Course 5 - Sprint 2: Manipulate Objects Using Unordered Collections and Construct Objects as a Key Value Pair

Learning Consolidation

1. Differentiate between HashSet and TreeSet.

HashSet	**TreeSet**
It is unordered in nature	By default, it stores elements in their natural ordering
Has faster processing time	Has slower processing time
Uses hasCode() and equals() for comparing	Uses compare() and compareTo() for comparing
Allows only one null element	Doesn't allow any null element

2. Why Map interface does not extend Collection interface?
    a. Because they are incompatible. Collection has a method add(Object o). Map can not have such method because it need key-value pair. There are other reasons also such as Map supports keySet, valueSet etc. Collection classes does not have such views.Due to such big differences, Collection interface was not used in Map interface, and it was build in separate hierarchy.

3. Explain ConcurrentHashMap? How it works?
    a. A hash table supporting full concurrency of retrievals and adjustable expected concurrency for updates. This class obeys the same functional specification as Hashtable, and includes versions of methods corresponding to each method of Hashtable. However, even though all operations are thread-safe, retrieval operations do not entail locking, and there is not any support for locking the entire

table in a way that prevents all access. This class is fully interoperable with Hashtable in programs that rely on its thread safety but not on its synchronization details.

4. How to design a good key for hashmap?
    a. Another good question usually followed up after answering how hashmap works. Well, the most important constraint is you must be able to fetch the value object back in future. Otherwise, there is no use of having such a data structure. If you understand the working of hashmap, you will find it largely depends on hashCode() and equals() method of Key objects. So a good key object must provide same hashCode() again and again, no matter how many times it is fetched. Similarly, same keys must return true when compare with equals() method and different keys must return false. For this reason, immutable classes are considered best candidate for HashMap keys.

5. What is Comparable and Comparator interface?
    a. In java. all collection which have feature of automatic sorting, uses compare methods to ensure the correct sorting of elements. For example classes which use sorting are TreeSet, TreeMap etc. To sort the data elements a class needs to implement Comparator or Comparable interface. That's why all Wrapper classes like Integer,Double and String class implements Comparable interface. Comparable helps in preserving default natural sorting, whereas Comparator helps in sorting the elements in some special required sorting pattern. The instance of comparator if passed usually as collection's constructor argument in supporting collections.

6. Predict the output of the following program:

```
import java.util.*;
import java.util.Map.*;
public class MapDemo {
    public static void main(String args[]) {
        HashMap participant = new HashMap();
        participant.put(1 + 1, "John");
        participant.put(0 + 1, "Jenny");
        participant.put(2 + 1, "Charles");
        Set set = participant.entrySet();
        Iterator itr = set.iterator();
        while (itr.hasNext()) {
            Map.Entry m = (Entry) itr.next();
            System.out.print(m.getKey() + "
" + m.getValue() + ", ");
        }
        System.out.println();
        itr = set.iterator();
        while (itr.hasNext()) {
            Map.Entry m = (Map.Entry) itr.next();
            System.out.print(m.getKey() + "
" + m.getValue() + ", ");
        }
    }
}
```