Shopping Trip

Purpose

To review abstract classes and polymorphism

Directions

The total cost of a group of items at a grocery store is based on the sum of the individual product prices and the tax (which is 5.75%). Products that are considered "necessities" are not taxed, whereas products that are considered "luxuries" are.

For this practice problem you will need to download Shopping Trip Starting Code.zip from Pilot.

The Product class is abstract, and it has a method called getTotalPrice. Your task is to create two subclasses of Product: NecessaryProduct and LuxuryProduct and implement the getTotalPrice method in each of these classes appropriately. Then modify the driver program to instantiate four products (two necessary and two luxury) and store them in the product array, print out each item in the array, and display the total cost of the items.

You should not make any changes at all to Product.java, and you should only add to ShoppingTripStartingCode.java. Do not change any code that is already present.

Example

(Cheese and bread are necessities and soda and candy are luxuries)

```
Cheese $1.50
Soda $3.50
Bread $2.25
Candy $2.00

The total cost is $9.57
```

The computation is 1.50 + (3.50 * 1.0575) + 2.25 + (2.00 * 1.0575) = 9.57. Note that the price displayed next to each product is the price without tax.

Keep in mind that to receive credit your code needs to work for all valid inputs, not just for the particular example shown above.