Project Classes and Methods -

My project consists of 4 main classes and 2 sub classes.

First my Customer class has an empty constructor, a set type method and a string instance to overload printing. The set type method will return itself as Loyal customer or Bargain hunter, allowing the set type method to be used as aggregation to access the two sub classes. The two sub classes call the super class initiator, ask for a name and have 3 methods each. The Loyal methods checks whether the customer is of class Loyal customer or not, the history list method prints the purchase history if available and the str method overloads with different messages for each subclass.

The Shopping cart class creates a blank dictionary and counter in it’s constructor. It has only one method and the rest being overload methods. The add overload just increments counter and indexes the dictionary with the counter and the value as the passed value. The sub does the same but uses del to remove the item from the dictionary. It also first checks that the item exists in the cart. The str overload just prints the dictionary and is used for when ‘See current shopping cart’ is used. The get\_items method just returns the dictionary.

The Address and Payment class are the same and use a set method to ask for values

For it’s instances. It declares these in their constructors and finally when the str is

Overloaded it prints the information nicely.

User Manual -

To use the program the user must first create a customer. I the user chooses to create a

loyal customer they will have access to extra products and will have the option to view

they’re purchase history asynchronously. After entering a name they will have access to

both list products and add/remove products options. After an item has been added to the

cart successfully they will have access to the checkout and to view their current cart.

When an item is purchased and they are a loyal customer that purchase and it’s details

Will be visible in option 6.

Difficulties/Challenges -

Some difficulties I had with this project was deciding how to make the sub classes of

Customer. I choose composition for them in the end and they both inherit the super class

So they can call it’s str method and add to it. If I chose aggregation the sub classes could exist without the same super class if many customers are made, hence composition.

Another difficulty was with the Change cart function where an item is entered and must

be checked that it is in product list. I had to use Title() string method and I used add and

sub overloads for the cart so it is handled easily in this function. Also I have to return

empty cart because if the user fails to add and item or successfully removes an item and

the cart is empty they must not be allowed to go to checkout etc.