Andy Fleischer

Prof. Karla Fant

CS202

27 August 2021

Program 4-5 OOP Writeup

For program 4-5, we had to design a food-ordering system with different types of menus in a hierarchy. The menus had to implement an array of DLLs to allow for moving between sections of the menu for ordering. For part 2, I had to implement a balanced binary search tree (I chose to do a red black tree) of linear linked lists (for identical keys). I believe that my design followed these directions and was object oriented.

For the menu hierarchy, I decided to make an abstract base class called “Menu” from which my “Italian”, “Ramen”, and “Thai” were derived. Each of them implement their order packaging (which I will talk more about later), help menu, and cost calculation differently. But, they all have similar ways of reading from the external data file and ordering. To incorporate the array of DLL, I decided main would have an array of menus and menu would derive from a DLL. That way, the menu could easily call it’s own go_forward, go_backward, display, etc. DLL functions with no wrappers. Then, each DNode of the DLL inherited from another class Section which represented one section of a menu (appetizers, entrees, desserts, etc.). Section had a title and two maps: one to map the string of an item name to a float for its price, and one to map the item name string to an integer for how many the customer ordered. Also, Section can display, add and remove items, and package the order. I think this was very object oriented, since each class has their own data and tasks, and I don’t have classes handling other classes’ data, every class has a purpose.

To make the tree more object oriented and simpler, I made the tree nodes only have a String for the name (to alphabetize on) and one String for the order. My “package order” functions condense all of the data from the maps into one string with all of the items and the overall price. This is more object oriented, since a data structure should not have to worry about how to display its data or have complex data like a hash table. The display functions are much easier since I can just print out the strings and be done.