**HashSet**

* HashSet stores the elements by using a mechanism called **hashing.**
* HashSet contains unique elements only.
* HashSet allows null value.
* HashSet class is non synchronized.
* HashSet doesn't maintain the insertion order. Here, elements are inserted on the basis of their hashcode.
* HashSet is the best approach for search operations.

**// Java program to study HashSet**

import java.util.\*;

public class Hashing {

public static void main(String args[]) {

**// creating a HashSet**

HashSet<Integer> mySet = new HashSet<>();

**// add elements**

mySet.add(1);

mySet.add(2);

mySet.add(3);

mySet.add(2);

**// Printing the HashSet**

System.out.println(mySet);

**// Display size of HashSet**

System.out.println("Size of the HashSet is "+ mySet.size());

**// search for an element**

if(mySet.contains(6))

System.out.println("6 is found.");

else

System.out.println("6 is not found.");

**// remove an element**

mySet.remove(2);

System.out.println(mySet);

**// Iterate**

Iterator myIt = mySet.iterator();

while(myIt.hasNext()){

System.out.println(myIt.next());

}

}

}

**HashMap**

### 

* Java HashMap contains values based on the key (Key-value pair)
* Java HashMap contains only unique keys. It allows duplicate values but not duplicate keys.
* Java HashMap may have one null key and multiple null values.
* Java HashMap is non synchronized.
* Java HashMap maintains no order.
* Retrieval and insertion of elements is usually O(1) time complexity.
* Uses Hashing function

**// Java program to study HashMap**

import java.util.\*;

public class Hashing {

public static void main(String args[]) {

**// creating a HashMap**

HashMap <Integer,String> myMap = new HashMap<>();

**// add pairs**

myMap.put(1,"Sam");

myMap.put(2,"Ali");

myMap.put(3,"Rita");

myMap.put(4,"Sara");

**// Print HashMap**

System.out.println(myMap);

**// update a pair**

myMap.put(3,"Sita");

System.out.println(myMap);

**// search for a pair**

if(myMap.containsKey(6))

System.out.println("Key is present");

else System.out.println("Key is not present");

**// get value of a key**

System.out.println(myMap.get(1));

System.out.println(myMap.get(6));

**// display size of HashMap**

System.out.println("Size of my HashMap is " + myMap.size());

**// remove a pair**

myMap.remove(3);

System.out.println(myMap);

**// Iterate EntrySet**

for(Map.Entry<Integer,String> pairs : myMap.entrySet()){

System.out.println(pairs.getKey());

System.out.println(pairs.getValue());

}

**// Iterate KeySet**

Set<Integer> keys = myMap.keySet();

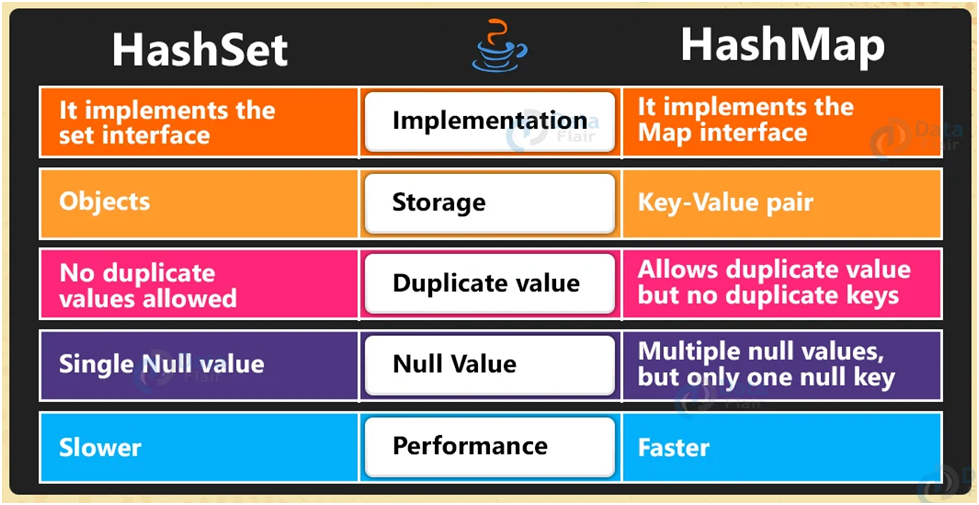
for(Integer key : keys)

System.out.println(key+" "+ myMap.get(key));

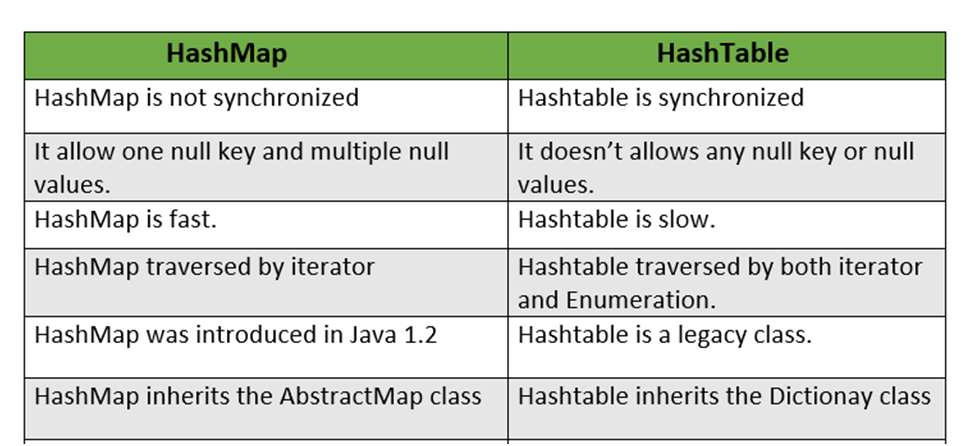
}

}

**HashSet va HashMap**



**HashMap vs HashTable**

****

**Video Recommendation**: HashSet - <https://youtu.be/eJiGN1h8XzM>

**READ UP**: HashSet - <https://www.javatpoint.com/java-hashset>

**Video Recommendation:** HashMap - <https://youtu.be/WeF3_nk-UqY>

**READ UP:** HashMap - <https://www.javatpoint.com/java-hashmap>

**Video Recommendation:** Hashing - <https://youtube.com/playlist?list=PL6Zs6LgrJj3uyNihSkIq9QcNMylpR_9ba>

**READ UP**: <https://www.baeldung.com/java-arraylist-vs-linkedlist-vs-hashmap>

**Questions for practise:**

### Write a program to find the highest frequency of a character in a given string

# Check if pair with given Sum exists in Array

# Find the length of largest subarray with 0 sum