Kyle Zalewski

CS202

Programming Assignment 1 Design

For Programming Assignment One, I intend to employ an ideology that was discussed in class wherein the idea of the “node” is traded in favor of a singular object that represents the beginning of a list. I think this will help simplify my design and remove emphasis from the data structure, which will allow me to focus more directly on object-oriented concepts rather than the data structure itself. For previous programs, my structure has included a couple necessary containing relationships. If I adopt the mentioned philosophy successfully, I believe that I can manage to only include specialization relationships between my classes. This will hopefully result in fewer classes overall and a simpler design.

In order to reduce the need for setter and getter functions, my classes will be derived from a common base class, which can manipulate the data and pass it down the hierarchy. My base class will be called Segment, and will be responsible for a particular portion of a road to the mountain. Derived from Segment will be Road, which will represent the collection of segments that make up a route. From road, three additional classes will be derived. These will be called WeatherDelay, Crowd, and Suggestion. These classes will be responsible for gathering and aggregating data from their parent class and returning output to the user.

WeatherDelay will take information from the road class and analyze it to determine how much delay will be caused on a particular route due to weather anomalies. Crowd will do the same but for the overall volume of traffic currently on the road and possibly at each resort to help the user make informed plans. The final derived class, Suggestion, will take aggregated data from the other two child classes and calculate a suggested path to return to the user after analyzing all important considerations regarding travel to and from the mountain.

As I work through making this program, I may end up needing to add more classes, as five classes seems like a small amount for the expected functionality. There is quite a lot of information that will potentially need to be considered for each recommendation, so perhaps splitting the load between two classes rather than one may be necessary.

For every class, I will write a default constructor, and for every class that houses dynamic memory (any character array or dynamically allocated array of any type) will also need a copy constructor (with arguments) and a destructor to ensure that memory is both allocated and properly deallocated when the necessity arises. Any class that uses English “strings” will thus need these functions to support the use of cstrings. All of my classes will take care of their own calculations concerning their own data, and only pass processed data to other classes.

As I learned in Lab 1, the use of initialization lists will help me to greatly reduce the amount of code needed to make this program work properly. The use of these lists allows me to initialize both the current class and its parent at the time the child object is created, reducing the amount of coding needed significantly, while also ensuring that I do not run into uninitialized variables or unallocated memory. One of my main goals for this assignment is to solidify my understanding of initialization lists and where they are necessary and helpful. I believe they will help greatly in all future assignments, as one issue I’ve run into in the past is effectively initializing all data in a hierarchy without the need for long chains of function calls reaching all the way back to the most indirect base class.