ENGR 101 - Introduction to Programming

Mini Project 2

January 8, 2018

(Due on Monday, January 22 by 5 pm)

In this mini project, you are going to develop a console-based e-commerce application! This software will be used by customers to order grocery items. The primary goal of this mini project is to practice using various data structures (i.e., list, dictionary, tuple). Hence, please use every opportunity to employ a data structure in your code. At various points, you will be forced to use particular data structure as explained in the "Implementation Notes" section at the end. The detailed explanations about how you are going to develop this application are provided below (Note: example console output from the application is written in ORANGE).

Text-based User Interface:

When your program first runs, it will ask the user to log in by providing a user name and password as follows. In your code, at least, the following users should be already defined with user name 'ahmet' and password 'sehir123', and 'meryem' with password '4444'. You may define additional users if you like (Please use the most proper data structure to store user name / password information. Hint: Consider a dictionary).

```
****Welcome to Sehir Online Market****

Please log in by providing your user credentials:
User Name:
```

As long as the user enters invalid credentials, the program should ask the user to try again. The user successfully logs in when the provided credentials are valid. The following shows the interface for the initial login screen with a scenario where the user first enters incorrect credentials, and then successfully logs in during the second trial.

```
****Welcome to Sehir Online Market****

Please log in by providing your user credentials:

User Name: ahmet
Password: 123

Your user name and/or password is not correct. Please try again!

User Name: ahmet
Password: 1234

Successfully logged in!
```

Once the user enters valid credentials, your program should greet the user with the following message, and provide a menu of the things to do. User will do a selection by entering the corresponding menu number, and accordingly different information will be shown and/or requested.

```
Welcome, ahmet! Please choose one of the following options by entering the corresponding
menu number.
```

```
Please choose one of the following services:
    1. Search for a product
    2. See Basket
    3. Check Out
    4. Logout
    5. Exit

Your Choice:
```

If the user enters an invalid menu entry, the program should warn the user to provide a valid menu number.

When **main menu item 1** is chosen, the user will be asked to provide search term(s). Then, your program will display a list of all items from inventory whose name **includes** the user-provided search term(s), and whose stock amount is greater than 0. For instance, when the user searches for "Juice", the program should display all items whose name includes 'juice' and available in the stocks as follows. Product name checking should be case insensitive (i.e., you should not differentiate between lower and upper case letters).

```
Please choose one of the following Services:
    1. Search for a product
    2. See Basket
    3. Check Out
    4. Logout
    5. Exit

Your Choice: 1
What are you searching for? Juice
found 3 similar items:

    1.grape juice  9$
    2.orange juice  9$
    3.apple juice  9$

Please select which item you want to add to your basket (Enter 0 for main menu):
```

Note that each matching item is assigned a number, and the program asks the customer to enter the number corresponding to a product to add to his/her basket. A customer may cancel the current search session and go back to main menu by entering 0 at this stage. Once the customer chooses a product to add to the basket, the program will ask for the amount. If the requested amount is not exceeding the stock amount of the selected product in the inventory, the product should be added to the basket, and the customer should be notified about it, and then the main menu should be displayed.

```
Please select which item you want to add to your basket (Enter 0 for main menu): 3
Adding apple juice. Enter Amount: 2
Added apple juice into your Basket.
Going back to main menu...

Please choose one of the following services:
    1. Search for a product
    2. See basket
    3. Check out
    4. Logout
    5. Exit

Your Choice:
```

If the requested amount is over the stock amount, then the program should ask the user to provide another amount (or Enter 0 to go back to main menu).

```
Please select which item you want to add to your basket (Enter 0 for main menu): 1
Adding grape juice. Enter Amount: 1000
Sorry! The amount exceeds the limit, Please try again with smaller amount
    Amount (Enter 0 for main menu):
```

If the search result is empty, the user will be requested to provide another search term (or, the user may go back to main menu by entering 0).

```
What are you searching for? Lemon
Your search did not match any items. Please try something else (Enter 0 for main menu):
```

Anywhere in the program, if the user makes a selection outside of the menu shown, the program must ask the user to re-enter a valid menu item.

