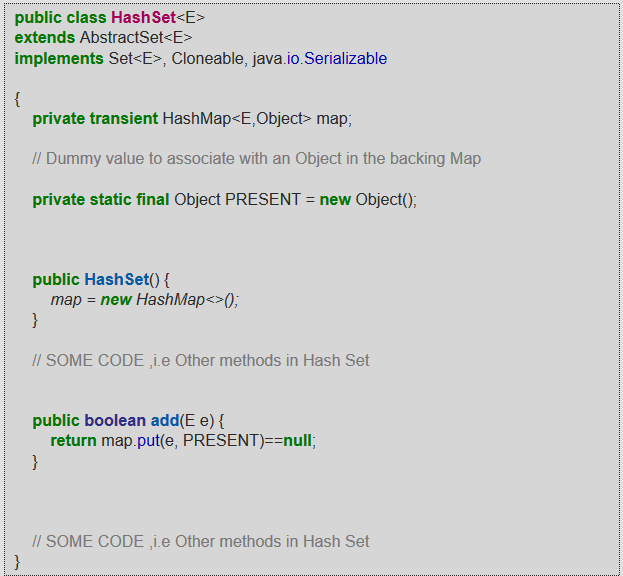
**Internal implementation of Set/HashSet (How Set Ensures Uniqueness)**

**Set Implementation Internally in Java**  
Each and every element in the set is unique .  So that there is no duplicate element in set .

When you open the HashSet implementation of the add() method in Java Apis that is rt.jar , you will find the following code in it.



So , we are achieving uniqueness in Set,internally in java  through HashMap . Whenever you create an object of HashSet it will create an object of HashMap as you can see in the italic lines in the above code .

As we know in HashMap each key is unique . So what we do in the set is that we pass the argument in the add(Elemene E) that is E as a key in the HashMap . Now we need to associate some value to the key , so what Java apis developer did is to pass the Dummy  value that is ( new Object () ) which is referred by Object reference PRESENT .

So , actually when you are adding a line in HashSet like  hashset.add(3)   what java does internally is that it will put that element E here 3 as a key in the HashMap(created during HashSet object creation) and some dummy value that is Object's object is passed as a value to the key.

Now if you see the code of the HashMap put(Key k,Value V) method , you will find something like this  
  
 public V put(K key, V value) {  
//Some code  
}  
  
The main point to notice in above code is that put (key,value) will return  
  
1.  null , if key is unique and added to the map  
2.  Old Value of the key , if key is duplicate  
  
So , in HashSet add() method ,  we check the return value of map.put(key,value) method with null value.

public boolean add(E e) {  
            return map.put(e, PRESENT)==null;  
 }

So , if map.put(key,value) returns null ,then  
map.put(e, PRESENT)==null      will return true and element is added to the HashSet.  
  
  
  
So , if map.put(key,value) returns old value of the key ,then  
map.put(e, PRESENT)==null      will return false and element is  not added to the HashSet .