Sorting
Mechanics of Sorting in Java

- Java has built-in ways to sort arrays and ArrayLists:
  
  ```java
  Arrays.sort(myArray);
  Collections.sort(myArrayList);
  ```

- Note that these are static methods in the Arrays and Collections classes, **NOT** instance methods!
  
  ```java
  NOT: myArray.sort();
  NOT: myList.sort();
  ```
Sorting Example

- `temperatureArray:`

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>51.7</td>
<td>33.9</td>
<td>21.6</td>
<td>62.1</td>
<td>59.0</td>
<td>44.5</td>
</tr>
</tbody>
</table>

- **After** `Arrays.sort(temperatureArray):`

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21.6</td>
<td>33.9</td>
<td>44.5</td>
<td>51.7</td>
<td>59.0</td>
<td>62.1</td>
</tr>
</tbody>
</table>
Sorting Example

- `temperatureArrayList`: 
  
  | 51.7 | 33.9 | 21.6 | 62.1 | 59.0 | 44.5 |

- **After**

  Collections.sort(temperatureArrayList):

  | 21.6 | 33.9 | 44.5 | 51.7 | 59.0 | 62.1 |
Types of things you can sort

- The `Arrays.sort()` method applies to arrays, but arrays have types.
  - E.g., `int[]`, `double[]`, `Object[]`, `Point[]`

- `Arrays.sort()` can sort:
  - any primitive array type
  - any Object array type that implements the `Comparable<T>` interface.
The `Collections.sort()` method applies to ArrayLists, but ArrayLists have types.

- E.g., `ArrayList<Integer>`, `ArrayList<Object>`, `ArrayList<String[]>`

Likewise, `Collections.sort()` can sort:

- `ArrayList<T>`, if `T` implements the `Comparable<T>` interface
The Comparable<T> Interface

- The interface declares one public method:
  ```java
  public int compareTo(T other);
  ```

- What does it mean for a class to implement this interface?
  1. The class must declare that it implements it
     e.g.: `public class BankAccount
           implements Comparable<BankAccount>`
  2. The class must actually do the dirty work of implementing it
Example

public class BankAccount

    implements Comparable<BankAccount>

{

    private int balance = 0;

    ...

    public int compareTo(BankAccount other) {
        if(other==null) {
            return
                -1; // null objects less than
                // everything else
        }

        return other.balance - this.balance;
    }

}
Example, continued

```java
BankAccount ba1 = new BankAccount(100);
BankAccount ba2 = new BankAccount(10000);

System.out.println(ba1.compareTo(ba2));
System.out.println(ba2.compareTo(ba1));
```

Output:
99900
-99900
Exercise

- Implement an Exam class with
  - A private field for the score (an int)
  - A constructor that initializes the score
  - An “implements” statement for the Comparable interface
  - An implementation of the compareTo method
  - A main method that creates an array of 10 Exam objects, sorts them, and prints them.