As you arrive...

We won’t talk about it too much in class, but sometimes you might want to loop over all the keys and values in a map.

Open up your code and modify your Recitation 3 prep code to loop across each of the keys of the map. Make your code print output like this for all the values of the map:

- At key ‘7’ I had the value ‘Mike is cool’
- At key ‘65’ I had the value ‘SPAM SPAM SPAM SPAM’
- At key ‘300’ I had the value ‘qqqqq’

To see how, check out the keySet() function in the HashMap class (it’s in the Java docs)

What we will do today

1. Write the processResponse code for Jotto
2. Get you capable of writing code that uses Maps to solve APTs

Structure of processResponse

- Check all of the words in the myWordList
- For each word in the list, see if the number of letters in common differs from what the player says. If so, that word should be eliminated from myWordList.
- From our the new myWordList, select a new random word

What does this code print out?

```java
ArrayList<Integer> list = new ArrayList<Integer>();
for(int i = 0; i < 10; i++) {
    list.add(i);
}
for(int i = 0; i < list.size(); i++) {
    list.remove(i);
}
System.out.println(list);
```

A. [], because we’ve removed all the elements. It’s not like this is a trick question or anything.
B. [1 3 5 7 9] – just the odd elements
C. [0 2 4 6 8] – just the even elements
D. [0 1 2 3 4 5 6 7 8 9], because ArrayLists are immutable
Structure of processResponse

- Check all of the words in the myWordList
- For each word in the list, see if the number of letters in common match the number from the player. If so, that word could be a solution: add it to our list of possible words.
- From our list of possible words, select a new random word
- Replace myWordList with list of possible words

Write processResponse in JottoModel.java

1. Use a loop to iterate through all the words in myWordList
2. Check how many letters in common the last guess had with each word in the list (use the handy function commonCount)
3. Words that have a same number of letters in common need to added to the new list of possible words
4. Select a new random word from the possible words
5. Replace myWordList with our new possible words

//how to get a random number between 0 and 9 in java
Random r = new Random();
int between0and9 = r.nextInt(10);

If you have trouble, ask those around you and if nobody knows...raise your hand

Fixing the bug

A. We don’t need to modify myWordList...we’ll just keep the “matching words” in a local variable and select a random one from that
B. We’ll make two fields myCompleteWordList and myTemporaryWordList. We’ll initialize myTemporaryWordList from the complete one at the beginning of every game
C. We’ll keep a new list of booleans that indicates which words have been eliminated (or maybe a Set?), but we won’t change myWordList
D. We’ll do A, but we’ll keep a list of all the previous guessed words too, so we can remove all the words eliminated by every guess

What we will do today

1. Write the processResponse code for Jotto
2. Get you capable of writing code that uses Maps to solve APTs
My solution to the prep assignment

```java
import java.util.*;

public class Rec3 {
    public static void main(String[] args) {
        HashMap<Integer, Integer> map = new HashMap<Integer, Integer>();
        map.put(77, 12);
        map.put(74, 12);
        map.put(77, 1000);
        //prints at value 77: 1000
        System.out.println("At value 77: "+map.get(77));
    }
}
```

My solution to iterating over a Map

```java
import java.util.*;

public class Rec3 {
    public static void main(String[] args) {
        HashMap<Integer, Integer> map = new HashMap<Integer, Integer>();
        map.put(77, 12);
        map.put(74, 12);
        map.put(77, 1000);
        Set<Integer> keys = map.keySet();
        for(int key : keys) {
            System.out.println("At key "+key+" I had the value "+map.get(key)+" ");
        }
    }
}
```

A sample APT