Interface Exercises

These exercises are for your own benefit, and are entirely optional. Feel free to collaborate and share your answers with other students. These problems are taken from various sources at Cornell and on the Internet, too numerous to cite individually. Some hints are at the bottom of the page.

1. Write a definition for "interface Collection". Try to think of as many methods as you think belong to the Collection interface, and what their signatures should be. Make sure each of your methods makes sense for *any* (or most, at least) Collection implementation, such as a linked list, a tree, an array, etc. Now compare your definition to the one in java.util.Collection. (There will be certain functionality you may not yet understand – exceptions, and iterators, for example).

2. Rewrite the class List so that it implements your Collection interface.

3. Rewrite the class Tree so that it implements your Collection interface. Data items should be stored at each cell in the tree. When adding a new element, it is up to you to put it in the tree in a "good" place. E.g., should it go at the leaves? Or can you find an empty spot higher up in the tree?

4. Rewrite the class Tree so that it implements your Collection interface. Data items should be stored only at the leaves of the tree.

5. Does the following definition make sense? Why?

```
public interface Clever { }
```

Hints & Spoilers:

5. Take a look at java.lang.Clonable or Java.io.Serializable.