CS 185

Aggregation of Objects

In this program you will implement an Item class and a Shopping cart class and use them in a simple simulation.

- **Put both classes in one file called** `A2.py`
- **You must name classes, files and methods exactly as described below for your code to be compatible with the simulation code and my test code.**

PART 1 (25 points)

1. Item Class (20 pts)

Create a class `Item` to represent items that can be stored in a shopping cart.

a) The attributes of the class (self) should be
   - `name`: stores the name of the item
   - `price`: stores the price for a **single** quantity of this item
   - `quantity`: stores the quantity of this item purchased

b) (2 pts) Add a constructor that takes `name`, `price` and `quantity` as arguments and sets the attributes of self to the given values.

c) Your class should have the following methods:

i. (3 pts) `updateQuantity(amt)`: increase this object's quantity by the given `amt`

ii. (3 pts) `gettotal()`: return the total price of this item (price * quantity)

iii. (3 pts)`__str__()`: return a string with the item's name, quantity, and price.
For example: "apple quantity:4, price per item:1"

iv. (2 pts) `__eq__(other)`: return true if this item has the same name as the `other` item given as argument and false otherwise

v. (2 pts) `__gt__(other)`: returns true if this item has a price greater than the other item given as an argument and false otherwise

2. (5 points) Testing Item: Add a main method to A2.py with code for testing all methods of the Item class. Next to each test write your expected results as a comment. **(Test cases without expected results receive 0 credit)**

- Be sure **not** to call magic methods directly by name. For example use print to test the __str__ method, use == to test __eq__ , etc.

3. (5 points) Comment each method you wrote in part 1c using the format:

#Purpose:
#Input:
#Output