Programming Assignment #1
CS 202 Programming Systems

***** Make sure to read the Background Information first!**
It applies to all programming assignments this term***

Always backup your files PRIOR to using tar
Double check that the arguments specified with for tar are correct!
Late penalties will apply to all submissions incorrectly archived/uploaded

Background for All Programming Assignments:
When beginning with this project, the first thing to keep in mind is that we are no longer working on CS163 programs! In CS163 we were concerned about creating Abstract Data Types and the class construct facilitated this. Instead, this term we will be focusing on how to create Object Oriented Solutions. An ADT may be part of that solution – but it certainly shouldn't be the primary focus. Instead you want to strive for classes to have specific "jobs" and have classes derived from more general classes, whenever appropriate. We will be working in situations where there are multiple classes, so you will want to focus on dividing the design into smaller components that have specific jobs working together to solve the problem.

Every assignment this term needs to have at least 5 classes. With these, think about how to design the classes such that they reduce the amount of work another class needs to do. The idea is if we have "robot" like classes doing the smaller tasks or "jobs", that by the time we get to a larger class that has more to manage – it will have little left to do! We can achieve this by delegating. Often the over-use of "getters" can cause the opposite to happen – and instead of delegating the managing class has to fundamentally do all of the work itself.

Program #1 Overview:
With the wildfires, we find ourselves in an unusual situation of having many people in desperate need of assistance. So many devastating wildfires. That on top of the pandemic may have left people needing assistance in a way they never would have previously predicted. You may have watched (or been part of) the relief efforts giving water, food, clothing, and shelter to those in need. People have donated their RVs and homes. Truly this is the strength of Oregonians coming together. But imagine the chaos that could take place at donation centers. We need software to help! This is where our story begins…

Program #1

For Program #1, you will be creating an object oriented program that will support emergency relief efforts. We want to allow easy cataloging of donations dropped off or made available and then allow those in need to browse what is available; to make this useful, we will want to keep a history of what people have searched for. To manage the scope of this project, the following are the minimum requirements:

1. Support three different types of relief efforts, one of which needs to be housing. The other two types can be of your design. Ideas could be supplies, food, clothing, and many more.
2. At least one of the types supported needs to support the notion of an expiration date.
3. At least one of the types supported should interact with another class type

The main point of this assignment is to make the development simpler through the use of OO techniques. Therefore, first think about breaking this down into a series of classes and create them independent of the entire problem. Then, push up the common elements into a common base class; this is the important part! And lastly, avoid classes with only setters and getters with the exception of a node class!

Here are some possible classes to consider in your design:

1. Three classes that represent the three types of relief efforts you are going to support
2. The base class that contains all of the common elements (data members and functions) (topic 1)
3. There should be at least 5 functions for each of the three classes that manage the different relief efforts, besides constructors and destructors
4. Every class that manages dynamic memory needs a default constructor, a copy constructor (week #2 topic 3) and a destructor.
5. At least one single inheritance hierarchy is required supporting public derivation (topic1 and topic 2)

Keep classes small and functions small. A large class or function means that the problem has not yet been broken down into its basic components (objects).

Required Data Structures

<u>All repetitive data structure algorithms</u> must be implemented using recursion in CS202. Implementation of the data structures requires full support of insert, removal, display, retrieval, and remove-all.

1. You will need three different data structures to hold the different type of relief items. Let’s use a circular linked list. You may use this data structure in any way that makes the most sense to you in your design.
2. Create a Linear Linked list to keep track of the browsing history, what searched for when examining the available items. This should be ordered by frequency.