CS 635 Advanced OO Design and Programming
Spring Semester, 2013
Assignment 4

San Diego State University -- This page last updated 4/16/13

Due May 2

1. Flyweight with a flyweight pool. String comparison can be slow. Strings in Java are immutable so there is no reason to have more than one instance of a given string in a program. Implement a flyweight pool for strings without using the builtin string flyweight pool. Now when using a string your code can use the flyweight instance of the string. This will allow you to use pointer comparison rather than string comparison. Your flyweight pool should allow flyweights to be garbage collected if the program no longer has a reference to a given flyweight. In C++ strings are mutable, so add a decorator to make the flyweight immutable.

2. Decorator. Implement two separate decorators for an ArrayList. The first decorator should make the ArrayList immutable (Don't use the build in decorator for this). The second decorator allows one to treat the one dimensional ArrayList as a two dimensional ArrayList. That is one can create a normal ArrayList and then treat it as two dimensional. For example we may want to treat it as containing 5 columns and a growing number of rows. Then you should be able to add an element to the third column in the second row. Of course that would be location 7 in the underlying ArrayList. But the decorator should convert the column and row indices into the correct location in ArrayList not the client. Can you make your decorators so that they can be composed?

3. Adapter. Implement an adapter that converts a Java Enumeration into a Java Iterator.

Grading

Item	Percent of Grade
Working Code	20%
Unit Tests	10%
Proper implementation of Patterns	60%
Quality of Code	10%