Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-six)

6.6: Maps

Maps are appropriately named in that it values are stored not sequentially, but in pairs with a unique key. The key and value can be any object. Keys function like an index in arrays, except that it does not have to be a numeric index.

You retrieve values using get() with an object key as a parameter. To insert an object, use put() and pass the key and the value as parameters.

Table: Object key

Method	Description
E get(E key)	Return the value associated with key
E put(E key, E value)	Inserts key with an associated value
E remove(E key)	Removes key from the map
Set keySet( )	Returns a Set of all the keys in the map
Set values( )	Returns a Set of all the values in the map
boolean containsKey(E key)	Returns a boolean on whether key exists in the map
boolean containsValue(E value)	Returns a boolean on whether value exists in the map
void clear( )	Deletes all of the elements in a map

Concrete map classes include HashMap and TreeMap. The HashMap and the TreeMap differ only in the implementation which affects the order of its elements. In a HashMap, the order of elements is not guaranteed, but they are in a TreeMap.

To iterate through a map, get the collection of its keys using keySet( ). Then iterate through the set of keys and pass it as a parameter to get( ) to get the associated value. If you want to retrieve only the values, you can get its collection using values( ).

The code below demonstrates adding key/value pairs to a HashMap and iterating through the contents using its keys.

```
HashMap hm = new HashMap();
StateQuarter sq = new StateQuarter("CA", "California",  2005);
hm.put(sq.abbrev, sq);
sq = new StateQuarter("DE", "Delaware",  1999);
hm.put(sq.abbrev, sq);
sq = new StateQuarter("HI", "Hawaii",  2008);
hm.put(sq.abbrev, sq);

for (Object o : hm.keySet()) {
```

```
    String key = (String)o;
    StateQuarter sqo =  (StateQuarter)hm.get(key);
    System.out.println(sqo.abbrev  + " = " + sqo.name + "(" + sqo.release_year + ")");
}
```

The output of this is:

HI = Hawaii(2008)

DE = Delaware(1999)

CA = California(2005)

Note that your output might not match the above as the order of the keys is not guaranteed.