A#3 Shopping Cart Analysis

Problem: The problem is to implement a shopping cart that sells three categories of "Items": groceries, electronics, and clothing. The user adds purchases via a file with a set command format <operation> <category> <name> <price> <quantity> <weight> <optional field 1> <optional field 2>, with spaces in between each field. and lines for each set of commands. The program will compute and display the total price for the cart, tax as applicable, and shipping cost for each item (Note that all shipping of items is separate).

Questions: What classes will we need and what do they do?

- From requirements, we have an Item base class that has name, price, quantity, and weight, where weight is whole pounds, quantity is non-negative whole number (including 0), price is real number in dollars & cents, and name is a string with no spaces. We have three subclasses: Groceries, Electronics, and Clothing, where groceries can either be perishable or not, and electronics can be fragile or not.

Rules for the models?

- Perishable groceries require premium shipping, non-perishable don't.
  Electronics that are shipped to TX, NM, VA, AZ, AK have no sales tax.
  Fragile electronics require premium shipping. Clothing doesn't have premium shipping available. Shopping cart is an ArrayList of objects in alphabetical order

Pricing guidelines?

- Sales tax is 10% for clothing and electronics (where applicable for electronics)
- Standard shipping costs are 20* weight * quantity, with premium shipping incurring 20% more than standard
- Each item gets shipped separately.

I/O operations

- insert: instantiate object of appropriate type and add to ArrayList
- search: search for name of object and output # of them to screen
- delete - delete item based on name and output # deleted
- update - update quantity of item based on name and output name and new quantity
- print: print contents of cart by name alphabetically, showing all attribute values for each as well as total charges. Total charges for entire shopping cart will follow.

How to handle errors? - Report bad transaction, move on to next transaction

Additional Thoughts: each item has calculatePrice() method that returns price based on the rules