

# 211hw8

## Homework 08 (Due 10:00am Friday, March 19, 2010)

Submit via Owl-Space

### Preliminaries

This homework on manipulating Java lists must be done using the Intermediate language level of DrJava. We are providing skeleton classes for each problem with unimplemented methods for you to write. Each such class includes a method with a name like `listString()` that converts lists to more readable String notation similar to that used in Scheme.

### Composite Design Pattern for List

1. Write the `arrangements` (permutations) function from HTDP problem 12.4.2 in HW2 as a Java method for a `Word` class provided in the file `Word.djl`. This file includes definitions of the composite pattern classes `Word` and `WordList`. Decompose the problem in exactly the same form as [this solution](#) to problem 12.4.2. We are providing skeletons for the classes `Word` and `WordList` in the file `WordList.dj1`; use them.
2. Write the `mergesort` function from the last problem in HW4 (using exactly the same top-down approach described in HW4) as a Java method in the composite pattern class `ComparableList` provided in the file `ComparableList.dj1`.
3. Do Exercise 21.2.3 from HTDP using the Java composite pattern class `ObjectList` provided in the file `ObjectList.dj1`. This file includes the interface `Predicate`, which is the type of Java function arguments passed to the `filter` method, and abstract method stubs for `filter`, `eliminateExp`, `recall`, and `selection`. Exercise 21.2.3 provides Scheme code for the `filter` function which you should directly translate to the corresponding Java method code in `ObjectList`. Note that your `filter` method should work for arbitrary `ObjectLists`. In coding the methods `eliminateExp`, `recall`, and `selection`, use the Java type `Number` (compared using method `doubleValue` in place of the Scheme number type and the Java type `Object` in place of the Scheme `symbol` type. The method `doubleValue()` in `Number` returns the value of this converted to a `double`. You will need to cast the `Object` input of the `test` method to type `Number` when filtering lists of numbers.
4. Do Problem 2 from Homework 5 in Java using the composite hierarchy of classes provided in the file `ArithExpr.dj1`.