

### CSE 331 23su – Section 3 – Test Design

Suppose we have a class that stores a sorted list of Strings, possibly containing duplicates. When a new String is added to the list, it is inserted in the proper place so that the strings in the list are maintained in non-decreasing order. Here is the interface:

```
public class SortedStringList {
    public SortedStringList() {...}

    public boolean isEmpty() {...}
    public int size() {...}
    public int indexOf(String elt) {...} // return -1 if not found
    public String get(int pos) {...}    // first position is 0,
        // throw IndexOutOfBoundsException if pos<0 or pos>=size()

    public void add(String elt) {...}
}
```

Your job is to create a test suite for the `SortedStringList`. Your tests should only use the operations given above. Start with the most basic tests you can think of, and, with each successive test, marginally increase the complexity of the test over the preceding tests. For each test, you only need to describe the test setup (inputs) and expected result(s) (outputs). You should not write JUnit code – just give the basic logic of the tests.

Remember: each test should, as much as possible, only test for one new thing, so that if a test fails, the failure provides specific clues for where the error might be located. A first example is given below to help get started.

Input:                    create new empty list `s`  
Expected output:        `s.size() = 0`