CPW 143 In-Class Exercise for **`Map`** and **`Set`**

1. Write a method named **`sortAndRemoveDuplicates`** that takes a **`List`** of **`Integer`**s as its parameter and rearranges its elements into sorted ascending order, as well as removing all duplicate values from the list. For example, the list **`[7, 4, -9, 4, 15, 8, 27, 7, 11, -5, 32, -9, -9]`** would become **`[-9, -5, 4, 7, 8, 11, 15, 27, 32]`** after a call to your method. Use a **`Set`** as a part of your solution.

2. Write a method named **`is1to1`** that takes a **`Map`**, whose keys and values are **`String`**s, as a parameter and returns **`true`** if no two keys map to the same value. For example, **`{Marty=206, Hawking=123, Smith=929, Newton=123}`** should return **`false`**, but should **`{Marty=206, Hawking=123, Smith=929, Newton=456}`** should return **`true`**. The empty map is considered 1-1, so return **`true`**.

3. Write a method named **`subMap`** that takes two **`Map`**s from String to String as its parameters and return **`true`** if every key in the first **`Map`** is also a key in the second **`Map`** and maps to the same value in the second **`Map`**. For example, **`{Marty=206, Smith=929, Newton=123}`** is a submap of **`{Hawking=123, Newton=123, Marty=206, Smith=929}`**. The empty map is a submap of every map.

Set Methods

`add(value)`	adds the given value to the **`Set`**
`contains(value)`	returns **`true`** if the given value is found in this **`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
`toString()`	returns a string such as **`"[3, 42, -7, 15]"`**
`addAll(Collection)`	adds all elements from the given collection
`containsAll(coll)`	returns **`true`** if this set contains every element from the given set
`equals(set)`	returns **`true`** if given other set contains the same elements
`removeAll(coll)`	removes all elements in the given collection from this set
`retainAll(coll)`	removes elements *not* found in the given collection from this set

Map Methods

`put(key, value)`	adds a mapping from the given key to the given value; if the key already exists, replaces its value with the given one
`get(key)`	returns the value mapped to the given key (**`null`** if not found)
`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
`toString()`	returns a **`String`** such as **`"{a=90, d=60, c=70}"`**
`keySet()`	returns a **`Set`** of all keys in the map
`values()`	returns a **`Collection`** of all values in the map
`putAll()`	adds all key/value pairs from the given **`Map`** to this **`Map`**
`equals()`	returns **`true`** if given **`Map`** has the same mappings as this one