CS202

Program #2

Design Analysis

Hyunchan Kim

As long as we have learned from derivation to dynamic binding based on object oriented programming, this assignment is going to be building a hierarchy using dynamic binding with data structures. This program is composed of several classes from a rough design. The classes are basically composed of a menu class and ingredient class which holds the menu and ingredients data that comprises the restaurant program. And the data is dealt with a binary search tree using linear linked list. Every node holds a linear linked list that contains ingredient data. This class and its associated members form a base class hierarchy from which the binary tree node derives from. Further the design contains a linear linked list node class where dynamic binding is used. Dynamic binding is used mostly in the menu classes like entrees, appetizer, and drinks. The base class provides a hierarchy where dynamic binding might be used. The design has lack an abstract base class with pure virtual functions this was mostly done out of time constraints after having designed the program erroneously to begin with. Some of the objects in the design use other objects that are contained as members to instantiate other member functions, which in this sense may not be an object oriented approach. For instance, once a list is made in the binary tree node class the functions such as ingredient data satisfaction is updated. Each data structure has the full implementations of insert, retrieve, delete, display and etc. This was done as program requirements most program requirements are satisfied but the pure abstract base class function is not one of those. And one of the most important factor is that the binary search tree is balanced. This data structure may be used for the rest of the term.

What I think is inefficient is dealing with the balanced binary search tree. It was really hard to design the balanced tree at the first time but I fairly got used to it after understanding in-order successor and other notions. And also, implementing a linear linked list to every node of binary search tree is quite confusing so I had to focus on how that would work and draw a design. It would be better for the future assignment using this data structure. If I were more confident with the balanced tree I would have developed them more effectively and made more useful functions. I might have used a different data structure if I could get to choose. It would probably be linked list of arrays. In my opinion, it’s easier to

handle the data with menu and ingredients. Now I presume that I am pretty familiar with my design hierarchy and met the dynamic binding requirement. Each class had a clear task and responsibility.