

# CS 251: Intermediate Software Design

Program Assignment 4  
Due Wednesday, March 28<sup>th</sup>, 2007

This programming assignment focuses upon using a variety of patterns to implement a program that will build and traverse a binary tree using various traversal strategies. The patterns will be used as follows:

- **Singleton**, which is used to implement an `Options` singleton that parses and keeps track of the command-line options.
- **Strategy**, which is used to implement the appropriate type of queueing strategy (such as `AQueue` or `LQueue`) and traversal strategy (such as level-order, in order, pre order, and post order).
- **Adapter**, which will be used to integrate your `LQueue` and `AQueue` into a new `Queue` and `Queue.Adapter` class hierarchy that can be used to dynamically select which type of queueing strategy to use in the program at runtime. The use of `Adapter` ensures that no changes are required to the existing `LQueue` and `AQueue` classes.
- **Factory**, which is used to create the appropriate types of queueing and traversal strategies indicated by the `Options` Singleton.

Moreover, graduate students need to implement the following additional patterns (which are optional for undergraduates):

- **Abstract Factory** and **Factory Method**, which are used instead of individual `Factory` functions to consolidate all the factories into a single concrete factory class.
- **Bridge**, which is used to avoid exposing “naked” pointers and to simplify memory management, e.g., by reference counting throughout the program.

## Getting Started

You can get the “shell” for the program from [www.cs.wustl.edu/~schmidt/cs251/assignment4](http://www.cs.wustl.edu/~schmidt/cs251/assignment4). There is a single file called `main.cpp` that are partially filled out for you to give you the gist of how to implement the program. Please note, however, that this file is *very* incomplete, *i.e.*, it just sketches the solution and many not even compile, so just use it as a rough draft for your solution and feel free to modify any or all of it as you see fit. You’ll also need to create the appropriate `Makefile` or project file. Note that you’ll need to reuse the files from your `AQueue`, `Array`, and `LQueue` implementations.