Map interface->

A map contains values on the basis of key i.e. key and value pair.

Each key and value pair is known as an entry. Map contains only unique keys.

Map is useful if you have to search, update or delete elements on the basis of key.

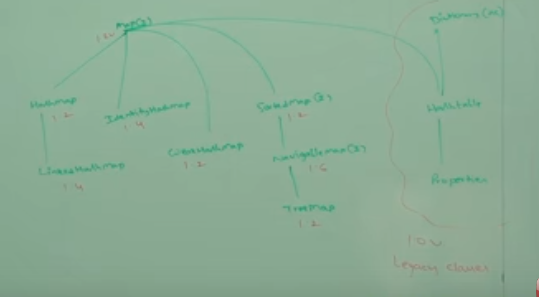
\*Map is not a child interface of collection.

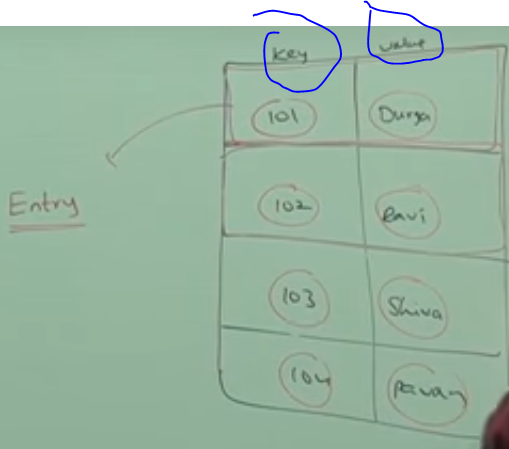
\*if we want to represent a group of objects as keyvalue pairs then we should go for map.

\*key and values are object only ,,

\*dublicates are not allowd for key…but values can be dublicated.

\*each key value pair is called “entry” hence map is consider as a collection of entry objects.

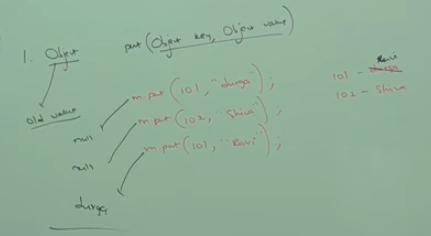




Methods of map->

1.object put(object key,object value)

If we try to add dublicates it reset to this and old value get return through object.and if not get duplicates the nibject simply return null.



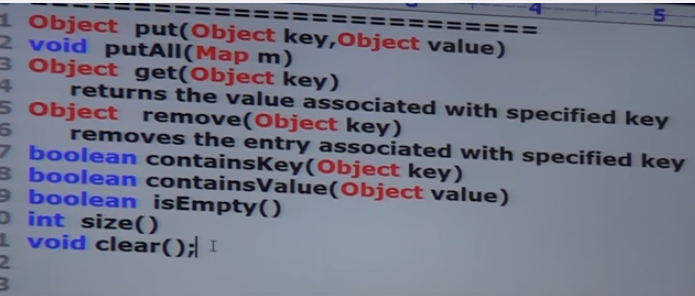
2.void put all(MAP map)   
It is used to insert the specified map in this map.

3. object get(object key)->value we get;if key is not there then null returns.

4.object remove (object key)-> It is used to delete an entry for the specified key.

5.boolean containKey(object key)-> It is used to search the specified key from this map.

M.size(),M.isEmpty()….



COLLECTION VIEW->

{

IF we want to only call key then we call the method ->

Set keyset();//duplicates are not allowed.

IF we want to only call values then we call the method ->

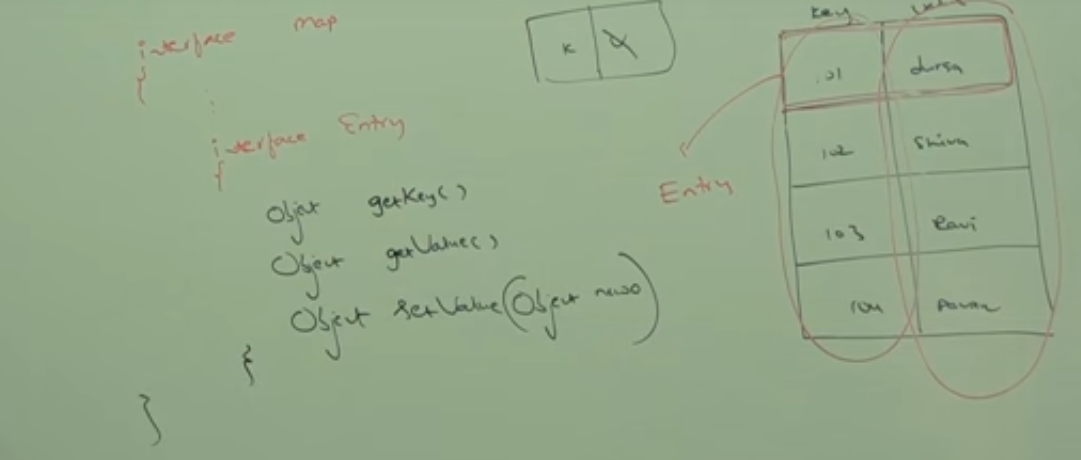
Collection values();//duplicates allowed

IF we want to call entry(key+value) then

Set enterySet();

}

Now entry interface->



Three methods in ENTRY

-a map is a group of key values pairs.and each key value is called ENTRY.WITHOUT EXISTING MAP OBJECT THERE IS NO CHANCE OF EXISTING ENTRY OBJECT.HENCE

Entry interaface inside a map intera face.this method can only accesbile onEntry object.

Method of ENTRY INTERFACE->

1.getkey();

2.getvalue();

3.setvalue();