This program is a simulation of grocery store run. The several methods, three classes, application, and all their attributes work with that idea in mind. The program is first setup the initialization phase where all the needed data in order for it to run is instantiated. From that, then the run begins where the user is greeted and asked to pick a section to purchase from or exit the program via checkout. After, then they are shown the section and then put into a loop where they can display their section or receipt, purchase items, remove brought items, and exit the loop. After such they are brought back to main, where they can move to another section and then do it again or checkout and exit the program.

Though, the rules of it should understood. All inputs from the user are automatically in lowercase, and the user must reply with simple wording (like talking) for the program to work. From these rules, it will be easier to have the program work properly. In many cases, the program will just put you back to where you were due to its looping nature, so it might easier to learn how it works.

The three classes are a table class, a storage class that’s child of table, and an item class. The table class isn’t used in the application but is a primitive version of the storage class that it bases itself of off. The storage class is a child of the table class, stores the purchase data from and for the application, and finds and sends the data that application needs. The purchase method in the application transfers and duplicates the data into the purchase table for storage. The remove method can also remove the data from that table if the item’s amount is at or below zero. These are relevant to the class since these methods in the application work with the storage class to allow the user to purchase items that are stored to be either removed whenever or checked out at the end of the program.