Sorting and Searching in the Java Library - Sorting

- You do not need to write sorting and searching algorithms
  - Use methods in the Arrays and Collections classes
- The `Arrays` class contains static sort methods.
- To sort an array of integers:

```
int[] a = . . . ;
Arrays.sort(a);
```

  - That sort method uses the Quicksort algorithm (see Special Topic 14.3).
- To sort an `ArrayList`, use `Collections.sort`

```
 ArrayList<String> names = . . .;
Collections.sort(names);
```

  - Uses merge sort algorithm

Sorting and Searching in the Java Library – Binary Search

- `Arrays` and `Collections` classes contain static `binarySearch` methods.
  - If the element is not found, returns -*k* - 1
  - Where *k* is the position before which the element should be inserted
- For example

```
int[] a = { 1, 4, 9 };
int v = 7;
int pos = Arrays.binarySearch(a, v);
// Returns -3; v should be inserted before position 2
```

Comparing Objects

- `Arrays.sort` sorts objects of classes that implement `Comparable` interface:

  ```
  public interface Comparable
  {
     int compareTo(Object otherObject);
  }
  ```

- The call `a.compareTo(b)` returns
  - A negative number if `a` should come before `b`
  - 0 if `a` and `b` are the same
  - A positive number otherwise

Comparing Objects

- Several classes in Java (e.g. String and Date) implement Comparable.
- You can implement Comparable interface for your own classes.
- The Country class could implement Comparable:

```
public class Country implements Comparable<Country>
{
   public int compareTo(Country other)
   {
      if (this.area > other.area) return 1;
      if (this.area < other.area) return -1;
      return 0;
   }
}
```

Comparing Objects

- You could pass an array of countries to `Arrays.sort`

```
Country[] countries = new Country[n];
// Add countries
Arrays.sort(countries); // Sorts by increasing area
```

Self Check 14.27

Why can't the `Arrays.sort` method sort an array of `Rectangle` objects?

Answer: The `Rectangle` class does not implement the `Comparable` interface.

Self Check 14.28

What steps would you need to take to sort an array of `BankAccount` objects by increasing balance?

Answer: The `BankAccount` class needs to implement the `Comparable` interface. Its `compareTo` method must compare the bank balances.

The Comparator Interface

- Sometimes want to sort an array/list of objects where objects belong to class that does not implement Comparable (e.g. Rectangle)

- Sometimes want to sort using different *key* value (e.g. sort Country based on population rather than area)

- Don't want to rewrite compareTo() method using different key value and recompile!

- Solution: use Comparator interface

The Comparator Interface

- Arrays.sort() and Collections.sort() have a version that accept an object whose class implements the Comparator interface

- See example Country