When **main menu item 2** is chosen, the user will be able to see the things he/she added to their basket, the price of the item per piece, the amount of the items in the basket, and lastly, the total price of the items in the basket.

```
Your basket contains:
    1.grape juice price=9$ amount=1 total=9.0$
    2.egg price=2$ amount=2 total=4.0$
Total 13.0$
```

Please note that if the basket is empty, your program should show that the user's basket is empty, and the total price of items in the user's basket is equal to $0. After displaying items in your basket, you'll be directed to another menu (to avoid confusion, refer to this menu as **basket sub menu**):

```
Please choose an option:
    1. Search for a product
    2. See Basket
    3. Check Out
    4. Logout
    5. Exit

Your selection:
```

If user chooses **basket sub-menu option 1**, user will select an item from their basket and update its amount. You should display the content of user's basket, after the changes are applied. Reminder: Again, the new amount should be checked to make sure that it is not exceeding the stock amount. (Note: the inventory stock amounts of items in basket are not decreased until checkout.)

```
Please select which item to change its amount: 1
Please type the new amount: 2

Your basket now contains:
    1.grape juice price=9$ amount=2 total=18.0$
    2.egg price=2$ amount=2 total=4.0$

Please Choose an option:
    1.Update amount
    2.Remove an item
    3.Check out
    4.Go back to main menu

Your selection:
```

After this user will be redirected to the **basket sub-menu** again**.** If user chooses **basket sub-menu option 2**, user will select an item from their basket that will be removed from their basket. Again, please display the content of user's basket, after the removal. After this, user will be redirected to the **basket sub-menu** again**.**

If user chooses **basket sub-menu option 3**, or **main menu item 3,** user will proceed to check out and your program will print the receipt as shown below. For the last part of your receipt, program should display the real date and time of the transaction (you may use now() function from datetime module).

```
Processing your receipt...

******* Sehir Online Market ********
************************************
    44 44 0 34
    sehir.edu.tr
------------------------------------
grape juice 9$ amount=2 total=18.0$
egg 2$ amount=2 total=4.0$
------------------------------------
 Total   22.0$
------------------------------------
2018/01/08   17:00
Thank You for using our Market!
```

During checkout, the program should decrease the stock amount of each user-bought product properly in the inventory. After checkout, the program should show the main menu.

If user chooses **main menu item 4,** the user should log out, and the program should ask for user name and password again as explained at the beginning. At this stage, the user's basket content should still be stored, even after logging out. When the same user logs in again, s/he should be able to see items that s/he added into the basket in his previous session(s).

If user chooses **main menu item 5,** the program should exit.

Implementation Notes:

- A dictionary must be used to keep user account information.
- A nested dictionary must be implemented for the basket (key: user name, value: another dictionary where key is product name, value is amount).
- Whenever an operation is performed by the user, the above data structures should be updated properly to reflect the changes.
- An additional dictionary will be used for the Market's inventory. Please copy and paste the following inventory dictionary into your project code file, and use it in your project. In this dictionary, keys are product names, values are lists where the first item shows the stock amount, while the second item shows the unit price.

```
inventory={'asparagus':[10,5],'broccoli':[15,6],'carrots':[18,7],
        'apples':[20,5],'banana':[10,8],'berries':[30,3],
        'eggs':[50,2],'mixed fruit juice':[0,8],'fish sticks':[25,12],
        'ice cream':[32,6], 'apple juice':[40,7], 'orange juice':[30,8], 'grape juice':[10,9],}
```

- You may check the link below to get further information about datetime module:

(https://docs.python.org/2/library/datetime.html)

Warnings:

- **Do not** talk to your classmates on project topics when you are implementing your projects (This is serious). **Do not** show or email your code to others (This is even more serious). If you need help, talk to your TAs or the instructor, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have **serious** consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve "**should**", "**should not**", "**do not**", and other underlined/bold font statements.
- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.
- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is **important** to understand all the details in your submitted code. You may be interviewed about any part of your code.

How and when do I submit my project? :

- Projects may be done individually or as a small group of two students (doing it individually is recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for details).
- Submit your own code in a **single** Python file. Name your code file with your and your partner's first and last names (see below for naming). o
    - If your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do **not** use any Turkish characters in file name).
    - If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
    - Those who **do not** follow the above naming conventions **will get 5 pts off** of their grade.
- Submit it online on LMS by **5 pm on Monday, Jan. 22, 2018**

Late Submission Policy:

- -10%: Submissions between 17:01 – 18:00 on the due date
- -20%: Submissions between 18:01 – midnight (00:00) on the due date
- -30%: Submissions which are 24 hour late.
- -50%: Submissions which are 48 hours late.
- Submission more than 48 hours late will not be accepted.

Grading Criteria? :

Code Organization			Functionality				
Meaningful variable names (3 pts)	**Proper use of functions – compact code with no repetitions (4 pts)**	**Sufficient commenting (4 pts)**	**Search functionality with all of its submenus (25 pts)**	**Basket functionality with all of its submenus (30 pts)**	**Checkout functionality with all of its submenus (20 pts)**	**Logout functionality (10 pts)**	**Others (5 pts)**

Interview evaluation: Your grade from interview will be between 0 and 1, and it will be used as a coefficient to compute your final grade. For instance, if your initial grade was 80 before the interview, and your interview grade is 0.5, then your final grade will be 80*0.5 = 40. Not showing up for the interview appointment will **result in** grade 0.

Have further questions?:

Please contact your TAs if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**

Good Luck!