Part 1[2 marks] – When will your shipping container management system have to make a new container (name 2 situations)?

The shipping container management system will have to make a new container when the container is going towards every new destination. Another situation is when the container is full where the weight of goods exceeds the 2000 lb. capacity. Therefore, another container must be made that also has the same destination.

Part 2 [6 marks] - List at least 2 and at most 3 of the classes you plan to create for your shipping container management system. What information (data about packages/containers) will each of your classes deal with (you can discuss your container and package classes here, or any others you choose to create)?

Shipyard class – Primarily, the main data class that also contains the monitoring functions. In doubly linked lists, it keeps track of containers queued in the system where the containers are identified as hold packages information. Another list it holds is all the packages to be placed in the containers.

There is also two other doubly linked lists that hold a shipped out packages and containers list.

BONUS FUNCTION: checks if another package in processing system to the same destination has over 300lbs free, if so try to move excess over capacity to try to save container space. If it can’t ship with the reduction of the light container, leave it as is.

Overview of data requirements in Shipyard class: Shipped Container List, Shipped Package List, In-process container list, In-process container list

Container class – It will hold the container id which is uniquely generated by the system. Other information includes: container id, container weight, destination, number of packages. Each container can hold 2000 lbs. If the packages in a container exceed 2000 lbs, it will create another container to store more packages. Container contents are doubly linked lists that store package contents with respect to having the lightest packages on top. To do this, it will contain a weight sorting function that places the lightest package at the top of the container.

Package class – It will hold a uniquely generated package id for each package. Other details such as package name, weight, and address are also given.

Part 3[7 marks] - Provide the following partial details of your **top level class** for your management system (the one your main program uses to run the entire system):

**Mqueuesys** – Management Queue System will hold functions to operate the shipyard operations. It will serve as an interface to use, display, and change the status of the shipyard, list containers, list packages, and also be able to locate container and package contents/destination. In the Mqueuesys class, it will have the class Shipyard which will contain the majority of data such as Package and Container classes. The Shipyard class will also contain a doubly linked list for shipped containers and a doubly linked list for shipped packages.

**add\_package(pkg\_id, destination, name, weight, address)** – asks user to input details of the package so that it can be added to the system.

pre-conditions: the list of packages is enabled and functioning.

post-conditions: package is now entered into system.

**remove\_package()** - ask user for the package id to remove or else to find package based on destination (user is then given a list of packages based on criteria). A system must be able to remove packages because sometimes packages need to be removed.

pre-conditions: the list of packages available and packages in the system.

post-conditions: package selected is now deleted from the system.

**find\_package()** – To ask user for package id or destination to look up all packages heading towards the destination. This will let us know the details of the package and to verify package information such as cases where we have a discrepancy to confirm.

pre-conditions: the package system is functioning and list of packages is available.

post-conditions: it will find the package and tell the user the package information and whether it has been shipped or not (if it exists in shipped list, being processed or not in the system at all).