**Program 1 Design**

**Overview of Program**

The goal of this program is to catalog and organize various relief efforts. It will allow the user to add different types of assistance to the program as well as have a separate functionality allowing a user to look up what relief efforts have already been added to the program. There will be three types of relief efforts; a provisions effort that is related towards food and drinks, a housing effort that will maintain lodging information, and a clothing effort that will allow the user access to various clothing that has been donated. Additionally, there will also be a search history functionality that will keep track of items that users of the program might look up.

**Base Class and Derived Classes**

The program will be implemented utilizing Object Oriented Programming. The base class of the program will be a *relief effort*. The base class itself will be able to hold the user’s name and information. The functions that will be used are a read function to prompt and read in from the user, a display function that will display the users information, a change name function that will allow the user to change their name, a change address function that will allow them to change their address and an alter dependents function that will allow them to add or subtract dependents as needed. There will be four derived classes from this base class, the three classes will be the three types of relief efforts: provisions, housing, and clothing, as well as a search history class that will log information searched for by the user.

The *provisions* class will maintain information on what kind of food or drink it is and will also be allotted an expiration date. The functions that will be used are a read function to prompt and read in from the user, a display function that will display any given provision, as change clothing function that will allow the user to change an article of clothing that has been previously selected, a check expiration function that will further inform the user on their provisions expiration date and a compare quantity function that will compare the amount of provisions to the amount of dependents.

The second the derived class, *housing*, will maintain all information involving lodging for those who are providing it or may need it. The class will contain various locational information about the lodging, as well as how many rooms will be available, and how many beds are open. The functions that will be used are a read function to prompt and read in from the user, a display function that will display any given clothing, a change clothing class that will allow the user to change a piece of clothing previously selected, a check weather function that will further inform the user on the appropriateness of the clothing based of the weather and a compare zip code function that will compare the user’s zip code to the one chosen in the relief effort.

The last relief effort type class from the relief effort base class will be the *clothing* class. This class will catalog all the clothing being dropped off for donation. The clothing class will maintain what type of clothing the donation is, what size it is, and for what kind of weather the clothing is best suited for. The functions that will be used are a read function to prompt and read in from the user, a display function that will display any given house, a change housing class that will allow the user to change information about a house previously selected, a check occupancy function that will further inform the user on the maximum amount of occupants in a house and a compare size function that will compare the user’s age to the general size of the clothing.

The final derived class will be the *search history* class. This class will log all of the different searches that the user makes throughout the program. The information will be stored into a linear linked list node that will be ordered by frequency of searches as well be created and manages by its own *search history list* class. The functions of the list class will be Add, Remove, Display All, Display Type and Retrieve and the list will be created recursively.

Due to the relief efforts being derived from the relief effort base class and the node class being derived from the different relief effort types, no setters or getters will need to be used when needing to enter or change any information involving these classes. Through use of the hierarchy, the classes will be themselves “plus” their parents, thus removing the need to shuttle data back and forth between them and enable the use of derived classes using their parents member functions to alter the base classes data as needed.

**Data Structures**

The program will be implemented using a circular linked list for each relief effort type. This will be accomplished by the use of a node class. Each of the three relief efforts will have their own circular linked list node class. The node classes will be derived from the relief efforts. Additionally, there will also be a list manager class who will have a “has a” relationship with all of the node classes. The primary role is to manage the nodes from the circular linked list class and facilitate the creation and militainment of the data structure.

The list class will be the only class that utilizes setters and getters. This class does not contain the information that is stored in each node, it only manages it. This necessitates the use of setters and getters construct the circular linked list with its next and rear pointers. The functions of the list class will be Add, Remove, Display All, Display Type and Retrieve, there will be three versions of each of these functions, one for each relief effort type as well as a Wrapper function to enable abstraction. The lists will be created recursively.

**Summary**

This program will be created, through the use of Object Oriented Programming. There will be a base class named relief effort that has 3 derived classed, each being their own specific type of relief effort. Each of those classes will have their own derived class children this is a node of a Circular Linked List class. The CLL classes will be managed by a list class with a “has a” relationship. Utilizing single inheritance, the program will create a systematic process for the user to access the relief effort information.