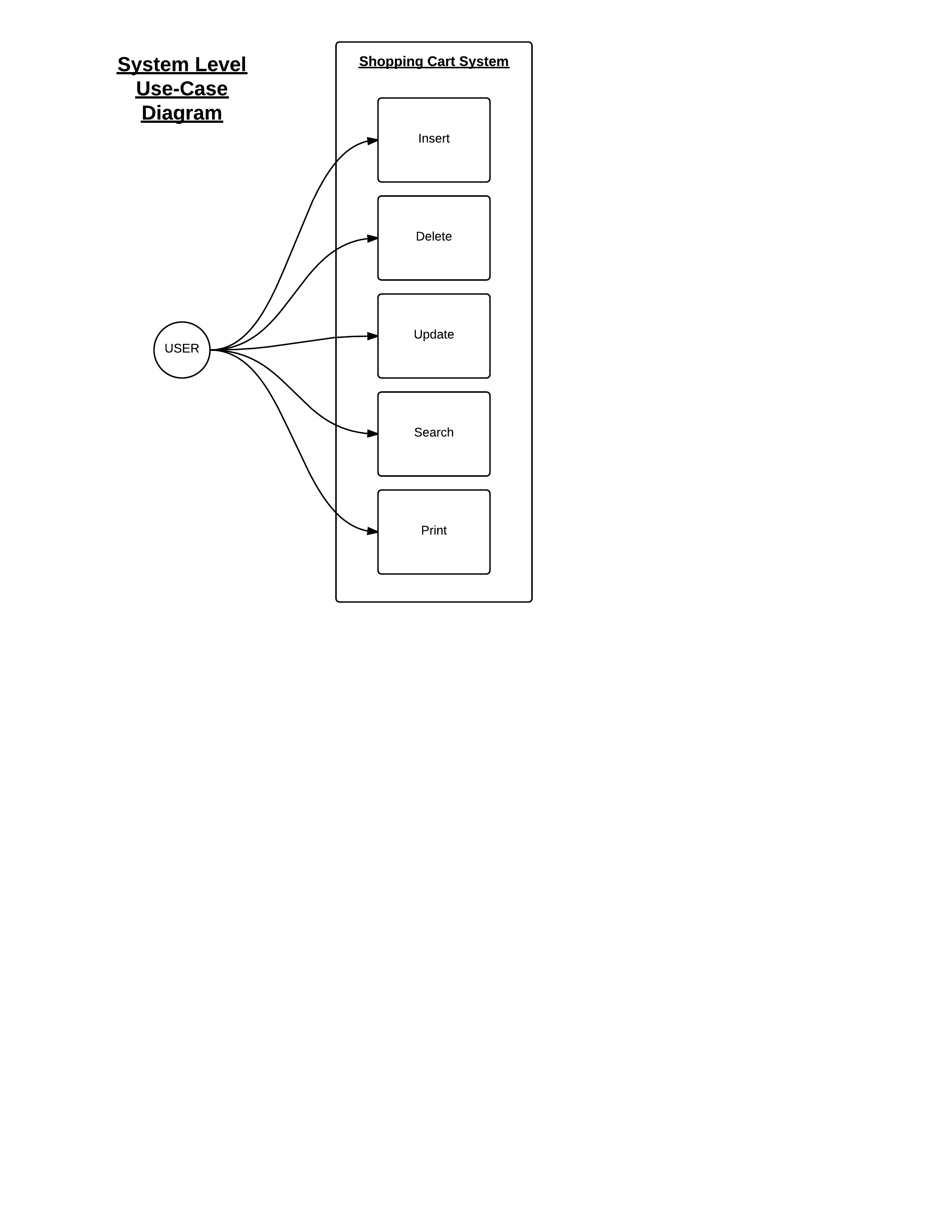
System Level Use Case Diagram:



ADT Level Description of Each Class:

Class: A3Driver

The driver class initializes an arrayList of objects of type “Item” called shoppingCart. This class also initializes an arrayList of type String which contains all inputted instructions into this program. This arrayList is used to process and execute the instructions one at a time, while checking for errors in them.

Class: Item

The item class implements an object that contains the following defining variables:

String name;

**double** unitPrice;

**int** quantity;

**double** itemSubTotal;

**double** shippingPrice;

**double** salesTax;

**int** weight;

**double** totalPrice;

This class performs the desired functions of the shopping cart system (insert, delete, update, search, print). It implements these functions by maintaining and manipulating the above variables.

Class: Electronics

The Electronics class extends the item class and is used for shoppingCart Items that are defined as Electronics. This class implements the special rules that apply to Items defined as Electronics:

“Electronics are taxed differently, based on the state to which they are delivered. The states2 of TX, NM, VA, AZ, AK have no sales tax. Further, electronics may be fragile or non-fragile. Fragile electronics require premium shipping.”

