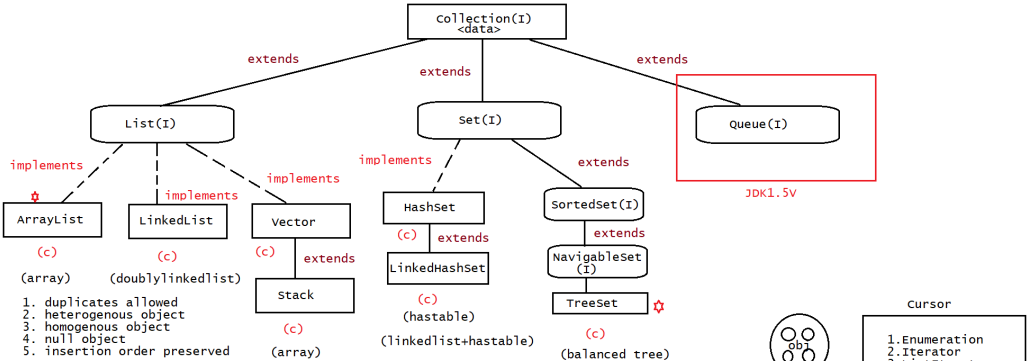
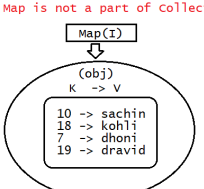
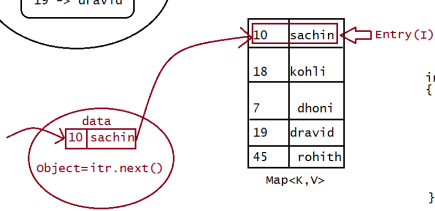


Map is not a part of Collection

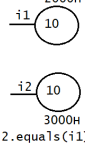


Note:
Value is retrieved with the help of key
Internally key would be stored through hashing technique(hashtable)
Key can't be duplicated whereas value can be duplicated.




```
interface Map<K,V>
{
   interface Entry
   {
      public abstract K getKey();
      public abstract V getValue();
      public abstract V setValue(V);

   }
}
```

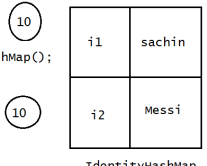
```
HashMap h = new HashMap();

//Creating a key
Integer i1= new Integer(10);
Integer i2= new Integer(10);

//Adding the data to HashMap
h.put(i1,"sachin");
h.put(i2,"Messi");

System.out.println(h);
```

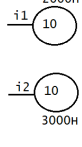


If it is HashMap
JVM calls
equals(obj2)
to identify
whether keys are
duplicated or not

i2.equals(i1)=true

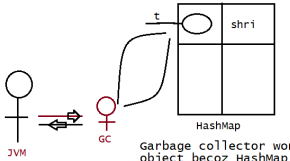
```
IdentityHashMap h = new IdentityHashMap();

   //Creating a key
   Integer i1= new Integer(10);
   Integer i2= new Integer(10);

   //Adding the data to HashMap
   h.put(i1,"sachin");
   h.put(i2,"Messi");

   System.out.println(h);
```





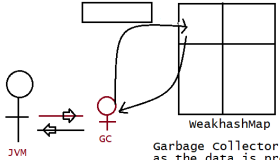
If it is IdentityHashMap
JVM calls
== operator
to identify
whether keys are
duplicated or not

i1==i2(false)



Garbage collector wont' clean temp object becoz HashMap would alway dominate the GarbageCollector

System.gc



Without existence of key there can't be any value

Garbage Collector will clean the object as the data is present in WeakHashMap. So we say GC dominates WeakHashMap

System.gc

```
temp  t =null ; //Clean the object
```

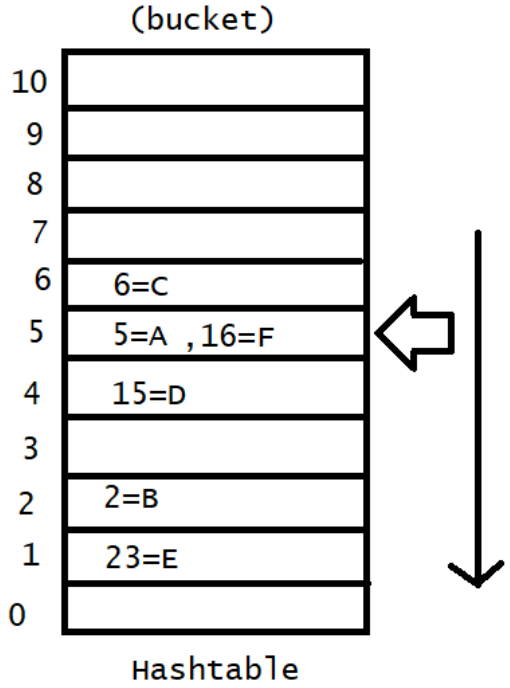
```
Hashtable hm = new Hashtable();//Default capacity is 11
 hm.put(new Temp(5),"A");
 hm.put(new Temp(2),"B");
 hm.put(new Temp(6),"C");
 hm.put(new Temp(15),"D");
 hm.put(new Temp(23),"E");
 hm.put(new Temp(16),"f");

 System.out.println(hm);
```

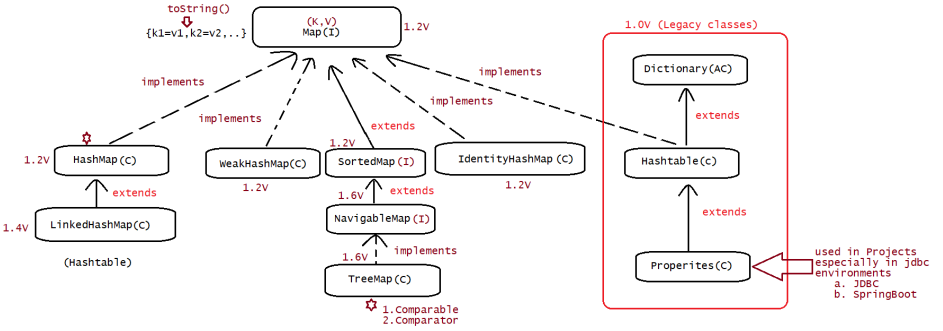
{6=C,16=F,5=A,15=D,2=B,23=E}

15%11= 4

23%11 = 1

16%11 = 5




toString()
{k1=v1,k2=v2,..}
(K,V)
Map(I)
1.2v
1.0V (Legacy classes)
implements
implements
implements
implements
extends
1.2v
HashMap(C)
WeakHashMap(C)
SortedMap(I)
IdentityHashMap(C)
1.2v
1.2v
1.2v
Dictionary(AC)
extends
Hashtable(c)
extends
1.4v
LinkedHashMap(C)
extends
1.6v
NavigableMap(I)
extends
(Hashtable)
1.6v
TreeMap(C)
implements
Properites(C)
used in Projects
especially in jdbc
environments
a. JDBC
b. SpringBoot
1.Comparable
2.Comparator