**CSI142\_GroupProject Short Documentation**

The domain scenario in which this app is utilized is a store that provides organic products like fruit and meat. The class structures comprise of private fields that are attributes unique to each object like seasonality (fruit) and cut type (meat). The superclass of the constructor chain (that displays inheritance) is "Product". In the product constructor, fields like the string "name" and the double "weight" which are common to attributes of objects in each subclass, are initialized. The constructor in the subclass "Meat" initializes the unique string "cutType" while that in "Fruit" initializes the boolean field "isSeasonal". Both subclasses initialize the fields in the superclass before initializing each unique field by using the “super" keyword and specifying each variable name. After compilation the user may instantiate objects that will appear in the catalogue in the main app to be viewed by the customer.

The sorting of objects (a feature of the app only visible to employees when the command "java MainApp employee" is entered in the console after compilation) is approached using the "selectionSort" and "insertionSort" utility classes, two classes that implement the sort algorithm. With the “selectionSort” class, the control loops used find the lowest index of the object in the array, the global minimum and swaps elements until the two are equal while with the” insertionSort” the key (the index of an array object) is compared to a variable initialized to equate the array object one position before the key and the key is placed in the right position. The array lists are searched using the linear search utility class where the array object is compared to the “target” (provided they are of the same type) and its availability is indicated (if found, the string "found" is returned and if not the integer -1 is returned).

There are two custom exceptions added to the MainApp. The "LimitExceededException" and the "ProductNotInStockException". These two checked exceptions are implemented by extending the Exception class in the custom classes that define the custom message displayed when the validation criteria are not met. This is done using the keyword super in the constructor of the custom exception class to initialize the string message displayed on the console. The "LimitExceededException" is utilized in the "weightValidation" method, where the condition of the control loop is set and the exception displayed if the condition (stipulated weight exceeded) is true. A similar approach is used in the "stockValidation" method where if a seasonal fruit is selected, the exception is displayed, and the customer is unable to select a weight of the produce they desire to validate (as you cannot select something unavailable).

There are two versions of the main application that can be run. The first one being the customer version, one that can be used after running the command "java MainApp customer” and the latter being the one that can be used after running the command "java MainApp employee". The version in which the customer can check for product availability and select how much of the product they want in accordance with the weight limit stipulated. The customer is prompted to enter what is in the catalogue and to enter how much of it they want. This can be done in any command line terminal and by a user with a novice level skill set. The employee version simply displays inventory information stored and entered manually in arrays. As mentioned above objects can be instantiated and then displayed using the overridden method created in each class (this is done by employees as they have an intermediary understanding of the app). To instantiate, enter the name of the object class followed by the name of the object (this can be anything brief) followed by an equal’s sign, the keyword new, the class name again and parenthesis. The data types of variable attributes defined and initialized in the constructors of each subclass will appear and data that correlates with each data type can be entered manually. To call the display method simply place the name of the object before a period and select the name of the display method you desire.