Class: Grocery

The Grocery class extends the item class and is used for shoppingCart Items that are defined as Grocery. This class implements the special rules that apply to Items defined as Grocery:

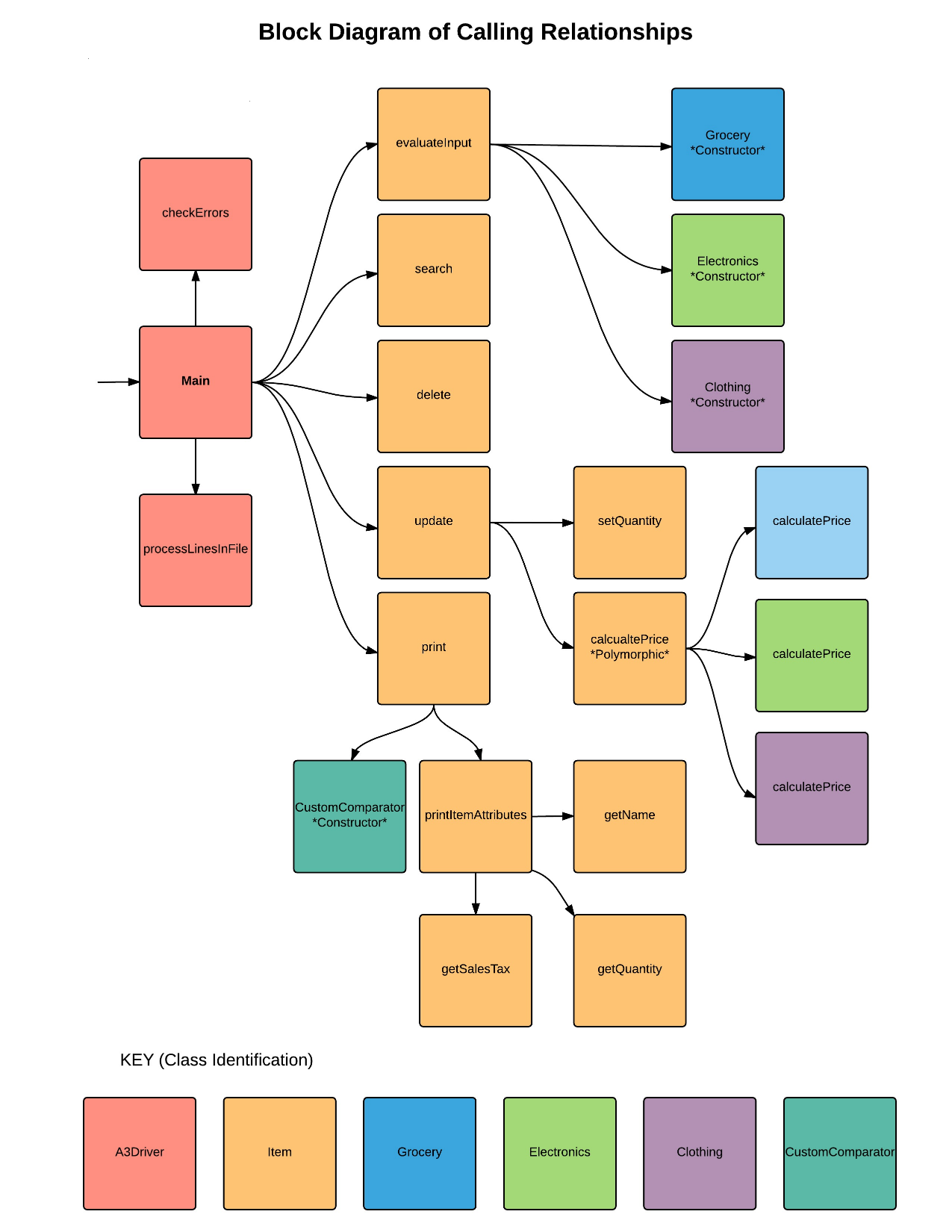
“Groceries are further classified as perishable or non-perishable. Perishable groceries require premium shipping. (Hint: Your derived class for groceries could have a field to indicate whether it is perishable or not.) Further, groceries incur no sales tax.”

