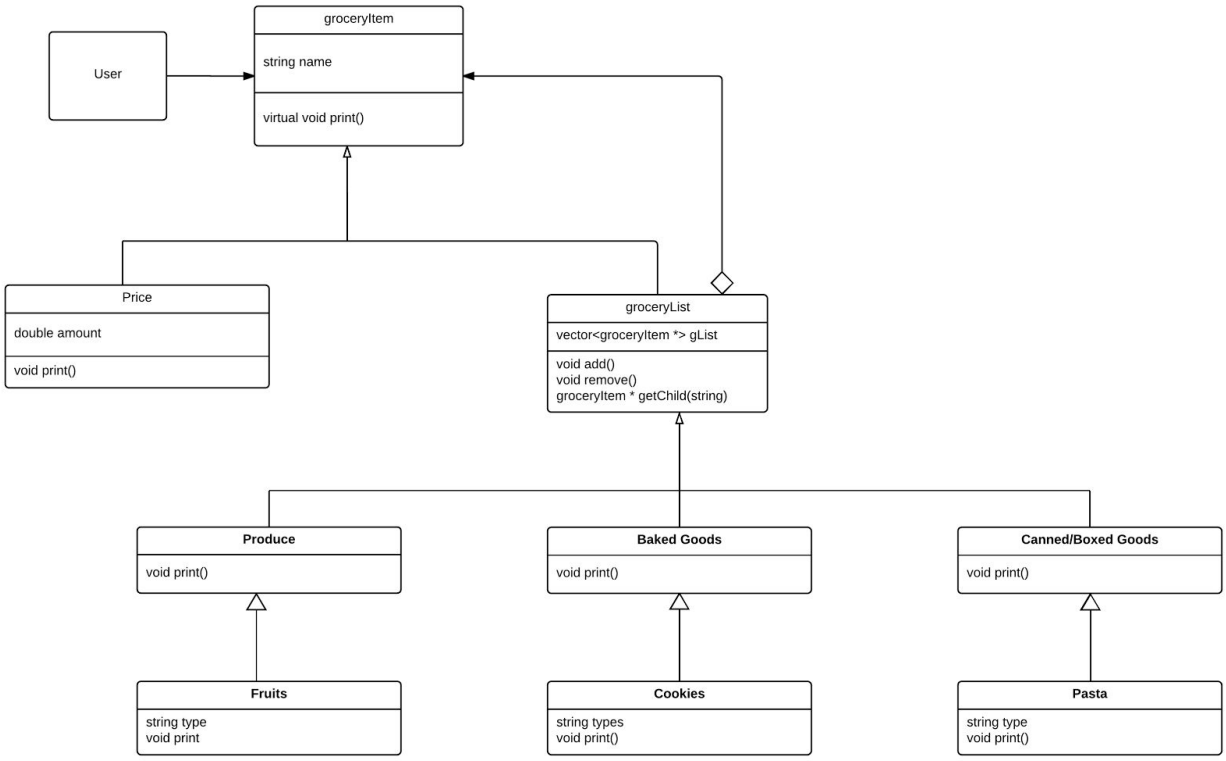
Assignment 1: Grocery Store Design
10/20 Fall 2017
Jonathan Ho and Brianna Nguyen

Introduction:

This will be a grocery store that sells a variety of items, such as food products and household necessities.

Diagram:



Classes/Class Groups:

- GroceryItem: base class for the rest of the classes in the grocery store, contains the name and a virtual void print() function.
- GroceryList: stores all the specific grocery items wanted, registering them into a vector based on their name.
- Price: contains price for each item and print() function.
- Produce: subclass of grocery items, but parent of specific fresh grocery ingredients (dairy, vegetables, fruits, ect.)
- Baked: subclass of grocery items, but parent for baked goods (cakes, cookies, etc.)
- Canned/Boxed Goods: subclass for grocery items, but parent for canned/boxed goods (soups, pastas, etc.)

Coding Strategy

We will split up the work by having one person write the classes for GroceryList, produce and half of the canned/boxed goods (soups and fruits), while the other person does GroceryItem, baked goods and the other half for canned/ boxed goods (vegetables and pasta/noodles). This

way, each person does the same amount of work since there is the same amount of classes needed to be prepared.

Roadblocks

Some code may be similar, such as having the same fruit or vegetable in different base classes with similar characteristics (for example, fruits and vegetables can be either raw or canned).. We will compare the two descriptions of both subclasses and see if we can either combine them into one, or make them unique enough to be their own subclass.

There will be too many items under the base class that we want to create, because there are many products under each base class. We will solve this by not trying to list too many unnecessary products.