Static again

Static methods
- **static method**: Part of a class, not part of an object.
  - good for code related to a class but not to each object’s state
  - does not understand the implicit parameter `this`; therefore, cannot access an object’s instance variables
  - if `public`, can be called from inside or outside the class

Static variables
- **static**: Part of a class, rather than part of an object.
  - Classes can have static variables.
  - Static fields are not replicated in each object; a single variable is shared by all objects of that class.
  - `private static type name;`
  - or, `private static type name = value;`
  - **Example**: `private static int count = 0;`

Example
```java
import java.util.Random;
public class Person {
    private static Random generator = new Random();
    public Person () { }
    public boolean vote() {
        return generator.nextBoolean();
    }
}
```

Another Example
```java
public class SortablePerson implements Comparable<SortablePerson>{
    private String firstName, lastName;
    private String info;
    private static boolean sortByLast = true;
    ``