Class: Clothing

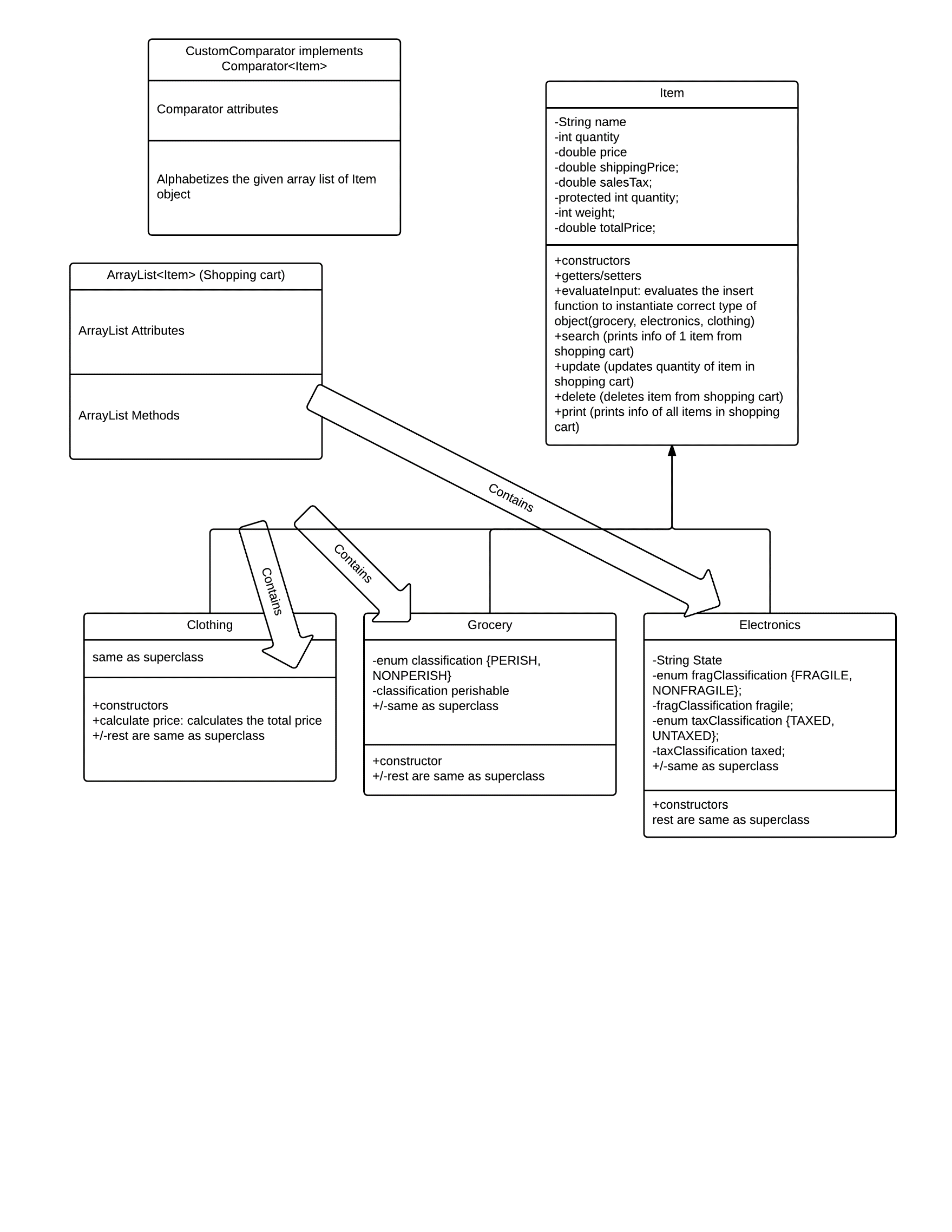
The Clothing class extends the item class and is used for shoppingCart Items that are defined as Clothing. This class implements the special rules that apply to Items defined as Clothing:

“Standard rules apply. Premium shipping is not available.”

Block Diagram:



UML Class Diagram:



High level Algorithm for my Driver (main method):

1. Check if there is a command line input (.txt file with transaction list)
2. If it doesn’t:
   1. Print Incorrect number of command arguments
   2. End program
3. Create an ArrayList of Strings to store each line from the file with the transaction list (this ArrayList is called input)
4. Fill ArrayList with each line from the transaction list file (by calling processLinesInFile method, with input and file name as parameters)
5. Create a new ArrayList of Items called shoppingCart, to store various Items that will be added
6. Make an iterator out of the input ArrayList
7. Iterate through the input ArrayList doing this until there are no more elements:
   1. Split each element of input by the space (“ “) character, and store it in a new static size array; call it splitString
   2. Check splitString for any errors; i.e. make sure the input is valid under the rules described by the model details and problem statement (from the assignment 3 documentation). [Do this by calling the check errors method, with the splitString array as an input]
   3. If there is an error:
      1. Print “Invalid Input”
      2. Go to next element in iterator
   4. If the first element is an insert:
      1. Call evaluateInput method in item class with splitString and shoppingCart as parameters; this will add the proper object to the shoppingCart.
   5. Else If the first element is a search:
      1. Call search method in item class with splitString and shoppingCart as parameters; this will search for the item with the name specified in the splitString that is in the shoppingCart, and it will print the item’s info.
   6. Else If the first element is an delete:
      1. Call the delete method in the Item class; this method will search for all items matching the name in splitString and delete all values.
   7. Else if the first element is an update:
      1. Call the update method in the Item class; this will search for the first occurrence of the item with the name specified in splitString and it will change its quantity to the one specified in splitString
   8. Else if the first element is a print:
      1. Call the print method in the Item class; this will print information of items in shoppingCart and then print the final price