will always goes to index zero of the table.
* Shown below is how a HashMap will look like after entering different sets of Key value pairs.



How HashMap works internally

**Get Operation in HashMap:**

Let us retrieve the value for the below key from HashMap using get function.

scores.get (“Rohit”);.

* So get operation does the same as that of put operation. When the get function is called it basically gets the hash code of the Key.

**hash (key),**Lets suppose Hash of (“Rohit”) is 2657860

* Once it gets the hash code then HashMap does the index computation  to find the index number using below formula and the same we already did while putting to HashMap.
  + Index = Reminder of (hash code / size of HashMap which is 16 by default)
  + Index = 2657860%16 => 4
* Now hashMap lookup at bucket index 4 for the hash code of the key “2657860”.
* Hash code “2657860” found then it lookup for the Key “Rohit” itself in that node or linklist.
* When both hash code and key gets matched with the Node then it looks for the Value in that node and returns the value “140” to the caller.
* When there are multiple entry nodes in a particular index. Like Rohit & Dhoni in same index, now we want to look for Dhoni’s score. It will check the first node for the hash code match, if it does not match then it will look up the second node for the hash code match, similarly it checks all entries/nodes in that index until it gets the hash code match, once it gets the has code match then it lookup for the its Key match.
* When both hash code and key gets matched with the Node then it is same as that of previous case, it looks for the Value in that node and returns the value “90” to the caller.

So in the above example it is explained how to put and get entries using Java HashMap and its was easy to understand the “How HashMap works internally”