MIS 304 Final Project – Spring 2021

Due on Canvas: Friday, May 14th 11:59pm

Invoice

A small baking supply shop wants to better manage their inventory. To help them, write a program to store and manage existing inventory, and allow customers to purchase or return bakery supplies. The shop has provided a text file containing existing inventory data for you. You will use classes to manage this data. The inventory data found in Inventory.txt looks like this:

```
244
Large Cake Pan
7
19.99
576
Assorted Sprinkles
3
…
```

You must write a program to process the data in Inventory.txt using instances of the Inventory class. Each time the customer purchases or returns an item, you must use an instance of TransactionItem to store that information. When a customer completes a transaction (a customer may both purchase and return items in the same transaction), you will display an invoice to the console, and create a new text file (UpdatedInventory.txt) with all inventory information and updated quantities.

Menu to be displayed for the user:

Based on the data from Inventory.txt, create a menu of items for the user. As the user purchases or returns items, you will re-display the menu with updated quantities. This menu should be generated dynamically from the (up-to-date) inventory data, NOT hard-coded.

```
ID  Item                         Price    Qty Available
244 Large Cake Pan               $19.99   7
576 Assorted Sprinkles           $12.89   3
212 Deluxe Icing Set             $37.97   6
Enter 0 when finished
```

Note: When running the program, the user will enter the ID of the item, NOT the item name (e.g. 244, not "Large Cake Pan"). Also, do not worry about the order that the items display in the menu or the output file – as long as all items are listed with correct quantities, you will get credit.

Program requirements:

1. Define a class to store inventory information. Please refer to the UML for class design details.
2. Define a class to store information about each item purchased or returned. Refer to the UML for class design details. This class is called TransactionItem.
3. Your program MUST read all of the information about the inventory from Inventory.txt and store each item in an instance of the Inventory class. It is strongly recommended that you use a list of objects to keep track of all inventory (each item in the list would be an instance of the class).
4. Your program MUST store information about an item purchased or returned in an instance of the TransactionItem class. It is strongly recommended that you use a list of objects to keep track of all items purchased or returned (each item in the list would be an instance of the class).
5. Customers can purchase more than one item before the invoice is generated. They can also return more than one item. They can also purchase and return items in the same session. (i.e., your program needs to process more than one item in one transaction. It should NOT just process one item, display the invoice and then quit).
6. Your program MUST define and call the following user-defined functions, in addition to the `main` function, in a module that is different from Inventory and TransactionItem (e.g. LastName_FirstName_FP.py). You may create additional functions, but the following functions in the main program are required for full credit:
    - process_inventory HINT: Open the input file, create the inventory instances and add them to the inventory list

- print_inventory HINT: Use the inventory data from the inventory list to create the menu.
- get_item_id from the user. Verify the id is a valid item number or 0. **If they enter 0, it means that they would like to quit the program.**
- write_updated_inventory to an output file (UpdatedInventory.txt). Please make sure that your output file has a different name from the input file so that we can run the program multiple times.
- print_invoice This function should print the details of the items purchased or returned (id, name, quantity, total price for the item) as well as the total of all items, tax, and total amount due. Sales tax is 8.5%

Other program details:

- If the user doesn't purchase or return any items, print a message thanking them for visiting. Do not print an invoice. Do print inventory data to UpdatedInventory.txt.
- Be sure that your program works for input files that contain any number of items in Inventory.txt. The data structure in each file will remain the same as the sample Inventory.txt file provided.
- If error handling or data validation is not specified in the instructions or grade script, you do NOT have to include this logic in your program.
- In the Inventory or TransactionItem classes, all program logic must be included within functions/methods. There should not be input or print statements in your Inventory or Transaction classes.

This assignment MUST be created individually by you or with one partner. You must turn in your OWN Python files. You MAY NOT share files with other pairs or individuals! YOU must write the source code yourself or with one partner.

Instructions

- Test your program to ensure that it works correctly.
- Be sure to run your final program using the data included in the sample input file to ensure your program works properly.
- Be sure to format your output according to the instructions above and in the sample grade script.
- You may perform the tasks in any order as long as your output follows the order of the output in the above instructions.

Options for completing the Final Project:

- **Option 1:** You may choose not to complete the project. If you choose not to complete the project, I ask you to turn in one paragraph explaining issues you run into. You will receive 65% for the Final Project grade if you choose this option. This is also the minimum grade you will receive for this project.

- **Option 2:** You MAY complete the project with ONE partner. If you choose to work with a partner, submit ONE set of .py files and make sure that both of your names are in the header of all files. Working with a partner is a great way to gain experience coding collaboratively!

- To create a group, there are two options:

    1. Choose your partner: self signup (with your group partner) on Canvas.
        - Canvas -> People -> Project -> Self Sign-up Final Project
    2. Request help finding a partner: Fill out the canvas survey and I will help you find a partner.

        - Canvas -> Quizzes -> Surveys -> Final Project--groups

- **Option 3:** You are NOT required to work with a partner. You may choose to work alone. Put only your name on your work.

To receive full credit:

- Submit your Python files (.py files) to Canvas. The class files should be named after the class using lowercase letters (e.g. inventory.py contains the Inventory class). The file with the main function should be named LastName_FirstName_LastName_FirstName_FP.py (if you worked with a partner) or LastName_FirstName_FP.py for full credit.
- Submit your Python files to Canvas using the Assignment submission feature by 11:59pm Austin time on 5/14/2020.