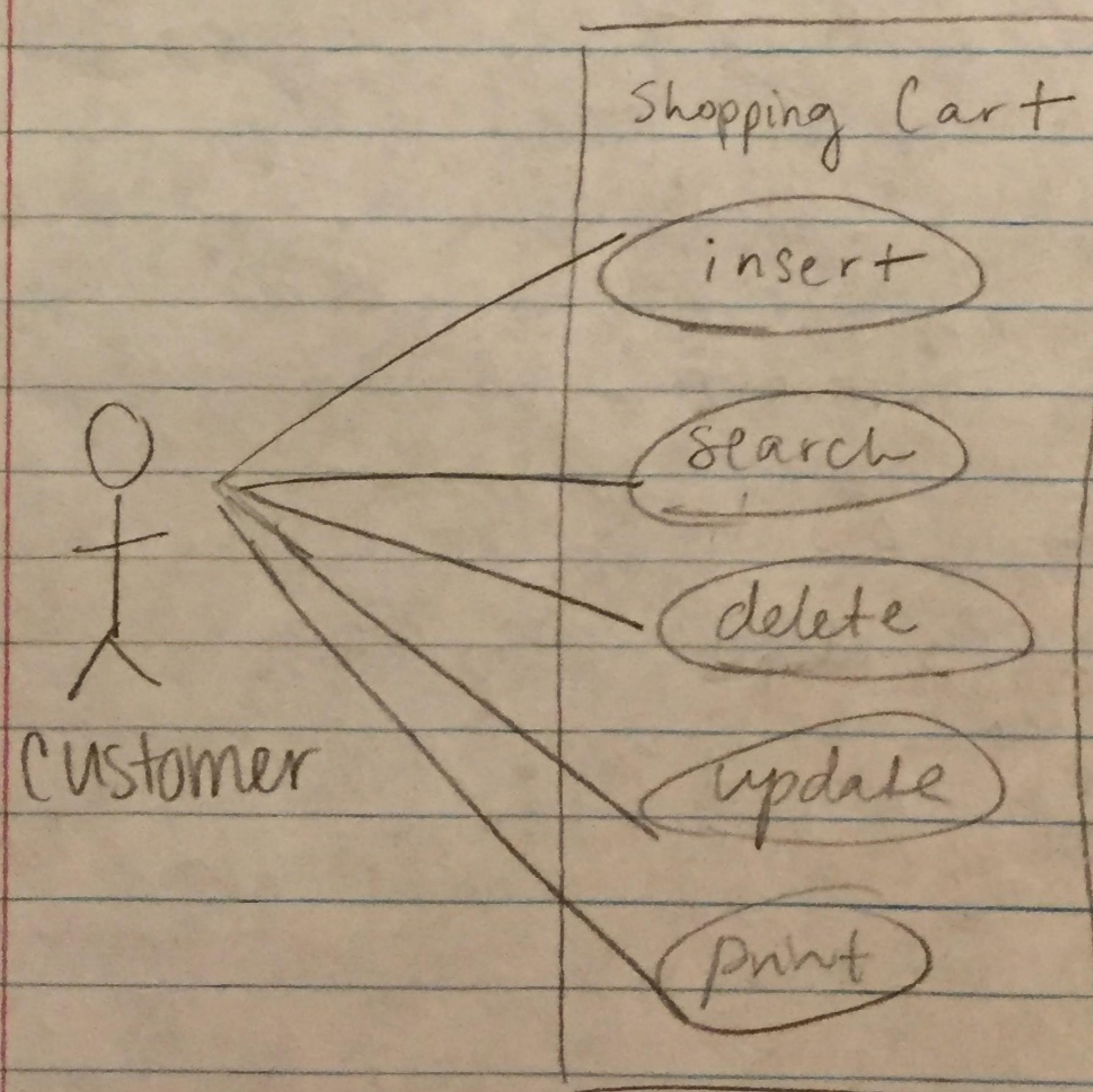


Assignment 3

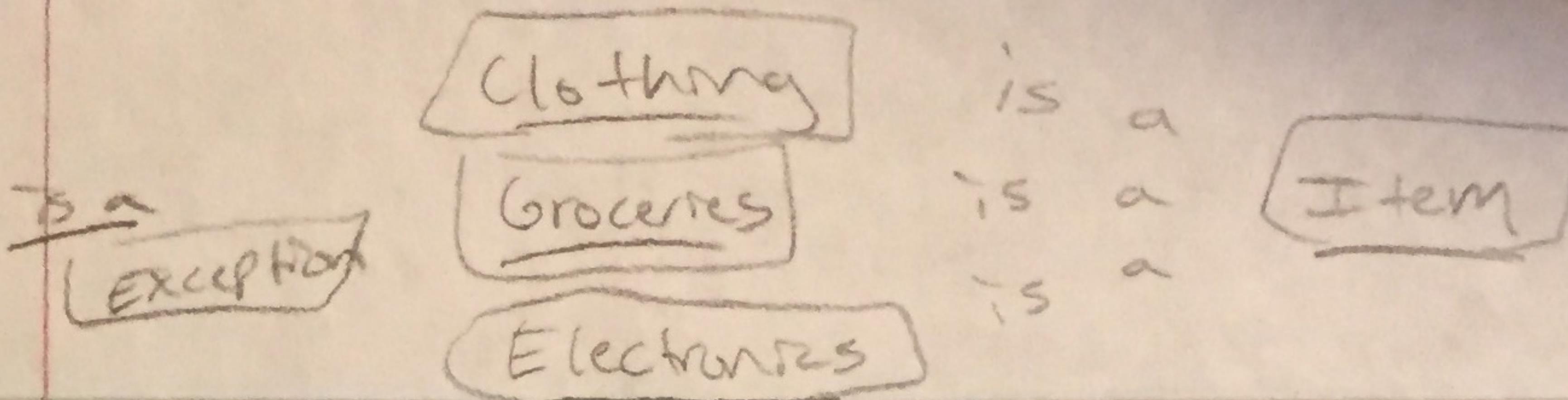
Analysis: We are trying to simulate an online shopping cart. The customer can buy groceries, electronics, and clothing. We have to parse the input string. Then we interpret and calculate the total the customer has to pay for their items using the pricing rules. We are assuming there will be invalid input and will account for that. We are also assuming tax is applied before shipping costs and that items are shipped individually.

Does every output need to be in a GUI?

Design: A System level use case Diagram



Our new exception



A UML Model of the needed classes & their relationship

Item
name: String
price: double
quantity: int
weight: int
getName()
set Name()
getPrice()
setPrice()
getQuantity()
setQuantity()
getWeight()
setWeight()
calculatePrice()
printItemAttributes()

Variables for each item

ADT

Functions for each item

printItemAttributes()

Clothing	Electronics
	fragile: boolean
calculatePrice() printItemAttributes()	state: string isFragile() setFragile() getState() setState() calculatePrice() printItemAttributes()

