CSE143X Section #11 Cheat Sheet

For all problems involving lists and sets, the contents will be displayed using the standard square bracket notation used by toString, as in:

```
        [12, 4, 19, 78, 42]
```

List<E> Methods (10.1)

add(value)	appends value at end of list
add(index, value)	inserts given value at given index, shifting subsequent values right
clear()	removes all elements of the list
indexOf(value)	returns first index where given value is found in list (-1 if not found)
get(index)	returns the value at given index
remove(index)	removes/returns value at given index, shifting subsequent values left
set(index, value)	replaces value at given index with given value
size()	returns the number of elements in list
addAll(list)	adds all elements from the given collection to the end of the list
contains(value)	returns true if the given value is found somewhere in this list
remove(value)	finds and removes the given value from this list
removeAll(list)	removes any elements found in the given collection from this list
iterator()	returns an object used to examine the contents of the list

Set<E> Methods (11.2)

add(value)	adds the given value to the set
contains(value)	returns true if the given value is found in the set
remove(value)	removes the given value from the set
clear()	removes all elements of the set
size()	returns the number of elements in the set
isEmpty()	returns true if the set's size is 0
addAll(collection)	adds all elements from the given collection to the set
containsAll(collection)	returns true if set contains every element from given collection
removeAll(collection)	removes any elements found in the given collection from this set
retainAll(collection)	removes any elements not found in the given collection from this set
iterator()	returns an object used to examine contents of the set

Iterator<E> Methods (11.1)

hasNext()	returns true if there are more elements to be read from collection
next()	reads and returns the next element from the collection
remove()	removes the last element returned by next from the collection

For all problems involving maps, the contents will be displayed using the standard key=value pairs enclosed in curly braces used by toString. For example, given the following Map:

```
        Map<Integer, String> months = new TreeMap<Integer, String>();
        months.put(3, "March");
        months.put(1, "January");
        months.put(2, "February");
```

the map would be displayed as follows:

```
        {1=January, 2=February, 3=March}
```

Map<K, V> Methods (11.3)

put(key, value)	adds a mapping from the given key to the given value
get(key)	returns the value mapped to the given key (null if none)
containsKey(key)	returns true if the map contains a mapping for the given key
remove(key)	removes any existing mapping for the given key
clear()	removes all key/value pairs from the map
size()	returns the number of key/value pairs in the map
isEmpty()	returns true if the map's size is 0
keySet()	returns a Set of all keys in the map
values()	returns a Collection of all values in the map
putAll(map)	adds all key/value pairs from the given map to this map