**Assignment III – Shopping Cart**

Alvin Tung – Kevin Yee

1. **Analyzing the Problem**
   1. **Inputs**

String inputs in the following format:

**<Operations> <Category> <Names> <Price> <Quantity> <Weight> <Opt1> <Opt2>**

**Possible Operations:**

*String:*Insert, Search, Delete, Update, Print

**Category:**

*String:* Groceries, Clothing, Electronics String

**Names:**

*String:* Optional

**Price:**

*Double*: Number

**Quantity:**

*Integer*: Number

**Weight:**

*Double*: Number

**Opt 1:**

For Groceries: Perishable or Non-Perishable. For Electronics: Fragile/NonFragile

**Opt 2:** For Electronics: State shipped to

* 1. **Processing the Solution:**

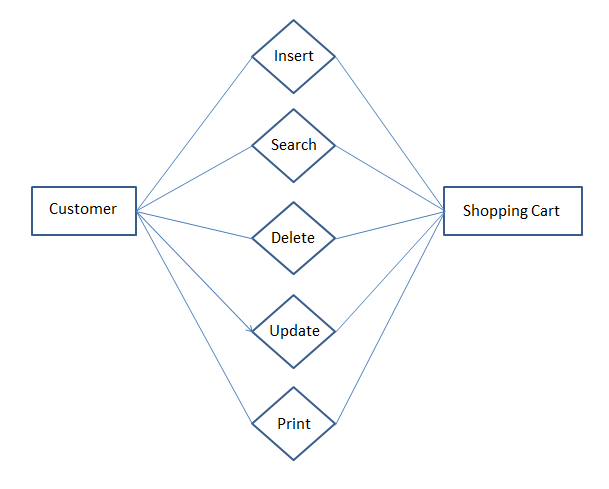
Analyze String Command with the following rules:

1. Determine Operation
2. Determine Item
3. Determine Price
   1. **Modeling Details**

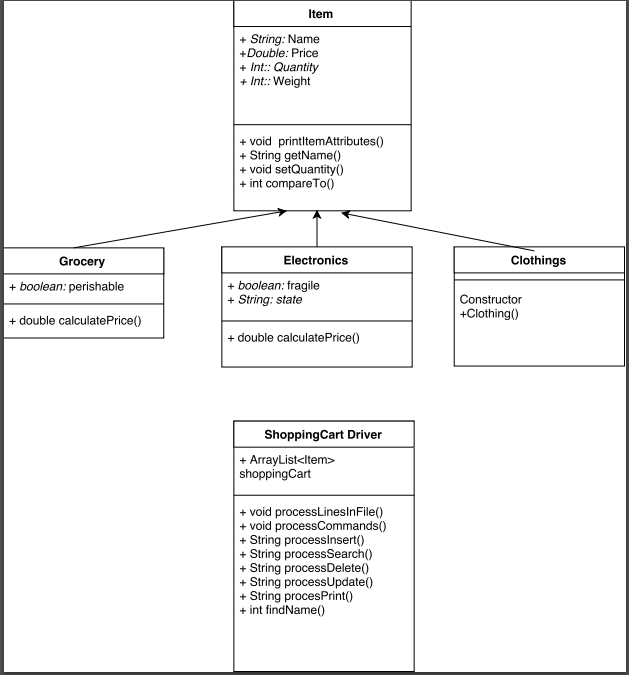
Determining the Price:

* 10% sales Tax on all States Except: TX, NM, VA, AZ, AK
* Shipping Cost Calculated through (20\*Weight)\*Quantity
* Premium Shipping: 20% over standard shipping cost
* Each Items are sold Separately

1. **System Level Use Case Diagram**



1. **UML Model**



+ printItemAttributes()

+ printItemAttributes()

+ setFragile()

+ void printItemAttributes()

+ setPerishable()

**4 ADT Level Description**

**Shopping Cart Driver:**

Parses String Inputs dependent on commands:

* Insert
* Search
* Delete
* Update
* Print

**Items:**

Super Class for Groceries, Clothing, Electronics

Stores Name of Items, Price, Quantity, and Weight

**Groceries:**

Extends Items

Determines if Items are perishable/non-perishable

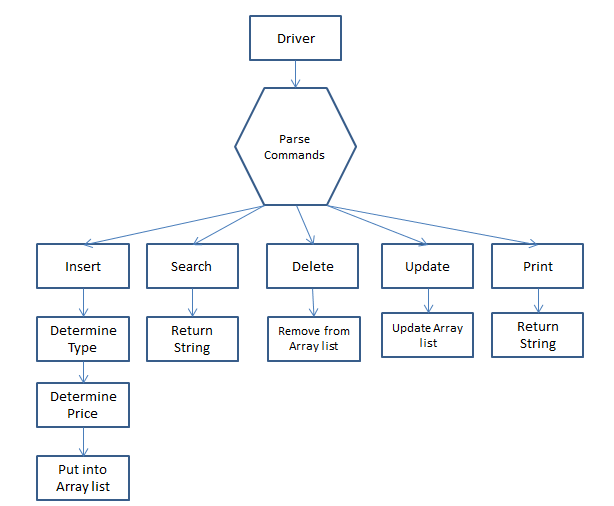
**Electronics:**

Extends Items

Determines if Items are Fragile/Non-Fraigle

Determines which States it must be shipped to.

1. **Functional Block Diagram**



1. **Algorithm**

While readline.next()

Commands[] = readlines.next();

If Commands[0] equals “insert” parseInsert()  
 If Commands[0] equals “delete” parseDelete()

If Commands[0] equals “search” parseSearch()

If Commands[0] equals “update” parseUpdate()

If Commands[0] equals “print” parsePrint()

Return

parseInsert()

GenerateItem(Commands);

Return

parseDelete()

Find item from Arraylist

Remove Index

Return

parseSearch()

Find item from Arraylist

Return Index

parseUpdate()

Find item from Arraylist

Update Values

Return

parsePrint()

Print Arraylist

Return