Juhwan Lee

Professor Karla Fant

Programming Systems

October 15th 2020

Program #1 Efficiency Write-up

The first programming assignment was very difficult. It was so different from the programming approach that I've been using so far that it took a lot of time to complete the program. I thought I understood what object-oriented programming is after watching several videos, but when I actually programmed it, I realized that I didn't fully understand it yet. I concentrated on forming a single inheritance, creating relief efforts class and three sub-classes (ex. housing class, food class, clothing class), but the node class and circular linked list class became very complex and inefficient structure. Both the circular linked list and linear linked list were programmed to use recursion, but I could not figure out how to unify a node class into one (ex. relief\_item \* data), so I decided to divide it into three different node classes (ex. housing data, food data, clothing data), and it caused circular linked list class to have so many functions. With the exception of problems in node class (using public data and separated into three classes) and circular link list class (having too many functions), overall my program meets the requirements of the first program. First of all, relief effort is the base class (I named it effort class) and there are three derived classes and those are housing class, food class, and clothing class. Common data member of derived classes (in my program is char \* name) should move up to the base class and it can be managed by base class member functions which can reduce the amount of code of sub classes. Moreover, all the constructor, copy constructor, and destructor of base class and derived classes were implemented with the use of correct initialization list and checked that it has no double free error or memory leak. And linear linked list class was implemented with recursion and it manages the browsing history data. And when the user try to display browsing history, it displays in order from high search frequency to low search frequency by using sorting list function (sort\_list()). This function re-orders all the nodes in the linear linked list from high frequency to low frequency and this function was also implemented recursively. By designating node class data as protected data rather than public data and consolidating node class into one, I can reduce the amount of functions in circular linked list class, however I could not figure out how to do it yet and so I decided to complete the program in this way.