Internal working of HashSet and HashMap?

How HashSet internally works?

Each and every element in set is unique. So that there is no duplicate element in set. So In Java, if we want to add element in set then write a code like this.

Example-

```
package com.velocity;

import java.util.HashSet;

public class HashSetDemo {

    public static void main(String[] args) {

        HashSet hs= new HashSet();
        hs.add(3);
        hs.add("ram");
        hs.add("Pune");
        System.out.println(hs);
    }
}
```

Output-

```
[3, Pune, ram]
```

Now, let add the duplicate element in the above code as

```
package com.velocity;

import java.util.HashSet;

public class HashSetDemo {

    public static void main(String[] args) {

        HashSet hs = new HashSet();
        hs.add(3);
        hs.add("ram");
        hs.add("Pune");
        hs.add(3);
        hs.add("Pune");
        System.out.println(hs);
    }
}
```

Output-

```
[3, Pune, ram]
```

Now what happen internally when you pass duplicate element in set then add () method of set object, it will return false and do not add to hashset as element is already present.

But main problem is arising that how it return false, here is the answer.

When you open hashset implementation of add method () in Java API's you will find the following code-

```
Public class HashSet <E> extends AbstractSet<E> implements Set<E>, clonable
java.o.serializable

{

Private transient Hashmap<E, Object> Map;

/* dummy value associate with object in map */


Private static final object Present= new object();


Public HashSet(){

Map= new Hashmap<>();

//some code, other method in hashset

}


Public Boolean add (E e) {

return map. Put(e, PRESENT)==null;


}
```

We are achieving uniqueness in set, internally java through hashmap. Whenever you create the object of hashset it will create the object of hashmap as see in above.

As we know, in hashmap each key is unique. We do in set is that we pass argument in add(Element E) that is E as key in hashmap, now we need to associate some value to key, so what java developer did to pass dummy value that is(new Object()); which is referred by object reference PRESENT.

So actually when you are adding line in hashset like hashset.add(3) what java internally is that it will put that element as E here as 3 key in hashmap and some dummy value that object is passed as value to key.

If you see code of hashmap put(K k, value v) method, you will find something like this,

```
Public v put (K Key, V value){
//some code


}
```

The main point is that .put(key,value) will return

1. Null, if key is unique and added to map.
2. Old value of key, if key is duplicate.

So in Hashset add() method, we check return value of map.put(key,value) method will null value i.e.

```
Public Boolean add(E e){


// code here
}
```

So If Map.put(key,value) return null, then map.put(e,PRESENT)==null, then map.put(e,PRESENT)==null will return true & element added to hashset.

So If Map.put(key,value) return old value of key, then map.put(e,PRESENT)==null, then map.put(e,PRESENT)==null will return false & element is not added to hashset.

Internal working of HashMap?

If you want to represent group of objects as key-value pair then you should go for map.

Example-

```
package com.velocity;

import java.util.HashMap;

public class HashMapDemo {

    public static void main(String[] args) {
        HashMap hm = new HashMap();
        hm.put("Vijay", 10);
        hm.put("Ajay", 20);
        hm.put("Pune", 30);
        hm.put("Sanjay", 50);
        hm.put("Ajay", 100);
        System.out.println(hm);
    }

}
```

Output-

```
{Pune=30, Ajay=100, Sanjay =50, Vijay=10}
```

How hashCode() and equals() methods works into HashMap?

hashCode() method

hashCode method is used to get the hashcode of every objects.

equals() method

equals method is used to check that 2 objects are equal or not. This method is provided by Object class. HashMap uses equals() to compare the key whether the are equal or not. If equals() method return true, they are equal otherwise not equal.

How to calculate index internally?

Put(K k, V v)   //Here put is the method having key and value pair

hash(k)   -> hash(Vijay) ->23560520

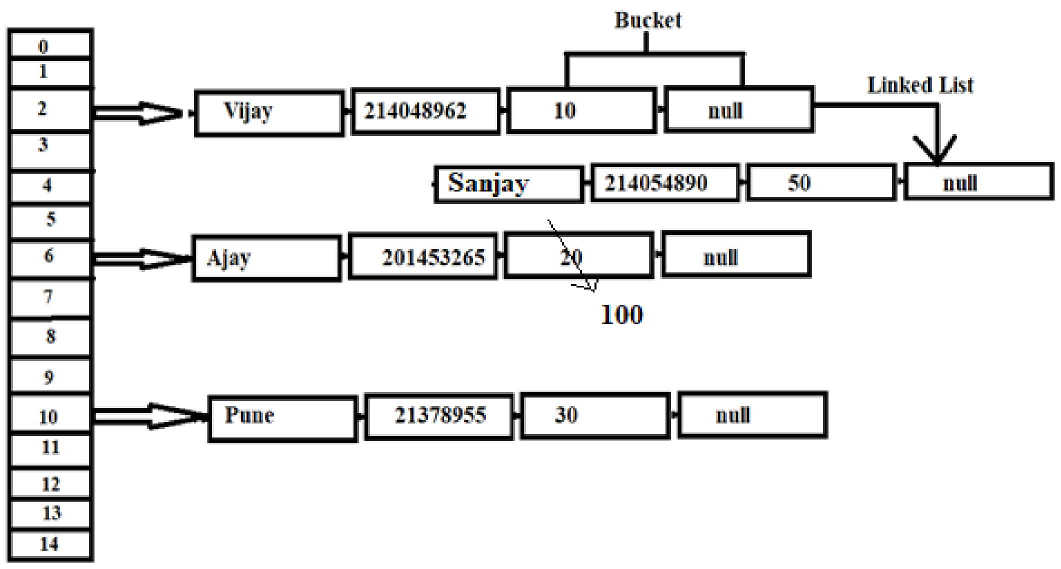
index= hash & (n-1)   here index=2

so Vijay will be stored at 2nd position.

Next time again I get the 2nd index or position for Sanjay then what will happen?

It will store it at 2nd and uses the linked list as shown in fig.

After than I get the 6th index so Ajay will be stored at 6th position then next time get the index 10th then pune will be placed at 10th positon.



How get method works here?

Now let’s try some get method to get a value. get(K key) method is used to get a value by its key. If you don’t know the key then it is not possible to fetch a value.