1. (2 pts) The hierarchy of all collections is the interface Collection found in the java.util package. The major subinterface, Set, of this class describes a collection containing unique elements.

2. (2 pts) State the hashCode contract.

1. The hashCode() method must produce a consistent result during a single execution of the program.
2. If the equals method returns true between two objects, then the hashCode() methods must return the same value.
3. If the equals method returns false between two objects, it is not necessary for the hashCode() methods to return different values.

3. (2 pts) Suppose we have an instance of a List defined to hold elements of type String. Recalling that the String class implements the Comparable interface, show how you would produce an instance of the List in which there are no duplicate Strings and the Strings are in alphabetical order.

```java
Set<String> set = new TreeSet<String>();
set.addAll(list);
list = new ArrayList<String>();
list.addAll(set);
```

4. (2 pts) If we have an array of elements of type int, then we would step through the elements of the array and print each one by using the following code.

```java
for (int counter=0;counter<array.length;counter++)
    System.out.println(array[counter]);
```

Suppose we have a List named list that contains elements of type Integer. Show how you would translate this loop into one that would step through the elements of the list and print them out.

```java
for (int counter=0;counter<list.size();counter++)
    System.out.println(list.get(counter));
```
5. (2 pts) Suppose we have a Map defined as Map<String, String> map = new HashMap<String, String>(). Show how you would display all of the keys and values stored in the Map.

```java
Iterator<String> iterator = map.keySet().iterator();
while (iterator.hasNext()) {
    String key = iterator.next();
    String value = map.get(key);
    System.out.println(key + " " + value);
}
```