No extra variables

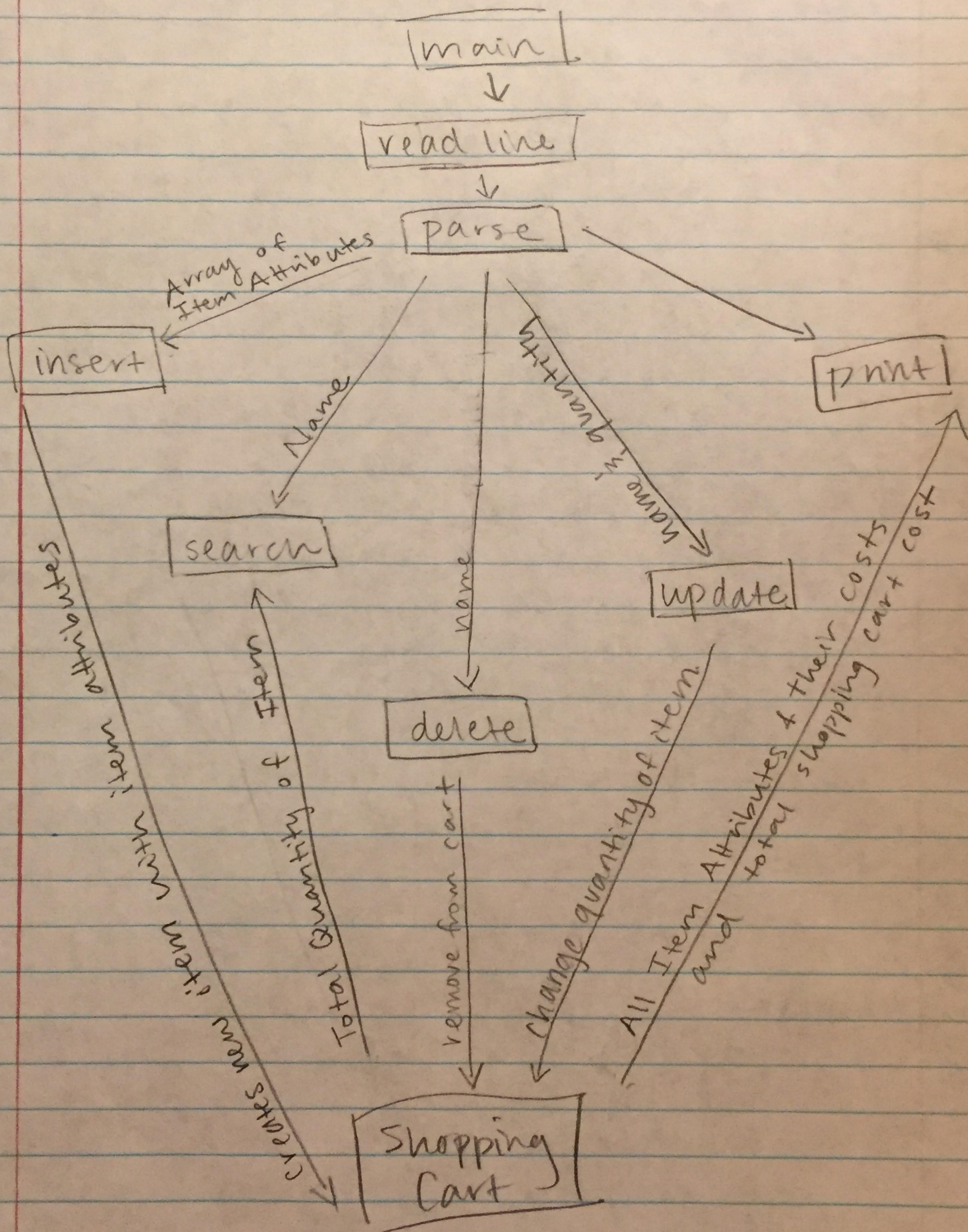
Methods that overwrite Item's methods

Grocery
perishable: boolean
isPerishable() setPerishable()
calculatePrice() printItemAttributes()

1 extra variable unique to Groceries.

2 extra variables unique to electronics

Functional Block Diagram showing the calling relationship between methods



Algorithm needed for driver logic (main method)

Start

Open File

while (file contains line) {

 while (line contains words) {

 Isolate word

 if ("insert") {

 insertItem();

 }

 if ("search") {

 searchItem();

 }

 if ("delete") {

 deleteItem();

 }

 if ("update") {

 updateItem();

 }

 if ("print") {

 print Item Attributes & Total Charge

 }

 } default : throw exception

}

deleteItem {

 loop until end of shopping cart

 if (name matches) {

 remove from array list

 i--

}

}

insert {

loop through names of items to
alphabetize them.

try (

switch (type of item)

clothing - add at index with variables

electronics - add at index with state, etc.

groceries - add at index with perishable.

default: throw variable

)

Catch any errors checking for
invalid input / insufficient input.

}

Search {

loop through arraylist comparing names
to searched name.

if equal, add quantity to total.
return total.

}

update {

loop through arraylist comparing names
to searched name

if equal, replace quantity with new quantity
break;

}