Java Object-Oriented Concepts Unit

Lesson 7 Lab 3: Vending Machine

SOFTWARE GUILD

The Learning House

427 S 4th Street #300

Louisville KY 40202

Lesson 7 Lab 3: Vending Machine

Objective:

The objective of this lab is to continue our practice in creating programs that contain multiple classes.

Requirements:

Your assignment for this lab is to create a program that simulates a vending machine. Your program must have the following features:

1. The program should display all of the items and their respective prices when the program starts, along with an option to exit the program.
2. The user must put in some amount of money before an item can be selected.
3. Only one item can be vended at a time.
4. If the user selects an item that costs more than the amount the user put into the vending machine, the program should display a message indicating insufficient funds and then redisplay the amount the user had put into the machine.
5. If the user selects an item that costs equal to or less than the amount of money that the user put in the vending machine, the program should display the change returned to the user. Change must be displayed as the number of quarters, dimes, nickels, and pennies returned to the user.
6. Vending machine items must be stored in a file. Inventory for the vending machine must be read from this file when the program starts and must be written out to this file just before the program exits. The program must track the following properties for each item:
   a. Item name
   b. Item cost
   c. Number of items in inventory
7. When an item is vended, the program must update the inventory level appropriately. If the machine runs out of an item, it should no longer be available as an option to the user. However, the items that have an inventory level of zero must still be read from and written to the inventory file and stored in memory.

Hint: to make change, create a Change class that takes the amount of change due to the user (in pennies) and then calculates the number of quarters, dimes, nickels, and pennies due back to the user. This class should have accessors for each of the coin types.

Before coding, you must create a flowchart and UML class diagram (in Dia) for your program.

Please show the diagrams to the instructor before coding.