1. After Creation of HashMap it will create 16 buckets in heap area.

Load Factor : If HashMap reaches more than 75% of it’s capacity then it doubles the existing capacity.

Bucket : Linked list

Contains nodes

Each node constains

* Data
* Pointer to next node

1. Use Put method to insert data(key,value pair)

Here key need to Object because while performing insertion hashcode will be generated by using Object class’s hashcode method which is why keey need to be Object type not primitive.

* Find hashcode of key
* Find bucket index
  + - Hashcode & (length-1)
    - Eg : 2236 & 15 -🡪 bucket index 12
* D

1. Hash Collision

* If the node is already have data then it checks whether the key is same or not by using

newKey.equals(existingKey)

* If both are same then value will be updated
* Else new node will be added after the existing node.

1. Java 8 Enhancement.

To get data we use get”key”) method

* If the no. of nodes reaches treefy Threshhold then linked list will be converted to tree
* it will use compareTo() method to compare between keys and place them in tree

1. one null key and multiple null values can be inserted

**Important:**

* **initial insertion (Before Hash Collision) 🡪 O(1)**
* **Handling Hash Collisions (Linked List) 🡪 O(n)**
* **Treeification (Transition from linked list to tree) 🡪 O(log n)**

Set:

No duplicates and no insertion order preserved.

HashSet:

* + Underlied data structure : hash table.
  + No duplicates.
  + No insertion order preserved(HashSet or HashMap no insertion is preserved why beacuse here we are using hashcode internaly to store data)
  + One null is possible
  + Same like hashmap it will create 16 buckets and once it filled 75% then size doubled.

|  |
| --- |
| **Note**: in hashmap thread will lock entire hashmap  Where as in concurrent hashmap thread will lock only that bucket |

HashTable:

Store key-value pair

Keys will always be unique.

Thread safe (synchronous)

ConcurrentHashMap:

For multithreaded environment.